UNH–IOL
NVMe Testing Service

Test Plan for NVMe ZNS Conformance
Version 16.0
Target Specification: NVMe ZNS 1.1
Technical Document

Last Updated: October 20, 2021

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

©2021 UNH–IOL

©2021 UNH–IOL

©2021 UNH–IOL

©2021 UNH–IOL

©2021 UNH–IOL

©2021 UNH–IOL

©2021 UNH–IOL

MODIFICATION RECORD

2021 May 3 (Version 15.0) Initial Release
David Woolf:

2021 September 23 (Version 16.0) Final Release
David Woolf:

1. Fixed typo in title of Test 2.1.11.
2. Added Test 1.3 Case 4: Get Log Page: SMART / Health Log to check the Write command effects.
3. Updated Test 2.2.7 to clarify the SLBA field to be used ind the Zone Management Send command with the Zone Send Action set to Open Zone that if the Select All bit is cleared to '0', and the zone specified by the SLBA field is in the ZSF:Full state, the ZSRO:Read Only state or the ZSO:Offline state, the command shall be aborted with a status code of Invalid Zone State Transition.
4. Added Test 2.2.10 to check requirements around the Zone Management Send command with the Zone Send Action set to Open Zone that if the Select All bit is set to '1', and the operation causes the number of Open Resources to exceed the value specified by the Maximum Open Resources field, then the command shall be aborted with a status code of Too Many Open Zones, and no zone state transitions shall occur.
5. Added Test 2.6.17 to check requirements around the Zone Descriptor Data Structure, that the Zone Capacity (ZCAP) field shall be less than or equal to the Zone Size field.
6. Updated procedure in Test 2.9.6 to ensure that zone was formatted with protection type 1 prior to executing Zone Append command.
7. Updated procedure in Test 2.9.7 to ensure that zone was formatted with protection type 3 prior to executing Zone Append command.
8. Added Test 2.9.8 to check Zone append operation when writing to zones in the ZSF states.
9. Updated Test 2.11.7 to check for the correct error, B8h instead of BCh.
10. Added Test 2.12.3 to check requirements that for zones with a valid write pointer: An attempt to read logical blocks greater than or equal to the write pointer and within that zone shall be regarded as an attempt to read unwritten logical blocks. Refer to the Deallocate section in the NVMe Base specification.
11. Added Tests 2.13.1, 2, 3, 4, 5, to check requirements in TP 4053 Zoned Namespaces that a logical block shall be marked as allocated when that logical block is written with: a Write command; a Write Uncorrectable command; a Write Zeroes command that does not deallocate the logical block (refer to the Deallocate section in the NVMe Base specification); a Copy command; and a Zone Append command.
12. Added Test 2.13.6 to check requirements that All logical blocks in a zone shall be marked as deallocated when the zone is in the ZSE:Empty State.
13. Added Test 2.14.1, 2, 3, 4 to check requirements around the Sanitize Command when the command has the following attributes: 'No Deallocate After Sanitize' bit =1b, NODMMAS=01b, NDI=0b and the Zone state over the command is ZSF:Full, Sanitize Status=001b, and Logical Block Content is set to 'Overwrite', that the logical block contents shall be the overwrite pattern specified in the Sanitize command.
14. Added Test 2.15.1, 2, 3, 4 to check requirements that Zones that have associated Active Resources shall transition to the ZSF:Full state when the zoned namespace becomes write protected. Refer to the Namespace Write Protection section in the NVMe Base specification.
15. Added Test 2.16 Cases 1, 2, 3, 4, 5, 6, to check proper operation of Zone Management Send with Action set to Close Zone.
16. Added Test 2.5.10 to check requirements around the Zone Management Send command with the Zone Send Action set to Set Zone Descriptor Extension, and the operation causes the number of Active Resources to exceed the value specified by the Maximum Active Resources field, then the command shall be aborted with a status code of Too Many Active Zones, and no zone state transitions shall occur.
17. Added Test 2.17.1, 2, 3, 4, 5 to check requirements around the Zone Management Send command that if the command SLBA field does not specify the starting logical block for a zone in the specified zoned namespace and the Select All bit is cleared to '0', then the command shall be aborted with a status code of Invalid Field in Command.
18. Added Test 3.1.1-13 to check requirements for the Compare Command when used with a ZNS device. Note that tests 3.1.1-8 are equivalent to tests 2.1.1-8 in the NVM Command Set test suite document.
19. Added Test 3.2.1-4 to check requirements for the Copy Command when used with a ZNS device. Note that tests 3.2.1-4 are equivalent to tests 2.14.1-4 in the NVM Command Set test suite document.

©2021 UNH–IOL

20. Added Test 3.3.1-12 to check requirements for the Dataset Management Command when used with a ZNS device. Note that tests 3.3.1-12 are equivalent to tests 2.2.1-12 in the NVM Command Set test suite document.
21. Added Test 3.4.1-3 to check requirements for the Flush Command when used with a ZNS device. Note that tests 3.4.1-3 are equivalent to tests 2.6.1-3 in the NVM Command Set test suite document.
22. Added Test 3.5.1-14 to check requirements for the Read Command when used with a ZNS device. Note that tests 3.5.1-11 are equivalent to tests 2.3.1-11 in the NVM Command Set test suite document.
23. Added Test 3.6.1-5 to check requirements for the Verify Command when used with a ZNS device. Note that tests 3.6.1-4 are equivalent to tests 2.1.1-4 in the NVM Command Set test suite document.
24. Added Test 3.7.1-13 to check requirements for the Write Command when used with a ZNS device. Note that tests 3.7.1-11 are equivalent to tests 2.4.1-11 in the NVM Command Set test suite document.
25. Added Test 3.8.1-12 to check requirements for the Write Uncorrectable Command when used with a ZNS device. Note that tests 3.8.1-7 are equivalent to tests 2.5.1-7 in the NVM Command Set test suite document.
26. Added Test 3.9.1-16 to check requirements for the Write Zeroes Command when used with a ZNS device. Note that tests 3.9.1-11 are equivalent to tests 2.7.1-11 in the NVM Command Set test suite document.
27. Added Test 4.1.1-5 to check requirements for the Reservation Report Command when used with a ZNS device. Note that these tests are equivalent to tests 7.1.1-5 in the NVM Command Set test suite document.
28. Added Test 4.2.1-5 to check requirements for the Reservation Registration Command when used with a ZNS device. Note that these tests are equivalent to tests 7.2.1-5 in the NVM Command Set test suite document.
29. Added Test 4.3.1-3 to check requirements for the Reservation Acquire Command when used with a ZNS device. Note that these tests are equivalent to tests 7.3.1-3 in the NVM Command Set test suite document.
30. Added Test 4.1.1-5 to check requirements for the Reservation Release Command when used with a ZNS device. Note that these tests are equivalent to tests 7.4.1-4 in the NVM Command Set test suite document.

2021 October 2 (Version 16.0)

Carter Snay:

1. Test case 1.1.1 updated from FYI to M.
2. Test case 1.1.2 updated from FYI to M.
3. Test case 1.3.1 updated from FYI to M.
4. Test case 1.3.2 updated from FYI to M. (Append support required)
5. Test case 2.1.1-5 updated from FYI to M.
6. Test case 2.2.1-4, 7 updated from FYI to M.
7. Test case 2.3.1, 5 updated from FYI to M.
8. Test case 2.6.1-12 updated from FYI to M.
9. Test case 2.9.1-5 updated from FYI to M. (Append support required)
10. Test case 2.12.1 updated from FYI to M. (RAZB support required)

©2021 UNH–IOL

ACKNOWLEDGMENTS

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

David Woolf UNH InterOperability Laboratory

©2021 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 ZNS 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 ZNS specification, hereafter referred to as the "NVMe ZNS Specification"). Successful completion of these tests provide a reasonable level of confidence that the Device Under Test (DUT) will function properly in many NVMe ZNS environments.

The tests contained in this document are organized in order to simplify the identification of information related to a test, and to facilitate in the actual testing process. Tests are separated into groups, primarily in order 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

©2021 UNH–IOL

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.

©2021 UNH–IOL

REFERENCES

The following documents are referenced in this text:

1. NVMe Zoned Namespaces Command Set Specification Revision 1.1 (May 18, 2021)

©2021 UNH–IOL

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

©2021 UNH–IOL

Group 1: Zoned Admin Command Set

Overview:

This section describes a method for performing conformance verification for NVMe ZNS products implementing Zoned Admin Commands defined in Chapter 3 of the NVMe ZNS Specification.

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).

©2021 UNH–IOL

Test 1.1 – Identify Namespace Data Structures (M)

Purpose: To verify that an NVMe ZNS capable controller can properly process an Identify Command.

References:

[1] NVMe ZNS Specification 3.1

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: January 7, 2021

Discussion: The Zoned Namespace Command Set supports two I/O Command Set specific Identify Namespace data structures:

a) the NVM Command Set Identify Namespace data structure; and
b) the Zoned Namespace Command Set Identify Namespace data structure.

Test Setup: See Appendix A.

Case 1: Zoned Namespace Command Set Identify Namespace Data Structure (CNS 05h) (M)

Test Procedure:

1. Configure the Testing Station to transmit an Identify Command with CNS=05h.

Observable Results:

1. Verify that the Identify command completes with Status Success.
2. Verify that the ZSZE value is not cleared to 0h.
3. Verify that the MOR field is less than or equal to the MAR field.

Case 2: Zoned Namespace Command Set Identify Controller Data Structure (CNS 06h) (M)

Test Procedure:

1. Configure the Testing Station to transmit an Identify Command with CNS=01h and read the MDTS field.
2. Configure the Testing Station to transmit an Identify Command with CNS=06h and read the ZASL field.

Observable Results:

1. Verify that the Identify command completes with Status Success.
2. Verify that the ZASL field is less than or equal to the MDTS field.

Possible Problems: None.

©2021 UNH–IOL

Test 1.2 – Asynchronous Events (FYI)

Purpose: To verify that an NVMe ZNS capable controller can properly generate Asynchronous Event Notifications when a Zone Descriptor changes.

References:

[1] NVMe ZNS Specification 3.3

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: January 25, 2021

Discussion: If enabled, an Asynchronous Event is transmitted when a Zone Descriptor is changed.

Test Setup: See Appendix A.

Case 1: Zoned Descriptor Changed Notice Enabled (FYI)

Test Procedure:

1. Configure the Testing Station to send an Identify Command for CNS 01h to the DUT.
2. Check Bit 27 of the Optional Asynchronous Events Supported (OAES) field in the Identify Controller Data Structure, I/O Command Set Independent (CNS 01h). If this bit is set to 0 then this test is not applicable.
3. Configure the Testing Station to transmit a Set Feature command for FID 0Bh Asynchronous Event Configuration to set Zone Descriptor Changed Notices to 1.
4. Perform an action on a Zone which will generate a Zone Descriptor Changed asynchronous event notice. One such action may be a Format NVM command which will cause the Zone to transition to the ZSE or ZSO state.
5. Wait for the DUT to issue an Asynchronous Event Notification with a Zoned Descriptor Changed Notice.
6. Configure the Testing Station to transmit a Get Log Page command for LID BFh Changed Zone List.
7. Configure the Testing Station to transmit a Zone Receive command with Action set to Report Zones for any zone that is indicated as changed in the received Changed Zone List.

Observable Results:

1. Verify that the DUT generated an Asynchronous Notification in response to the Zone Descriptor being changed.
2. Verify that the Report Zones data structure returned by the DUT in response to the Zone Receive command with Action set to Report Zones indicated that the Zone was in the ZSE or ZSO state, if a Format NVM command was used to cause the Zone Descriptor Changed asynchronous event notice. If another command was used, ensure that the Report Zones indicated that the Zone was in the expected state.

Case 2: Zoned Descriptor Changed Notice Disabled (FYI)

Test Procedure:

1. Configure the Testing Station to send an Identify Command for CNS 01h to the DUT.
2. Check Bit 27 of the Optional Asynchronous Events Supported (OAES) field in the Identify Controller Data Structure, I/O Command Set Independent (CNS 01h). If this bit is set to 0 then this test is not applicable.
3. Configure the Testing Station to transmit a Set Feature command for FID 0Bh Asynchronous Event Configuration to set Zone Descriptor Changed Notices to 0.
4. Perform an action on a Zone which normally would generate a Zone Descriptor Changed asynchronous event notice. One such action may be a Format NVM command which will cause the Zone to transition to the ZSE or ZSO state. It may be best to use the same action as used in Case 1 above.
5. Configure the Testing Station to transmit a Get Log Page command for LID BFh Changed Zone List.
6. Configure the Testing Station to transmit a Zone Receive command with Action set to Report Zones for any zone that is indicated as changed in the received Changed Zone List.

©2021 UNH–IOL

Observable Results:

1. Verify that the DUT did not generate an Asynchronous Notification in response to the Zone Descriptor being changed.
2. Verify that the Report Zones data structure returned by the DUT in response to the Zone Receive command with Action set to Report Zones indicated that the Zone was in the ZSE or ZSO state, if a Format NVM command was used to cause the Zone Descriptor Changed asynchronous event notice. If another command was used, ensure that the Report Zones indicated that the Zone was in the expected state.

Possible Problems: None.

©2021 UNH–IOL

Test 1.3 – Log Pages (FYI)

Purpose: To verify that an NVMe ZNS capable controller can properly process a Get Log Page Command.

References:

[1] NVMe ZNS Specification 3.4

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: January 21, 2021

Discussion: The Zoned Namespace Command Set supports Log Pages specific to ZNS.

Test Setup: See Appendix A.

Case 1: Get Log Page: Changed Zone List (M)

Test Procedure:

1. Configure the Testing Station to transmit a Get Log Page Command with LID=BFh Changed Zone List.

Observable Results:

1. Verify that the Get Log Page command completes with Status Success.
2. Verify that the each ZSLBA value appears only once in the Zone Identifier List and that unused entries are zero-filled.
3. Verify that the controller sets the Number of Zone Identifiers field in the log page to the number of valid entries that follow.

Case 2: Get Log Page: SMART / Health Information Log with Zone Append(M)

Test Procedure:

1. Determine support for the Zone Append Command by performing the following steps. If the Zone Append command is not support this test is not applicable.
    a. Configure the Testing Station to set CC.CSS to indicate support for ZNS.
    b. Configure the Testing Station to perform a Get Log Page command for LID=05h, Commands Supported and Effects.
    c. Check the CSUPP bit in IOCS125 of the returned log page. IOCS125 corresponds to support for the Zone Append command. The CSUPP bit indicates if this command is supported or not.
2. Configure the Testing Station to transmit a Get Log Page Command with LID=02h SMART/ Health Information.
3. Configure the Testing Station to transmit a Zone Append command, with valid ZSLBA and PRINFO fields.
4. Configure the Testing Station to transmit a Get Log Page Command with LID=02h SMART/ Health Information

Observable Results:

1. Verify that the second SMART / Health Information Log returned had a higher Data Units Written value than the first SMART / Health Information Log proportional to the Zone Append command which was performed.

Case 3: Get Log Page: Endurance Group Log (FYI)

Test Procedure:

1. Configure the Testing Station to transmit an Identify Command with CNS=01h and read the CTRATT field.
2. Determine if Endurance Groups are supported by checking Bit 4 of the CTRATT field. If this bit is set to 0 then this test is not applicable.

©2021 UNH–IOL

3. Determine support for the Zone Append Command by performing the following steps. If the Zone Append command is not support this test is not applicable.
    a. Configure the Testing Station to set CC.CSS to indicate support for ZNS.
    b. Configure the Testing Station to perform a Get Log Page command for LID=05h, Commands Supported and Effects.
    c. Check the CSUPP bit in IOCS125 of the returned log page. IOCS125 corresponds to support for the Zone Append command. The CSUPP bit indicates if this command is supported or not.
4. Configure the Testing Station to transmit a Get Log Page Command with LID=09h Endurance Group Log.
5. Configure the Testing Station to transmit a Zone Append command, with valid ZSLBA and PRINFO fields.
6. Configure the Testing Station to transmit a Get Log Page Command with LID=09h Endurance Group Log.

Observable Results:

1. Verify that the second Endurance Group Log returned had a higher Data Units Written value than the first Endurance Group Log proportional to the Zone Append command which was performed.

Case 4: Get Log Page: SMART / Health Information Log with Write command (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Get Log Page Command with LID=02h SMART/ Health Information.
2. Configure the Testing Station to transmit a Write command to write to an open Zone.
3. Configure the Testing Station to transmit a Get Log Page Command with LID=02h SMART/ Health Information

Observable Results:

1. Verify that the second SMART / Health Information Log returned had a higher Data Units Written value than the first SMART / Health Information Log proportional to the Write command which was performed.

Possible Problems: None.

©2021 UNH–IOL

Group 2: Zoned Namespace Command Set I/O Commands

Overview:

This section describes a method for performing conformance verification for NVMe ZNS products implementing Zoned I/O Commands defined in Chapter 4 of the NVMe ZNS Specification.

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).

©2021 UNH–IOL

Test 2.1 – Zone Management Send: Finish Zone (FYI)

Purpose: To verify that an NVMe ZNS capable controller can properly process a Zoned Management Send command.

References:

[1] NVMe ZNS Specification 4.3.1.2

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: January 7, 2021

Discussion: The Zone Management Send command requests an action on one or more zones. The command uses the Data Pointer, Command Dword 10, Command Dword 11 and Command Dword 13 fields. All other command specific fields are reserved. The Zoned Managemetn Send command supports the following actions: Close Zone, Finish Zone, Open Zone, Reset Zone, Offline Zone, Set Zone Descriptor Extension.

Test Setup: See Appendix A.

Case 1: ZSIO Finish Zone Select All=0 (M)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSIO: Implicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Finish Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSF: Full State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 2: ZSEO Finish Zone Select All=0 (M)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSEO: Explicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Finish Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSF: Full State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 3: ZSC Finish Zone Select All=0 (M)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSC: Closed state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Finish Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

©2021 UNH–IOL

1. Verify that the current Zone State at the end of the test is ZSF: Full State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 4: ZSE Finish Zone Select All=0 (M)

Test Procedure:
1. Configure the Testing Station to put the DUT into the ZSE: Empty state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Finish Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:
1. Verify that the current Zone State at the end of the test is ZSF: Full State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 5: ZSF Finish Select All=0 (M)

Test Procedure:
1. Configure the Testing Station to put the DUT into the ZSF: Full state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Finish Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:
1. Verify that the current Zone State at the end of the test is ZSF: Full State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 6: ZSRO Finish Zone Select All=0 (FYI)

Test Procedure:
1. Configure the Testing Station to put the DUT into the ZSRO: Read Only state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Finish Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:
1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status ‘Invalid Zone State Transition’ BFh.

Case 7: ZSO Finish Zone Select All=0 (FYI)

Test Procedure:
1. Configure the Testing Station to put the DUT into the ZSO: Offline state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Finish Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:
1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status ‘Invalid Zone State Transition’ BFh.

©2021 UNH–IOL

Case 8: ZSIO Finish Zone Select All=1 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSIO: Implicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Finish Zone and Select All=1, and a valid SLBA, and a valid SLBA.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSF: Full State.
2. Verify that each Zone Management Send command completes with Status Success.
3. Verify that the SLBA field is ignored and the Zone Management Send command completes according to Select All =1.

Case 9: ZSEO Finish Zone Select All=1 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSEO: Explicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Finish Zone and Select All=1, and a valid SLBA.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSF: Full State.
2. Verify that each Zone Management Send command completes with Status Success.
3. Verify that the SLBA field is ignored and the Zone Management Send command completes according to Select All =1.

Case 10: ZSC Finish Zone Select All=1 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSC: Closed state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Finish Zone and Select All=1, and a valid SLBA.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSF: Full State.
2. Verify that each Zone Management Send command completes with Status Success.
3. Verify that the SLBA field is ignored and the Zone Management Send command completes according to Select All =1.

Case 11: ZSF Finish Zone Select All=1 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSF: Full state. This can be accomplished by configuring the Testing Station acting as a Host to send a Zone Management Send command with the action of Finish Zone and Select All=1, and a valid SLBA.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Perform a Sequential Write operation to the Zone in the ZSF state.

©2021 UNH–IOL

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSF: Full State.
2. Verify that the Write operation is aborted with status Zone is Full (B9h).
3. Verify that the SLBA field is ignored and the Zone Management Send command completes according to Select All =1.

Possible Problems: None.

©2021 UNH–IOL

Test 2.2 – Zone Management Send: Open Zone (FYI)

Purpose: To verify that an NVMe ZNS capable controller can properly process a Zoned Management Send command.

References:

[1] NVMe ZNS Specification 4.3.1.3

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: January 7, 2021

Discussion: The Zone Management Send command requests an action on one or more zones. The command uses the Data Pointer, Command Dword 10, Command Dword 11 and Command Dword 13 fields. All other command specific fields are reserved. The Zoned Managemetn Send command supports the following actions: Close Zone, Finish Zone, Open Zone, Reset Zone, Offline Zone, Set Zone Descriptor Extension.

Test Setup: See Appendix A.

Case 1: ZSIO Open Zone Select All=0 (M)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSIO: Implicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Open Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSEO: Explicitly Opened State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 2: ZSC Open Zone Select All=0 (M)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSC: Closed state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Open Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSEO: Explicitly Opened State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 3: ZSE Open Zone Select All=0 (M)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSE: Empty state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Open Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

©2021 UNH–IOL

1. Verify that the current Zone State at the end of the test is ZSEO: Explicitly Opened State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 4: ZSEO Open Zone Select All=0 (M)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSEO: Explicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Open Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSEO: Explicitly Opened State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 5: ZSRO Open Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSRO: Read Only state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Open Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status 'Invalid Zone State Transition' BFh.

Case 6: ZSO Open Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSO: Offline state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of OpenZone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status 'Invalid Zone State Transition' BFh.

Case 7: ZSF Open Zone Select All=0 (M)

Test Procedure:

1. Configure the Testing Station to put a Zone on the DUT into the ZSF: Full state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Open Zone and Select All=0 to the Zone in the ZSF state.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status 'Invalid Zone State Transition' BFh

©2021 UNH–IOL

Case 8: ZSC Open Zone Select All=1 valid SLBA (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSC: Closed state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Open Zone and Select All=1, and a valid SLBA.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSEO: Explicitly Opened State. Verify that each Zone Management Send command completes with Status Success.
2. Alternately, if the operation causes the number of Open Resources to exceed the value specified by the Maximum Open Resources field, then the command shall be aborted with a status code of Too Many Open Zones, and no zone state transitions shall occur.
3. Verify that the SLBA field is ignored and the Zone Management Send command completes according to Select All =1.

Case 9: ZSC Open Zone Select All=1 invalid SLBA (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSC: Closed state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Open Zone and Select All=1, and an invalid SLBA, which should be ignored since the Select All bit is set to 1.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSEO: Explicitly Opened State. Verify that each Zone Management Send command completes with Status Success.
2. Alternately, if the operation causes the number of Open Resources to exceed the value specified by the Maximum Open Resources field, then the command shall be aborted with a status code of Too Many Open Zones, and no zone state transitions shall occur.
3. Verify that the SLBA field is ignored and the Zone Management Send command completes according to Select All =1.

Case 10: Explicitly Open Zone Select All=0 valid SLBA Exceed Open Resources (FYI)

Test Procedure:

1. Configure the Testing Station to issue an Identify Command for CNS 05h, I/O Command Set Specific Identify Namespace Data Structures and record the Maximum Open Resources value (MOR). If the MOR field is set to FFFFFFFFh this test is not applicable.
2. Configure the Testing Station to put the DUT into the ZSE state using a Zone Management Send command with Action of Reset.
3. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
4. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Open Zone and Select All=0, and a valid SLBA. Repeat this step for different zones until the number of open zones exceeds the MOR value.

Observable Results:

©2021 UNH–IOL

1. If MAR is greater than MOR, verify that the Zone Management Send command which causes the number of Open Resources to exceed the value specified by the Maximum Open Resources field is aborted with a status code of Too Many Open Zones BEh.
2. If MAR is equal to MOR, verify that the Zone Management Send command which causes the number of Open Resources to exceed the value specified by the Maximum Open Resources field is aborted with a status code of Too Many Active Zones BDh.

Possible Problems: None.

©2021 UNH–IOL

Test 2.3 – Zone Management Send: Reset Zone (FYI)

Purpose: To verify that an NVMe ZNS capable controller can properly process a Zoned Management Send command.

References:

[1] NVMe ZNS Specification 4.3.1.4

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: January 7, 2021

Discussion: The Zone Management Send command requests an action on one or more zones. The command uses the Data Pointer, Command Dword 10, Command Dword 11 and Command Dword 13 fields. All other command specific fields are reserved. The Zoned Managemetn Send command supports the following actions: Close Zone, Finish Zone, Open Zone, Reset Zone, Offline Zone, Set Zone Descriptor Extension.

Test Setup: See Appendix A.

Case 1: ZSIO Reset Zone Select All=0 (M)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSIO: Implicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Reset Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSE: Empty State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 2: ZSEO Reset Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSEO: Explicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Reset Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSE: Empty State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 3: ZSC Reset Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSC: Closed state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Reset Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

©2021 UNH–IOL

1. Verify that the current Zone State at the end of the test is ZSE: Empty State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 4: ZSF Reset Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSF: Full state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Reset Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSE: Empty State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 5: ZSE Reset Zone Select All=0 (M)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSE: Empty state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Reset Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSE: Empty State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 6: ZSRO Reset Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSRO: Read Only state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Reset Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status ‘Invalid Zone State Transition’ BFh.

Case 7: ZSO Reset Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSO: Offline state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Reset Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status ‘Invalid Zone State Transition’ BFh.

©2021 UNH–IOL

Case 8: ZSIO Reset Zone Select All=1 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSIO: Implicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Reset Zone and Select All=1, and a valid SLBA.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSE: Empty State.
2. Verify that the final Zone Descriptor returned has set the Write Pointer zone attribute to the ZSLBA of the zone.
3. Verify that the final Zone Descriptor returned has the following zone attribute bits cleared to 0:
    a. Zone Descriptor Extension Valid
    b. Finish Zone Recommended
    c. Reset Zone Recommended
    d. Zone Finished by Controller
4. Verify that each Zone Management Send command completes with Status Success.
5. Verify that the SLBA field is ignored and the Zone Management Send command completes according to Select All =1.

Case 9: ZSEO Reset Zone Select All=1 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSEO: Explicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Reset Zone and Select All=1, and a valid SLBA.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSE: Empty State.
2. Verify that the final Zone Descriptor returned has set the Write Pointer zone attribute to the ZSLBA of the zone.
3. Verify that the final Zone Descriptor returned has the following zone attribute bits cleared to 0:
    a. Zone Descriptor Extension Valid
    b. Finish Zone Recommended
    c. Reset Zone Recommended
    d. Zone Finished by Controller
4. Verify that each Zone Management Send command completes with Status Success.
5. Verify that the SLBA field is ignored and the Zone Management Send command completes according to Select All =1.

Case 10: ZSC Reset Zone Select All=1 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSC: Closed state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Reset Zone and Select All=1, and a valid SLBA.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

©2021 UNH–IOL

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSE: Empty State.
2. Verify that the final Zone Descriptor returned has set the Write Pointer zone attribute to the ZSLBA of the zone.
3. Verify that the final Zone Descriptor returned has the following zone attribute bits cleared to 0:
    a. Zone Descriptor Extension Valid
    b. Finish Zone Recommended
    c. Reset Zone Recommended
    d. Zone Finished by Controller
4. Verify that each Zone Management Send command completes with Status Success.
5. Verify that the SLBA field is ignored and the Zone Management Send command completes according to Select All =1.

Case 11: ZSF Reset Zone Select All=1 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSF: Full state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Reset Zone and Select All=1, and a valid SLBA.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSE: Empty State.
2. Verify that the final Zone Descriptor returned has set the Write Pointer zone attribute to the ZSLBA of the zone.
3. Verify that the final Zone Descriptor returned has the following zone attribute bits cleared to 0:
    a. Zone Descriptor Extension Valid
    b. Finish Zone Recommended
    c. Reset Zone Recommended
    d. Zone Finished by Controller
4. Verify that each Zone Management Send command completes with Status Success.
5. Verify that the SLBA field is ignored and the Zone Management Send command completes according to Select All =1.

Possible Problems: None.

©2021 UNH–IOL

Test 2.4 – Zone Management Send: Offline Zone (FYI)

Purpose: To verify that an NVMe ZNS capable controller can properly process a Zoned Management Send command.

References:

[1] NVMe ZNS Specification 4.3.1.4

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: January 7, 2021

Discussion: The Zone Management Send command requests an action on one or more zones. The command uses the Data Pointer, Command Dword 10, Command Dword 11 and Command Dword 13 fields. All other command specific fields are reserved. The Zoned Managemetn Send command supports the following actions: Close Zone, Finish Zone, Open Zone, Reset Zone, Offline Zone, Set Zone Descriptor Extension.

Test Setup: See Appendix A.

Case 1: ZSRO Offline Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSRO: Read Only state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Offline Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSO: Offline State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 2: ZSO Offline Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSO: Offline state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Offline Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSO: Offline State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 3: ZSIO Offline Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSIO: Implicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Offline Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

©2021 UNH–IOL

Observable Results:
1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status 'Invalid Zone State Transition' BFh.

Case 4: ZSEO Offline Zone Select All=0 (FYI)

Test Procedure:
1. Configure the Testing Station to put the DUT into the ZSEO: Explicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Offline Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:
1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status 'Invalid Zone State Transition' BFh.

Case 5: ZSC Offline Zone Select All=0 (FYI)

Test Procedure:
1. Configure the Testing Station to put the DUT into the ZSC: Closed state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Offline Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:
1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status 'Invalid Zone State Transition' BFh.

Case 6: ZSF Offline Zone Select All=0 (FYI)

Test Procedure:
1. Configure the Testing Station to put the DUT into the ZSF: Full state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Offline Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:
1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status 'Invalid Zone State Transition' BFh.

Case 7: ZSE Offline Zone Select All=0 (FYI)

Test Procedure:
1. Configure the Testing Station to put the DUT into the ZSE: Empty state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Offline Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:
1. Verify that the current Zone State between steps 2 and 4 is unchanged.

©2021 UNH–IOL

2. Verify that the Zone Management Send command completes with Status ‘Invalid Zone State Transition’ BFh.

Possible Problems: None.

©2021 UNH–IOL

Test 2.5 – Zone Management Send: Set Zone Descriptor Extension (FYI)

Purpose: To verify that an NVMe ZNS capable controller can properly process a Zoned Management Send command.

References:

[1] NVMe ZNS Specification 4.3.1.4

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: January 7, 2021

Discussion: The Zone Management Send command requests an action on one or more zones. The command uses the Data Pointer, Command Dword 10, Command Dword 11 and Command Dword 13 fields. All other command specific fields are reserved. The Zoned Managemetn Send command supports the following actions: Close Zone, Finish Zone, Open Zone, Reset Zone, Offline Zone, Set Zone Descriptor Extension.

Test Setup: See Appendix A.

Case 1: ZSE Set Zone Descriptor Extension Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSE: Empty state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Set Zone Descriptor Extension and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSC: Closed State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 2: ZSRO Set Zone Descriptor Extension Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSRO: Read Only state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Set Zone Descriptor Extension Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status ‘Invalid Zone State Transition’ BFh.

Case 3: ZSO Set Zone Descriptor Extension Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSO: Offline state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Set Zone Descriptor Extension Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

©2021 UNH–IOL

1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status 'Invalid Zone State Transition' BFh.

Case 4: ZSIO Set Zone Descriptor Extension Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSIO: Implicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Set Zone Descriptor Extension Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status 'Invalid Zone State Transition' BFh.

Case 5: ZSEO Set Zone Descriptor Extension Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSEO: Explicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Set Zone Descriptor Extension Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status 'Invalid Zone State Transition' BFh.

Case 6: ZSC Set Zone Descriptor Extension Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSC: Closed state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Set Zone Descriptor Extension Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status 'Invalid Zone State Transition' BFh.

Case 7: ZSF Set Zone Descriptor Extension Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSF: Full state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Set Zone Descriptor Extension Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status 'Invalid Zone State Transition' BFh.

©2021 UNH–IOL

Case 8: All States Set Zone Descriptor Extension Zone Select All=1 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSF: Full state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Set Zone Descriptor Extension Zone and Select All=1.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
5. Repeat for the following starting states: ZSIO, ZSEO, ZSRO, ZSC, ZSE, ZSO.

Observable Results:

1. Verify that the current Zone State between steps 2 and 4 is unchanged.
2. Verify that the Zone Management Send command completes with Status 'Invalid Field in Command'.

Case 9: Zone Descriptor Extension Size = 0 (FYI)

Test Procedure:

1. Perform an Identify Command for CNS=05h and check the Zone Descriptor Extension Field Size value in the Identify Namespace Data Structure. If this is not set to 0h this test is not applicable.
2. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Set Zone Descriptor Extension Zone and Select All=0.

Observable Results:

1. Verify that the Zone Management Send command completes with Status 'Invalid Field in Command'.

Case 10: Closed Zone Exceeds Active Resources (FYI)

Test Procedure:

1. Configure the Testing Station to issue an Identify Command for CNS 05h, I/O Command Set Specific Identify Namespace Data Structures and record the Maximum Active Resources value (MAR) and the Maximum Open Resources value (MOR). If the MAR field is set to FFFFFFFFh this test is not applicable.
2. Configure the Testing Station to put the DUT into the ZSE state using a Zone Management Send command with Action of Reset.
3. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
4. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Open Zone and Select All=1, and a valid SLBA to a different Zone than the Zone that was Reset in the previous step. Repeat this step for different zones until the number of open zones is 1 less than the MAR value. Ensure that one zone remains in the ZSE state.
5. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Set Zone Descriptor Extension to the Zone in the ZSE state. This should cause the number of active zones to exceed the MAR value.

Observable Results:

1. Verify that the Zone Management Send command with Action Set Zone Descriptor Extension which causes the number of Active Resources to exceed the value specified by the Maximum Active Resources field is aborted with a status code of Too Many Active Zones, and no zone state transition occurs.

Possible Problems: None.

©2021 UNH–IOL

Test 2.6 – Zone Management Receive: Report Zones (FYI)

Purpose: To verify that an NVMe ZNS capable controller can properly process a Zoned Management Receive command.

References:

[1] NVMe ZNS Specification 4.4

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: January 21, 2021

Discussion: The Zone Management Receive command returns a data buffer that contains information about zones.

Test Setup: See Appendix A.

Case 1: Report Zones: All Zones, Partial Report = 0 (M)

Test Procedure:

1. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 0h List all zones

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 2: Report Zones: All Zones, Partial Report = 1 (M)

Test Procedure:

1. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 0h List all zones

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 3: Report Zones: ZSE, Partial Report = 0 (M)

Test Procedure:

1. Configure the Testing Station to put the DUT in to the ZSE state.

©2021 UNH–IOL

2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 1h List the zones in the ZSE: Empty State

Observable Results:
1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 4: Report Zones: ZSE, Partial Report = 1 (M)

Test Procedure:
1. Configure the Testing Station to put the DUT in to the ZSE state.
2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 1h List the zones in the ZSE: Empty State

Observable Results:
1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 5: Report Zones: ZSIO, Partial Report = 0 (M)

Test Procedure:
1. Configure the Testing Station to put the DUT in to the ZSIO state.
2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 2h List the zones in the ZSIO: Implicitly Opened State

Observable Results:
1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

©2021 UNH–IOL

Case 6: Report Zones: ZSIO, Partial Report = 1 (M)

Test Procedure:

1. Configure the Testing Station to put the DUT in to the ZSIO state.
2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 2h List the zones in the ZSIO: Implicitly Opened State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 7: Report Zones: ZSEO, Partial Report = 0 (M)

Test Procedure:

1. Configure the Testing Station to put the DUT in to the ZSEO state.
2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 3h List the zones in the ZSEO: Explicitly Opened State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 8: Report Zones: ZSEO, Partial Report = 1 (M)

Test Procedure:

1. Configure the Testing Station to put the DUT in to the ZSEO state.
2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 3h List the zones in the ZSEO: Explicitly Opened State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.

©2021 UNH–IOL

4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 9: Report Zones: ZSC, Partial Report = 0 (M)

Test Procedure:

1. Configure the Testing Station to put the DUT in to the ZSC state.
2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 4h List the zones in the ZSC: Closed state

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 10: Report Zones: ZSC, Partial Report = 1 (M)

Test Procedure:

1. Configure the Testing Station to put the DUT in to the ZSC state.
2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 4h List the zones in the ZSC: Closed state

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 11: Report Zones: ZSF, Partial Report = 0 (M)

Test Procedure:

1. Configure the Testing Station to put the DUT in to the ZSF state.
2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 5h List the zones in the ZSF: Full State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.

©2021 UNH–IOL

2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 12: Report Zones: ZSF, Partial Report = 1 (M)

Test Procedure:

1. Configure the Testing Station to put the DUT in to the ZSF state.
2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 5h List the zones in the ZSF: Full State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 13: Report Zones: ZSRO, Partial Report = 0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT in to the ZSRO state.
2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 6h List the zones in the ZSRO: Ready Only State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 14: Report Zones: ZSRO, Partial Report = 1 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT in to the ZSRO state.
2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 6h List the zones in the ZSRO: Ready Only State

©2021 UNH–IOL

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 15: Report Zones: ZSO, Partial Report = 0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT in to the ZSO state.
2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 7h List the zones in the ZSO: Offline State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 16: Report Zones: ZSO, Partial Report = 1 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT in to the ZSO state.
2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 7h List the zones in the ZSO: Offline State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 17: Report Zones: Compare ZCAP and ZSZE (FYI)

Test Procedure:

1. Configure the Testing Station to transmit an Identify Command with CNS=05h. Record the ZSZE field in the LBA Format Extension Data Structure.

©2021 UNH–IOL

2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Report Zones, with the Partial Report bit set to 0 and Zone Receive Action Specific Field: 0h List all zones. Record the ZCAP field.

Observable Results:

1. Verify that the ZCAP field value is less than or equal to the ZSZE field value.

Possible Problems: None.

©2021 UNH–IOL

Test 2.7 – Zone Management Receive: Extended Report Zones Correct Format (FYI)

Purpose: To verify that an NVMe ZNS capable controller can properly process a Zoned Management Receive command.

References:

[1] NVMe ZNS Specification 4.4

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: January 21, 2021

Discussion: The Zone Management Receive command returns a data buffer that contains information about zones.

Test Setup: See Appendix A.

Case 1: Extended Report Zones: All Zones, Partial Report = 0 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a non-zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 0h List all zones

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 2: Extended Report Zones: All Zones, Partial Report = 1 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a non-zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 0h List all zones

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

©2021 UNH–IOL

Case 3: Extended Report Zones: ZSE, Partial Report = 0 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a non-zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSE state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 1h List the zones in the ZSE: Empty State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 4: Extended Report Zones: ZSE, Partial Report = 1 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a non-zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSE state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 1h List the zones in the ZSE: Empty State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 5: Extended Report Zones: ZSIO, Partial Report = 0 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a non-zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSIO state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:

©2021 UNH–IOL

a. 2h List the zones in the ZSIO: Implicitly Opened State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 6: Extended Report Zones: ZSIO, Partial Report = 1 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a non-zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSIO state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 2h List the zones in the ZSIO: Implicitly Opened State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 7: Extended Report Zones: ZSEO, Partial Report = 0 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a non-zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSEO state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 3h List the zones in the ZSEO: Explicitly Opened State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.

©2021 UNH–IOL

4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 8: Extended Report Zones: ZSEO, Partial Report = 1 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a non-zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSEO state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 3h List the zones in the ZSEO: Explicitly Opened State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 9: Extended Report Zones: ZSC, Partial Report = 0 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a non-zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSC state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 4h List the zones in the ZSC: Closed state

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 10: Extended Report Zones: ZSC, Partial Report = 1 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a non-zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSC state.

©2021 UNH–IOL

3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 4h List the zones in the ZSC: Closed state

Observable Results:
1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 11: Extended Report Zones: ZSF, Partial Report = 0 (FYI)

Test Procedure:
1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a non-zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSF state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 5h List the zones in the ZSF: Full State

Observable Results:
1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 12: Extended Report Zones: ZSF, Partial Report = 1 (FYI)

Test Procedure:
1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a non-zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSF state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 5h List the zones in the ZSF: Full State

Observable Results:
1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.

©2021 UNH–IOL

3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 13: Extended Report Zones: ZSRO, Partial Report = 0 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a non-zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSRO state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 6h List the zones in the ZSRO: Ready Only State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 14: Extended Report Zones: ZSRO, Partial Report = 1 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a non-zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSRO state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 6h List the zones in the ZSRO: Ready Only State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 15: Extended Report Zones: ZSO, Partial Report = 0 (FYI)

Test Procedure:

©2021 UNH–IOL

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a non-zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSO state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 7h List the zones in the ZSO: Offline State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Case 16: Extended Report Zones: ZSO, Partial Report = 1 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a non-zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSO state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 7h List the zones in the ZSO: Offline State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Success.
2. Verify that the DUT only returns zone descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the Zone Management Receive command.
3. Verify that the DUT only returns zone descriptors which match the criteria in the Zone Receive Action Specific field.
4. Verify that the zone descriptors returned by the DUT are sorted in ascending order by the ZSLBA value of each zone.

Possible Problems: None.

©2021 UNH–IOL

Test 2.8 – Zone Management Receive: Extended Report Zones Incorrect Format (FYI)

Purpose: To verify that an NVMe ZNS capable controller can properly process a Zoned Management Receive command.

References:

[1] NVMe ZNS Specification 4.4

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: January 21, 2021

Discussion: The Zone Management Receive command returns a data buffer that contains information about zones.

Test Setup: See Appendix A.

Case 1: Extended Report Zones: All Zones, Partial Report = 0 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 0h List all zones

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Invalid Field in Command.

Case 2: Extended Report Zones: All Zones, Partial Report = 1 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 0h List all zones

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Invalid Field in Command.

Case 3: Extended Report Zones: ZSE, Partial Report = 0 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSE state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:

©2021 UNH–IOL

a. 1h List the zones in the ZSE: Empty State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Invalid Field in Command.

Case 4: Extended Report Zones: ZSE, Partial Report = 1 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSE state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 1h List the zones in the ZSE: Empty State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Invalid Field in Command.

Case 5: Extended Report Zones: ZSIO, Partial Report = 0 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSIO state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 2h List the zones in the ZSIO: Implicitly Opened State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Invalid Field in Command.

Case 6: Extended Report Zones: ZSIO, Partial Report = 1 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSIO state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 2h List the zones in the ZSIO: Implicitly Opened State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Invalid Field in Command.

Case 7: Extended Report Zones: ZSEO, Partial Report = 0 (FYI)

Test Procedure:

©2021 UNH–IOL

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSEO state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 3h List the zones in the ZSEO: Explicitly Opened State

Observable Results:
1. Verify that each Zone Management Receive command completes with Status Invalid Field in Command.

Case 8: Extended Report Zones: ZSEO, Partial Report = 1 (FYI)

Test Procedure:
1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSEO state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 3h List the zones in the ZSEO: Explicitly Opened State

Observable Results:
1. Verify that each Zone Management Receive command completes with Status Invalid Field in Command.

Case 9: Extended Report Zones: ZSC, Partial Report = 0 (FYI)

Test Procedure:
1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSC state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 4h List the zones in the ZSC: Closed state

Observable Results:
1. Verify that each Zone Management Receive command completes with Status Invalid Field in Command.

Case 10: Extended Report Zones: ZSC, Partial Report = 1 (FYI)

Test Procedure:
1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSC state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 4h List the zones in the ZSC: Closed state

©2021 UNH–IOL

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Invalid Field in Command.

Case 11: Extended Report Zones: ZSF, Partial Report = 0 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSF state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 5h List the zones in the ZSF: Full State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Invalid Field in Command.

Case 12: Extended Report Zones: ZSF, Partial Report = 1 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSF state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 5h List the zones in the ZSF: Full State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Invalid Field in Command.

Case 13: Extended Report Zones: ZSRO, Partial Report = 0 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSRO state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 6h List the zones in the ZSRO: Ready Only State

Observable Results:

1. Verify that each Zone Management Receive command completes with Status Invalid Field in Command.

Case 14: Extended Report Zones: ZSRO, Partial Report = 1 (FYI)

Test Procedure:

1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.

©2021 UNH–IOL

2. Configure the Testing Station to put the DUT in to the ZSRO state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 6h List the zones in the ZSRO: Ready Only State

Observable Results:
1. Verify that each Zone Management Receive command completes with Status Invalid Field in Command.

Case 15: Extended Report Zones: ZSO, Partial Report = 0 (FYI)

Test Procedure:
1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSO state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 0. Repeat the command for the following Zone Receive Action Specific Field:
    a. 7h List the zones in the ZSO: Offline State

Observable Results:
1. Verify that each Zone Management Receive command completes with Status Invalid Field in Command.

Case 16: Extended Report Zones: ZSO, Partial Report = 1 (FYI)

Test Procedure:
1. Configure the Testing Station to transmit a Format NVM command to the DUT to format a given namespace with a zero Zone Descriptor Extension Size. The following steps in the test procedure should be performed on this same namespace.
2. Configure the Testing Station to put the DUT in to the ZSO state.
3. Configure the Testing Station to transmit a Zone Management Receive command, with action set to Extended Report Zones, with the Partial Report bit set to 1. Repeat the command for the following Zone Receive Action Specific Field:
    a. 7h List the zones in the ZSO: Offline State

Observable Results:
1. Verify that each Zone Management Receive command completes with Status Invalid Field in Command.

Possible Problems: None.

©2021 UNH–IOL

Test 2.9 – Zone Append (FYI)

Purpose: To verify that an NVMe ZNS capable controller can properly process a Zoned Append command.

References:

[1] NVMe ZNS Specification 4.5

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: January 21, 2021

Discussion: The Zone Append command writes data and metadata, if applicable, to the I/O controller for the zone indicated by the ZSLBA field.

Test Setup: See Appendix A.

Case 1: Zone Append (M)

Test Procedure:

1. Determine support for the Zone Append Command by performing the following steps. If the Zone Append command is not support this test is not applicable.
    a. Configure the Testing Station to set CC.CSS to indicate support for ZNS.
    b. Configure the Testing Station to perform a Get Log Page command for LID=05h, Commands Supported and Effects.
    c. Check the CSUPP bit in IOCS125 of the returned log page. IOCS125 corresponds to support for the Zone Append command. The CSUPP bit indicates if this command is supported or not.
2. Configure the Testing Station to transmit a Zone Append command, with valid ZSLBA and PRINFO fields.

Observable Results:

1. Verify that the Zone Append command completes successfully.

Case 2: ZSE Zone Append Incorrect Logical Block (M)

Test Procedure:

1. Follow the first step in Case 1 above to determine support for the Zone Append Command. If the Zone Append command is not support this test is not applicable.
2. Configure the DUT to enter the ZSE: Empty state.
3. Configure the Testing Station to transmit a Zone Append command with a ZSLBA value which does not specify the lowest logical block for the zone.

Observable Results:

1. Verify that each Zone Append command completes with Status Invalid Field in Command.

Case 3: ZSIO Zone Append Incorrect Logical Block (M)

Test Procedure:

1. Follow the first step in Case 1 above to determine support for the Zone Append Command. If the Zone Append command is not support this test is not applicable.
2. Configure the DUT to enter the ZSIO: Implicitly Opened state.
3. Configure the Testing Station to transmit a Zone Append command with a ZSLBA value which does not specify the lowest logical block for the zone.

Observable Results:

1. Verify that each Zone Append command completes with Status Invalid Field in Command.

©2021 UNH–IOL

Case 4: ZSEO Zone Append Incorrect Logical Block (M)

Test Procedure:

1. Follow the first step in Case 1 above to determine support for the Zone Append Command. If the Zone Append command is not support this test is not applicable.
2. Configure the DUT to enter the ZSEO: Explicitly Opened state.
3. Configure the Testing Station to transmit a Zone Append command with a ZSLBA value which does not specify the lowest logical block for the zone.

Observable Results:

1. Verify that each Zone Append command completes with Status Invalid Field in Command.

Case 5: ZSC Zone Append Incorrect Logical Block (M)

Test Procedure:

1. Follow the first step in Case 1 above to determine support for the Zone Append Command. If the Zone Append command is not support this test is not applicable.
2. Configure the DUT to enter the ZSC: Closed state.
3. Configure the Testing Station to transmit a Zone Append command with a ZSLBA value which does not specify the lowest logical block for the zone.

Observable Results:

1. Verify that each Zone Append command completes with Status Invalid Field in Command.

Case 6: Zone Append Incorrect PIREMAP for Type 1 Protection (FYI)

Test Procedure:

1. Follow the first step in Case 1 above to determine support for the Zone Append Command. If the Zone Append command is not support this test is not applicable.
2. Check the DPC field to determine the Protection Information types supported by the DUT. If the DUT does not support end to end data protection Type 1, then this case is not applicable.
3. Configure a zone on the DUT to be formatted with Type 1 protection if supported.
4. Configure the Testing Station to transmit a Zone Append command with a PIREMAP value cleared to 0 for Type 1 protection.

Observable Results:

1. Verify that each Zone Append command completes with Status Invalid Protection Information.

Case 7: Zone Append Incorrect PIREMAP for Type 3 Protection (FYI)

Test Procedure:

1. Follow the first step in Case 1 above to determine support for the Zone Append Command. If the Zone Append command is not support this test is not applicable.
2. Check the DPC field to determine the Protection Information types supported by the DUT. If the DUT does not support end to end data protection Type 3, then this case is not applicable.
3. Configure a zone on the DUT to be formatted with Type 3 protection.
4. Configure the Testing Station to transmit a Zone Append command with a PIREMAP value set to 1 for Type 3 protection.

Observable Results:

1. Verify that each Zone Append command completes with Status Invalid Protection Information.

Case 8: Zone Append to Zone in ZSF State (FYI)

©2021 UNH–IOL

Test Procedure:

1. Follow the first step in Case 1 above to determine support for the Zone Append Command. If the Zone Append command is not supported this test is not applicable.
2. Configure a zone on the DUT to be in the ZSF state.
3. Configure the Testing Station to transmit a Zone Append command to the Zone in the ZSF state.

Observable Results:

1. Verify that the Zone Append command completes with Status “Zone is Full” B9h.

Possible Problems: None.

©2021 UNH–IOL

Test 2.10 – Zone State Change due to Write Operation (FYI)

Purpose: To verify that an NVMe ZNS capable controller properly changes states due to write operations.

References:

[1] NVMe ZNS Specification 2.3.11

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: January 26, 2021

Discussion: The write pointer for a zone in the ZSE:Empty state, the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, or the ZSC:Closed state shall be increased by the number of logical blocks written on successful completion of a write operation.

Test Setup: See Appendix A.

Case 1: ZSE to ZSIO (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSE: Empty state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send write operation to the zone.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
5. Configure the Testing Station acting as a Host to send write operation to the zone.
6. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State after the first write operation is ZSIO: Implicitly Opened.
2. Verify that the write pointer reported for the zone in the second Zone Descriptor returned has increased by the number of logical blocks written on successful completion of the write operation relative to the write pointer in the first returned Zone Descriptor.

Case 2: ZSC to ZSIO (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSC: Closed state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send write operation to the zone.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
5. Configure the Testing Station acting as a Host to send write operation to the zone.
6. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State after the first write operation is ZSIO: Implicitly Opened.
2. Verify that the write pointer reported for the zone in the second Zone Descriptor returned has increased by the number of logical blocks written on successful completion of the write operation relative to the write pointer in the first returned Zone Descriptor.

Possible Problems: None.

©2021 UNH–IOL

Test 2.11 – Failed Write Operation (FYI)

Purpose: To verify that an NVMe ZNS capable controller properly handles incorrect write operations.

References:

[1] NVMe ZNS Specification 2.3.11

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: January 26, 2021

Discussion: The ZNS specification identifies certain error codes to be returned when a write operation is performed on an offline or read only zone.

Test Setup: See Appendix A.

Case 1: ZSRO (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSRO: Read Only state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send write operation to the zone.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the attempted write operation does not change the current Zone state.
2. Verify that the write operation is aborted with status Zone is Read Only (BAh).

Case 2: ZSO (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSO: Offline state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send write operation to the zone.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the attempted write operation does not change the current Zone state.
2. Verify that the write operation is aborted with status Zone is Offline (BBh).

Case 3: ZSE Write Bad SLBA (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSE: Empty state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send write operation to the zone specifying an SLBA which is not equal to the write pointer for that zone
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the attempted write operation does not change the current Zone state.
2. Verify that the write operation is aborted with status Zone Invalid Write (BCh).

©2021 UNH–IOL

Case 4: ZSIO Write Bad SLBA (FYI)

Test Procedure:
1. Configure the Testing Station to put the DUT into the ZSIO: Implicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send write operation to the zone specifying an SLBA which is not equal to the write pointer for that zone
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:
1. Verify that the attempted write operation does not change the current Zone state.
2. Verify that the write operation is aborted with status Zone Invalid Write (BCh).

Case 5: ZSEO Write Bad SLBA (FYI)

Test Procedure:
1. Configure the Testing Station to put the DUT into the ZSEO: Explicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send write operation to the zone specifying an SLBA which is not equal to the write pointer for that zone
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:
1. Verify that the attempted write operation does not change the current Zone state.
2. Verify that the write operation is aborted with status Zone Invalid Write (BCh).

Case 6: ZSC Write Bad SLBA (FYI)

Test Procedure:
1. Configure the Testing Station to put the DUT into the ZSC: Closed state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send write operation to the zone specifying an SLBA which is not equal to the write pointer for that zone
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:
1. Verify that the attempted write operation does not change the current Zone state.
2. Verify that the write operation is aborted with status Zone Invalid Write (BCh).

Case 7: NLB Exceeds Remaining Blocks (FYI)

Test Procedure:
1. Configure the Testing Station acting as a Host to send write operation to the zone specifying a Number of Logical blocks that exceeds the remaining number of logical blocks in that zone.

Observable Results:
1. Verify that the attempted write operation does not change the current Zone state.
2. Verify that the write operation is aborted with status Zone Boundary Error (BCh).

Possible Problems: None.

©2021 UNH–IOL

Test 2.12 – Zoned Reads (FYI)

Purpose: To verify that an NVMe ZNS capable controller properly handles Read operations.

References:
[1] NVMe ZNS Specification 2.3.1.2

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: January 26, 2021

Discussion: The ZNS specification identifies certain conditions for reading zones.

Test Setup: See Appendix A.

Case 1: Read Across Zone Boundaries = 1 (M)

Test Procedure:
1. Configure the Testing Station to send an Identify Command for CNS 05h. Check the Check the Read Across Zone Boundaries bit. If this bit is set to 0, this test is not applicable.
2. Perform a Read operation with an SLBA and Length which will span zone boundaries.

Observable Results:
1. Verify that the Read operation completes successfully.

Case 2: Read Across Zone Boundaries = 0 (FYI)

Test Procedure:
1. Configure the Testing Station to send an Identify Command for CNS 05h. Check the Check the Read Across Zone Boundaries bit. If this bit is set to 1, this test is not applicable.
2. Perform a Read operation with an SLBA and Length which will span zone boundaries.

Observable Results:
1. Verify that the Read operation is aborted with status Zone Boundary Error (B8h).

Case 3: Read Beyond Write Pointer when DULBE=1 (FYI)

Test Procedure:
1. The Testing Station should issue a Set Feature Command for the Error Recovery feature (FID=05h) to set DULBE to 1.
2. Configure the Testing Station to transition a Zone to the Explicitly Opened State. Perform a Write operation to that Zone.
3. Check the write pointer value.
4. If the write pointer is valid, perform a Read operation to the same Zone for a logical block beyond the write pointer value.

Observable Results:
1. Verify that the Read operation is aborted with status ‘Deallocated or Unwritten Logical Block’ (87h).

Possible Problems: None.

©2021 UNH–IOL

Test 2.13 – Logical Block Allocation (FYI)

Purpose: To verify that an NVMe ZNS capable controller properly marks blocks as allocated.

References:

[1] NVMe ZNS Specification 2.6

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: June 8, 2021

Discussion: The ZNS specification identifies certain requirements for marking blocks as allocated.

Test Setup: See Appendix A.

Case 1: Logical Block Allocated after Write (FYI)

Test Procedure:

1. Configure the Testing Station to perform a Write operation to a currently unallocated block.
2. Read that logical block and check whether it is marked as allocated or not.

Observable Results:

1. Verify that the Logical Block is marked as allocated after the Write operation completes successfully.

Case 2: Logical Block Allocated after Write Uncorrectable (FYI)

Test Procedure:

1. Configure the Testing Station to perform a Write Uncorrectable operation to a currently unallocated block.
2. Read that logical block and check whether it is marked as allocated or not.
3. If the DUT does not support the Write Uncorrectable command this test is not applicable.

Observable Results:

1. Verify that the Logical Block is marked as allocated after the Write Uncorrectable operation completes successfully.

Case 3: Logical Block Allocated after Write Zeroes (FYI)

Test Procedure:

1. Configure the Testing Station to perform a Write Zeroes operation which does not deallocate the logical block, to a currently unallocated block.
2. Read that logical block and check whether it is marked as allocated or not.
3. If the DUT does not support the Write Zeroes command this test is not applicable.

Observable Results:

1. Verify that the Logical Block is marked as allocated after the Write Zeroes operation completes successfully.

Case 4: Logical Block Allocated after Copy (FYI)

Test Procedure:

1. Configure the Testing Station to perform a Copy operation to a currently unallocated block.
2. Read that logical block and check whether it is marked as allocated or not.
3. If the DUT does not support the Copy command this test is not applicable.

Observable Results:

©2021 UNH–IOL

1. Verify that the Logical Block is marked as allocated after the Copy operation completes successfully.

Case 5: Logical Block Allocated after Zone Append (FYI)

Test Procedure:

1. Configure the Testing Station to perform a Zone Append operation to a currently unallocated block.
2. Read that logical block and check whether it is marked as allocated or not.
3. If the DUT does not support the Zone Append command this test is not applicable.

Observable Results:

1. Verify that the Logical Block is marked as allocated after the Zone Append operation completes successfully.

Case 6: Logical Block Deallocated after Zone Reset (FYI)

Test Procedure:

1. Configure the a Zone to be in the ZSIO or ZSEO state.
2. Perform a Zone Management Send command with action of Reset Zone.

Observable Results:

1. Verify that all the Logical Blocks in the zone are marked as deallocated.

Possible Problems: None.

©2021 UNH–IOL

Test 2.14 – Sanitize (FYI)

Purpose: To verify that an NVMe ZNS capable controller properly handles the Sanitize command.

References:

[1] NVMe ZNS Specification 3.5.2

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: June 8, 2021

Discussion: The ZNS specification identifies certain requirements for the Sanitize command
Test Setup: See Appendix A.

Case 1: Sanitize NODMMAS = 01b from ZSIO (FYI)

Test Procedure:

1. Configure a Zone on the DUT to be in the Zone Implicitly Opened State.
2. Configure the Testing Station to perform a Sanitize operation to that Zone with the following attributes:
    a. No Deallocate After Sanitize = 1b
    b. NODMMAS = 01b
    c. NDI=0b
3. Perform a Read operation to that logical block.

Observable Results:

1. Verify that the Logical Block is written with the overwrite pattern from the Sanitize command.
2. Verify that the Sanitize Status of the Logical Block is 001b.
3. Verify that the Zone State of the Zone is ZSF:Full.

Case 2: Sanitize NODMMAS = 01b from ZSEO (FYI)

Test Procedure:

1. Configure a Zone on the DUT to be in the Zone Explicitly Opened State.
2. Configure the Testing Station to perform a Sanitize operation to that Zone with the following attributes:
    a. No Deallocate After Sanitize = 1b
    b. NODMMAS = 01b
    c. NDI=0b
3. Perform a Read operation to that logical block.

Observable Results:

1. Verify that the Logical Block is written with the overwrite pattern from the Sanitize command.
2. Verify that the Sanitize Status of the Logical Block is 001b.
3. Verify that the Zone State of the Zone is ZSF:Full.

Case 3: Sanitize NODMMAS = 10b from ZSIO (FYI)

Test Procedure:

1. Configure a Zone on the DUT to be in the Zone Implicitly Opened State.
2. Configure the Testing Station to perform a Sanitize operation to that Zone with the following attributes:
    a. No Deallocate After Sanitize = 1b
    b. NODMMAS = 10b
    c. NDI=0b
3. Perform a Read operation to that logical block.

©2021 UNH–IOL

Observable Results:

1. Verify that the Logical Block is written with the overwrite pattern from the Sanitize command.
2. Verify that the Sanitize Status of the Logical Block is 001b.
3. Verify that the Zone State of the Zone is ZSF:Full.

Case 4: Sanitize NODMMAS = 10b from ZSEO (FYI)

Test Procedure:

1. Configure a Zone on the DUT to be in the Zone Explicitly Opened State.
2. Configure the Testing Station to perform a Sanitize operation to that Zone with the following attributes:
    a. No Deallocate After Sanitize = 1b
    b. NODMMAS = 10b
    c. NDI=0b
3. Perform a Read operation to that logical block.

Observable Results:

1. Verify that the Logical Block is written with the overwrite pattern from the Sanitize command.
2. Verify that the Sanitize Status of the Logical Block is 001b.
3. Verify that the Zone State of the Zone is ZSF:Full.

Possible Problems: None.

©2021 UNH–IOL

Test 2.15 – Write Protection (FYI)

Purpose: To verify that an NVMe ZNS capable controller properly handles Write Protected Zones.

References:
[1] NVMe ZNS Specification 3.5.2

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: June 8, 2021

Discussion: The ZNS specification identifies certain requirements for the Sanitize command
Test Setup: See Appendix A.

Case 1: Enable Write Protection from ZSIO (FYI)

Test Procedure:
1. Configure a Zone on the DUT to be in the Zone Implicitly Opened State.
2. Perform a Get Feature Command for the Write Protection Feature (FID=84h) to verify that No Write Protection is enabled.
3. Configure the Testing Station to perform a Set Feature command to that Namespace to enable Write Protection.

Observable Results:
1. Verify that the Zone transitions to the ZSF:Full State when the Namespace becomes write protected.

Case 2: Enable Write Protection from ZSEO (FYI)

Test Procedure:
1. Configure a Zone on the DUT to be in the Zone Explicitly Opened State.
2. Perform a Get Feature Command for the Write Protection Feature (FID=84h) to verify that No Write Protection is enabled.
3. Configure the Testing Station to perform a Set Feature command to that Namespace to enable Write Protection.

Observable Results:
1. Verify that the Zone transitions to the ZSF:Full State when the Namespace becomes write protected.

Case 3: Zone Send to Protected Namespace from ZSIO (FYI)

Test Procedure:
1. Configure a Zone on the DUT to be in the Zone Implicitly Opened State.
2. Perform a Get Feature Command for the Write Protection Feature (FID=84h) to verify that No Write Protection is enabled.
3. Configure the Testing Station to perform a Set Feature command to that Namespace to enable Write Protection.
4. Perform a Zone Send command with action ‘Finish Zone’.
5. Perform a Zone Send command with action ‘Open Zone’.
6. Perform a Zone Send command with action ‘Reset Zone’.
7. Perform a Zone Send command with action ‘Offline Zone’.

Observable Results:
1. Verify that each of the Zone Send commands was aborted with status of ‘Namespace is Write Protected’.

©2021 UNH–IOL

Case 4: Zone Send to Protected Namespace from ZSEO (FYI)

Test Procedure:

1. Configure a Zone on the DUT to be in the Zone Explicitly Opened State.
2. Perform a Get Feature Command for the Write Protection Feature (FID=84h) to verify that No Write Protection is enabled.
3. Configure the Testing Station to perform a Set Feature command to that Namespace to enable Write Protection.
4. Perform a Zone Send command with action 'Finish Zone'.
5. Perform a Zone Send command with action 'Open Zone'.
6. Perform a Zone Send command with action 'Reset Zone'.
7. Perform a Zone Send command with action 'Offline Zone'.

Observable Results:

1. Verify that each of the Zone Send commands was aborted with status of 'Namespace is Write Protected'.

Possible Problems: None.

©2021 UNH–IOL

Test 2.16 – Zone Management Send: Close Zone (FYI)

Purpose: To verify that an NVMe ZNS capable controller can properly process a Zoned Management Send command.

References:

[1] NVMe ZNS Specification 4.3.1.1

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: June 10, 2021

Discussion: The Zone Management Send command requests an action on one or more zones. The command uses the Data Pointer, Command Dword 10, Command Dword 11 and Command Dword 13 fields. All other command specific fields are reserved. The Zoned Managemetn Send command supports the following actions: Close Zone, Finish Zone, Open Zone, Reset Zone, Offline Zone, Set Zone Descriptor Extension.

Test Setup: See Appendix A.

Case 1: ZSIO Close Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSIO: Implicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Close Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSC: Closed State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 2: ZSEO Close Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSEO: Explicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Close Zone and Select All=0.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSC: Closed State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 3: ZSIO Close Zone Select All=1 (FYI)

Test Procedure:

1. Configure the Testing Station to configure the DUT to have multiple states in the ZSIO: Implicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Close Zone and Select All=1, and a valid SLBA, and a valid SLBA.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

©2021 UNH–IOL

Observable Results:

1. Verify that the current Zone State for all Zones previously in the ZSIO state at the end of the test is ZSC: Closed State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 4: ZSEO Close Zone Select All=1 (FYI)

Test Procedure:

1. Configure the Testing Station to configure the DUT to have multiple states in the ZSEO: Explicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Close Zone and Select All=1, and a valid SLBA.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State for all Zones previously in the ZSEO state at the end of the test is ZSC: Closed State.
2. Verify that each Zone Management Send command completes with Status Success.

Case 5: ZSE Close Zone Select All=0 (FYI)

Test Procedure:

5. Configure the Testing Station to configure a zone to the ZSE: Empty state.
6. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
7. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Close Zone and Select All=0, and a valid SLBA.
8. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that each Zone Management Send command completes with Status ‘Invalid Zone State Transition’.

Case 6: ZSF Close Zone Select All=0 (FYI)

Test Procedure:

1. Configure the Testing Station to configure a zone to the ZSF: Full state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Close Zone and Select All=0, and a valid SLBA.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that each Zone Management Send command completes with Status ‘Invalid Zone State Transition’.

Possible Problems: None.

©2021 UNH–IOL

Test 2.17 – Zone Management Send: Incorrect SLBA (FYI)

Purpose: To verify that an NVMe ZNS capable controller can properly process a Zoned Management Send command if an invalid SLBA is provided.

References:

[1] NVMe ZNS Specification 4.3

Resource Requirements: Tools capable of monitoring and decoding traffic on the chosen NVMe transport.

Last Modification: June 10, 2021

Discussion: If the command SLBA field does not specify the starting logical block for a zone in the specified zoned namespace and the Select All bit is cleared to '0', then the command shall be aborted with a status code of Invalid Field in Command.

Test Setup: See Appendix A.

Case 1: ZSIO Reset Zone Select All=0 Invalid SLBA (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSIO: Implicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Reset Zone and Select All=0, and an SLBA value which does not specify a starting logical block for a zone in the specified zoned namespace.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSIO.
2. Verify that each Zone Management Send command completes with Status 'Invalid Field in Command' 02h.

Case 2: ZSEO Reset Zone Select All=0 Invalid SLBA (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSEO: Explicitly Opened state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Reset Zone and Select All=0, and an SLBA value which does not specify a starting logical block for a zone in the specified zoned namespace.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:

1. Verify that the current Zone State at the end of the test is ZSEO.
2. Verify that each Zone Management Send command completes with Status 'Invalid Field in Command' 02h.

Case 3: ZSC Reset Zone Select All=0 Invalid SLBA (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSC: Closed state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Reset Zone and Select All=0, and an SLBA value which does not specify a starting logical block for a zone in the specified zoned namespace.

©2021 UNH–IOL

4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:
1. Verify that the current Zone State at the end of the test is ZSC.
2. Verify that each Zone Management Send command completes with Status 'Invalid Field in Command' 02h.

Case 4: ZSF Reset Zone Select All=0 Invalid SLBA (FYI)

Test Procedure:
1. Configure the Testing Station to put the DUT into the ZSF: Full state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Reset Zone and Select All=0, and an SLBA value which does not specify a starting logical block for a zone in the specified zoned namespace.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:
1. Verify that the current Zone State at the end of the test is ZSF.
2. Verify that each Zone Management Send command completes with Status 'Invalid Field in Command' 02h.

Case 5: ZSE Reset Zone Select All=0 Invalid SLBA (FYI)

Test Procedure:
1. Configure the Testing Station to put the DUT into the ZSE: Empty state.
2. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
3. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Reset Zone and Select All=0, and an SLBA value which does not specify a starting logical block for a zone in the specified zoned namespace.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.

Observable Results:
1. Verify that the current Zone State at the end of the test is ZSE.
2. Verify that each Zone Management Send command completes with Status 'Invalid Field in Command' 02h.

Possible Problems: None.

©2021 UNH–IOL

Group 3: NVM Command Set I/O Commands

Overview:

This section describes a method for performing conformance verification for NVMe ZNS products implementing NVM Command Set I/O Commands defined in Chapter 4 of the NVMe ZNS Specification.

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).

©2021 UNH–IOL

Test 3.1 – Compare Command (FYI)

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

References:

[1] NVMe Specification 6.6, 8.3.1.4, 8.3.1.4, NVMe v1.3 ECN 001
[2] NVMe ZNS Specification 3.3.1

Resource Requirements:

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

Last Modification: June 10, 2021

Discussion: The Compare command reads the logical blocks specified by the command from the medium and compares the data read to a comparison data buffer transferred as part of the command. If the data read from the controller and the comparison data buffer are equivalent with no miscompares, then the command completes successfully. If there is any miscompare, the command completes with an error of Compare Failure. If metadata is provided, then a comparison is also performed for the metadata.

The command uses Command Dword 10, Command Dword 11, Command Dword 12, Command Dword 14, and Command Dword 15. If the command uses PRPs for the data transfer, then the Metadata Pointer, PRP Entry 1, and PRP Entry 2 fields are used. If the command uses SGLs for the data transfer, then the Metadata SGL Segment Pointer and SGL Entry 1 fields are used. All other command specific fields are reserved.

Test Setup: See Appendix A.

Case 1: Valid SLBA (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports the Compare command. If the command is not supported this test is not applicable.
2. Configure the NVMe Host to issue a Write command to the controller in order to write a known sequence to the LBA 0000h.
3. Configure the NVMe Host to issue a Compare command comparing the known sequence to the LBA 0000h written to in Step 1.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status for the command.
2. Verify that the Compare command completes successfully.

Case 2: SLBA Out of Range (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports the Compare command. If the command is not supported this test is not applicable.
2. Configure the NVMe Host to issue a Write command to the controller in order to write a known sequence to the LBA 0000h.
4. Configure the NVMe Host to issue a Compare command to an SLBA which is out of range of the DUT, comparing the known sequence to the LBA 0000h written to in Step 1.

Observable Results:

1. Verify that the Compare command completes with status code LBA Out of Range (80h).

Case 3: SLBA In Range, NLB Goes out of range (FYI)

Test Procedure:

©2021 UNH–IOL

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports the Compare command. If the command is not supported this test is not applicable.
2. Configure the NVMe Host to issue a Write command to the controller in order to write a known sequence to the LBA 0000h.
3. Configure the NVMe Host to issue a Compare command to the controller to an SLBA which is in range of the DUT, but an NLB value that will push the Compare Command past the size of the namespace, comparing the known sequence to the LBA 0000h written to in Step 1.

Observable Results:

1. Verify that the Compare command completes with status code LBA Out of Range (80h).

Case 4: SLBA Out of Range, NLB > MDTS (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports the Compare command. If the command is not supported this test is not applicable.
2. Configure the NVMe Host to issue a Write command to the controller in order to write a known sequence to the LBA 0000h.
3. Configure the NVMe Host to issue a Compare command to the controller comparing the known sequence to the LBA 0000h written to in Step 1, to an SLBA which out of range of the DUT, and an NLB which is greater than MDTS. If MDTS is set to zero (unlimited), then this test case is not applicable.

Observable Results:

1. Verify that the Compare command completes with an error status code. The DUT can report any error status code.

Case 5: SLBA Out of Range, but Lower Dword = 00000000 (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports the Compare command. If the command is not supported this test is not applicable.
2. Configure the NVMe Host to issue a Write command to the controller in order to write a known sequence to the LBA 0000h.
3. Configure the NVMe Host to issue a Compare command to the controller comparing the known sequence to the LBA 0000h written to in Step 1, with an SLBA of FFFFFFFF00000000h ,which, when read as a 64 bit value is out of range of the DUT, and an NLB which is less than MDTS.

Observable Results:

1. Verify that the Compare command completes with status code LBA Out of Range (80h).

Case 6: Invalid Namespace ID (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports the Compare command. If the command is not supported this test is not applicable.
2. Configure the NVMe Host to issue a Write command to the controller in order to write a known sequence to the LBA 0000h.
3. Configure the NVMe Host to issue a Compare command to the controller comparing the known sequence to the LBA 0000h written to in Step 1, specifying an Invalid NSID outside of the controller specified range defined by NN (Number of Namespaces). If NN is the maximum possible value (0xFFFFFFFF) this sub test cannot be performed.

Observable Results:

1. Verify that the Compare command completes with status code Invalid Namespace or Format (0Bh).

Case 7: PRCHK Non-zero (FYI)

Test Procedure:

©2021 UNH–IOL

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports the Compare command. If the command is not supported this test is not applicable.
2. Ensure that the namespace under test is formatted with end to end data protection. If End to End Data Protection is not supported then this test is not applicable.
3. Configure the NVMe Host to issue a Write command with PRACT set to 1, and PRCHK Bit 00 set to 1, to the controller in order to write a known sequence to the LBA 0000h.
4. Configure the NVMe Host to issue a Compare command to the controller with PRACT set to 0, and PRCHK Bit 00 set to 1, comparing the known sequence to the LBA 0000h written to in Step 3.
5. Repeat steps 3 and 4 with PRCHK Bit 01 set to 1, and again with PRCHK Bit 02 set to 1.

Observable Results:
1. Verify that in each case the Compare command completes successfully.

Case 8: NSID=FFFFFFFFh (FYI)

Test Procedure:
1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports the Compare command. If the command is not supported this test is not applicable.
2. Check the NVMe specification version supported by the DUT. If the specification is not v1.4 or higher, this test is not applicable.
3. Configure the NVMe Host to issue a Write command to the controller in order to write a known sequence to the LBA 0000h.
4. Configure the NVMe Host to issue a Compare command with NSID=FFFFFFFFh comparing the known sequence to the LBA 0000h written to in Step 1.

Observable Results:
1. Verify that the Compare command did not complete successfully.

Case 9: Zone Boundary Error (FYI)

Test Procedure:
1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports the Compare command. If the command is not supported this test is not applicable.
2. Configure the Testing Station to put the DUT into the ZSIO state.
3. Configure the Testing Station acting as a Host to issue a Compare command to logical blocks in more than one zone.

Observable Results:
1. Verify that the Compare command is aborted with a status code of “Zone Boundary Error” B8h.

Case 10: Zone is Full (FYI)

Test Procedure:
1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports the Compare command. If the command is not supported this test is not applicable.
2. Configure the Testing Station to put the DUT into the ZSF state.
3. Configure the Testing Station acting as a Host to issue a Compare command to an LBA in the zone which is in ZSF state.

Observable Results:
1. Verify that the Compare command is aborted with a status code of “Zone is Full” B9h.

Case 11: Zone Invalid Write (FYI)

Test Procedure:

©2021 UNH–IOL

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports the Compare command. If the command is not supported this test is not applicable.
2. Configure the Testing Station to put the DUT into the ZSIO state.
3. Configure the Testing Station acting as a Host to issue a Compare command to an LBA different than the write pointer.

Observable Results:

1. Verify that the Compare command is aborted with a status code of "Zone Invalid Write" BCh.

Case 12: Compare Exceeds Maximum Active Resources (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports the Compare command. If the command is not supported this test is not applicable.
2. Configure the Testing Station to issue an Identify Command for CNS 05h, I/O Command Set Specific Identify Namespace Data Structures and record the Maximum Active Resources value (MAR) and the Maximum Open Resources value (MOR). If the MAR field is set to FFFFFFFFh this test is not applicable.
3. Configure the Testing Station to put the DUT into the ZSE state using a Zone Management Send command with Action of Reset.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
5. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Open Zone and Select All=1, and a valid SLBA. Repeat this step for different zones until the number of active zones is 1 less than the MAR value. If opening a zone will exceed the MOR value, some zones may be closed with a Zone Management Send command with action Close Zone, which will decrease the Open Resources count but maintain the Active Resources count.
6. Configure the Testing Station acting as a Host to send a Compare command for a new zone.

Observable Results:

1. Verify that the Compare command is aborted with a status code of "Too Many Active Zones" BDh.

Case 13: Compare Exceeds Maximum Open Resources (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports the Compare command. If the command is not supported this test is not applicable.
2. Configure the Testing Station to issue an Identify Command for CNS 05h, I/O Command Set Specific Identify Namespace Data Structures and record the Maximum Open Resources value (MOR). If the MOR field is set to FFFFFFFFh this test is not applicable.
3. Configure the Testing Station to put the DUT into the ZSE state using a Zone Management Send command with Action of Reset.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
5. Configure the Testing Station acting as a Host to issue a Compare command with a valid SLBA. Repeat this step to different zones until the number of implicitly opened zones exceeds the MOR value.

Observable Results:

1. If MAR is greater than MOR, verify that the Compare command which causes the number of Open Resources to exceed the value specified by the Maximum Open Resources field is aborted with a status code of Too Many Open Zones BEh.
2. If MAR is equal to MOR, verify that the Compare command which causes the number of Open Resources to exceed the value specified by the Maximum Open Resources field is aborted with a status code of Too Many Active Zones BDh.

Possible Problems: None.

©2021 UNH–IOL

Test 3.2 – Copy (FYI)

Purpose: To verify that an NVMe Controller proper implements the Copy Command if supported.

References:

[1] NVMe Specification Chapter 6
[2] NVMe ZNS Specification 3.3.2

Resource Requirements:

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

Last Modification: June 3, 2021

Discussion:. The Copy command is used by the host to copy data from one or more source logical block ranges to a single consecutive destination logical block range.

Test Setup: See Appendix A.

Case 1: Copy Command Supported (FYI)

Test Procedure:

1. Perform an Identify Controller Data Structure (CNS=01h) to the DUT. Check the ONCS field Bit 8 to determine if the DUT supports the Copy Command. If the command is not supported this test is not applicable.
2. Perform an Identify Namespace Data Structure (CNS=00h) to the DUT.

Observable Results:

1. Verify that the MSSRL and MCL fields in the Identify Namespace Data Structure are set to non-zero values.

Case 2: MSSRL Exceeded (FYI)

Test Procedure:

1. Perform an Identify Controller Data Structure (CNS=01h) to the DUT. Check the ONCS field Bit 8 to determine if the DUT supports the Copy Command. If the command is not supported this test is not applicable.
2. Perform an Identify Namespace Data Structure (CNS=00h) to the DUT, record the MSSRL value.
3. Perform a Copy Command with an NLB field greater than the MSSRL value.

Observable Results:

1. Verify that the Copy Command is aborted with a status of ‘Command Size Limit Exceeded’.

Case 3: MCL Exceeded (FYI)

Test Procedure:

1. Perform an Identify Controller Data Structure (CNS=01h) to the DUT. Check the ONCS field Bit 8 to determine if the DUT supports the Copy Command. If the command is not supported this test is not applicable.
2. Perform an Identify Namespace Data Structure (CNS=00h) to the DUT, record the MCL value.
3. Perform a Copy Command with the sum of all NLB values in all Source Range Entries exceeding the MCL value.

Observable Results:

1. Verify that the Copy Command is aborted with a status of ‘Command Size Limit Exceeded’.

©2021 UNH–IOL

Case 4: MSRC Exceeded (FYI)

Test Procedure:

1. Perform an Identify Controller Data Structure (CNS=01h) to the DUT. Check the ONCS field Bit 8 to determine if the DUT supports the Copy Command. If the command is not supported this test is not applicable.
2. Perform an Identify Namespace Data Structure (CNS=00h) to the DUT, record the MSRC value.
3. Perform a Copy Command with an NR field greater than the MSRC value.

Observable Results:

1. Verify that the Copy Command is aborted with a status of 'Command Size Limit Exceeded'.

Case 5: Zone Boundary Error (FYI)

Test Procedure:

1. Perform an Identify Controller Data Structure (CNS=01h) to the DUT. Check the ONCS field Bit 8 to determine if the DUT supports the Copy Command. If the command is not supported this test is not applicable.
2. Configure the Testing Station to put the DUT into the ZSIO state.
3. Configure the Testing Station acting as a Host to issue a Copy command to logical blocks in more than one zone.

Observable Results:

1. Verify that the Copy command is aborted with a status code of "Zone Boundary Error" B8h.

Case 6: Zone is Full (FYI)

Test Procedure:

1. Perform an Identify Controller Data Structure (CNS=01h) to the DUT. Check the ONCS field Bit 8 to determine if the DUT supports the Copy Command. If the command is not supported this test is not applicable.
2. Configure the Testing Station to put the DUT into the ZSF state.
3. Configure the Testing Station acting as a Host to issue a Copy command to an LBA in the zone which is in ZSF state.

Observable Results:

1. Verify that the Copy command is aborted with a status code of "Zone is Full" B9h.

Case 7: Zone Invalid Write (FYI)

Test Procedure:

1. Perform an Identify Controller Data Structure (CNS=01h) to the DUT. Check the ONCS field Bit 8 to determine if the DUT supports the Copy Command. If the command is not supported this test is not applicable.
2. Configure the Testing Station to put the DUT into the ZSIO state.
3. Configure the Testing Station acting as a Host to issue a Copy command to an LBA different than the write pointer.

Observable Results:

1. Verify that the Copy command is aborted with a status code of "Zone Invalid Write" BCh.

Case 8: Copy Exceeds Maximum Active Resources (FYI)

Test Procedure:

©2021 UNH–IOL

1. Perform an Identify Controller Data Structure (CNS=01h) to the DUT. Check the ONCS field Bit 8 to determine if the DUT supports the Copy Command. If the command is not supported this test is not applicable.
2. Configure the Testing Station to issue an Identify Command for CNS 05h, I/O Command Set Specific Identify Namespace Data Structures and record the Maximum Active Resources value (MAR) and the Maximum Open Resources value (MOR). If the MAR field is set to FFFFFFFFh this test is not applicable.
3. Configure the Testing Station to put the DUT into the ZSE state using a Zone Management Send command with Action of Reset.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
5. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Open Zone and Select All=1, and a valid SLBA. Repeat this step for different zones until the number of active zones is 1 less than the MAR value. If opening a zone will exceed the MOR value, some zones may be closed with a Zone Management Send command with action Close Zone, which will decrease the Open Resources count but maintain the Active Resources count.
6. Configure the Testing Station acting as a Host to send a Copy command for a new zone.

Observable Results:

1. Verify that the Copy command is aborted with a status code of "Too Many Active Zones" BDh.

Case 9: Copy Exceeds Maximum Open Resources (FYI)

Test Procedure:

1. Perform an Identify Controller Data Structure (CNS=01h) to the DUT. Check the ONCS field Bit 8 to determine if the DUT supports the Copy Command. If the command is not supported this test is not applicable.
2. Configure the Testing Station to issue an Identify Command for CNS 05h, I/O Command Set Specific Identify Namespace Data Structures and record the Maximum Open Resources value (MOR). If the MOR field is set to FFFFFFFFh this test is not applicable.
3. Configure the Testing Station to put the DUT into the ZSE state using a Zone Management Send command with Action of Reset.
4. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
5. Configure the Testing Station acting as a Host to issue a Copy command with a valid SLBA. Repeat this step to different zones until the number of implicitly opened zones exceeds the MOR value.

Observable Results:

1. If MAR is greater than MOR, verify that the Copy command which causes the number of Open Resources to exceed the value specified by the Maximum Open Resources field is aborted with a status code of Too Many Open Zones BEh.
2. If MAR is equal to MOR, verify that the Copy command which causes the number of Open Resources to exceed the value specified by the Maximum Open Resources field is aborted with a status code of Too Many Active Zones BDh.

Possible Problems: None.

©2021 UNH–IOL

Test 3.3 – Dataset Management Command (FYI)

Purpose: To verify that an NVMe Controller can properly execute the Dataset Management command.

References:

[1] NVMe Specification 6.7
[2] NVMe ZNS Specification 3.3.3

Resource Requirements:

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

Last Modification: November 26, 2018

Discussion: The Dataset Management command is used by the host to indicate attributes for ranges of logical blocks. This includes attributes like frequency that data is read or written, access size, and other information that may be used to optimize performance and reliability. This command is advisory; a compliant controller may choose to take no action based on information provided.

Test Setup: See Appendix A.

Case 1: Basic Operation (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure or check for non-zero values in the DMRL, DMRSL, and DMSL fields to determine if the DUT supports Dataset Management. If the command is not supported this test is not applicable.
2. Configure the NVMe Host to issue the Dataset Management command that indicates attributes for ranges of logical blocks.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.

Case 2: Deallocate (FYI)

The Dataset Management command may also be used to deallocate ranges of logical blocks by setting the Attribute – Deallocate (AD) field of Command Dword 11 to '1'. When the controller receives a Dataset Management command with the AD field set to '1', it may deallocate all provided ranges. If a read occurs to a deallocated range, the controller shall return all zeros, all ones, or the last data written to the associated LBA. If the deallocated or unwritten logical block error is enabled and a read occurs to a deallocated range, then the read shall fail with the Unwritten or Deallocated Logical Block status code.

An LBA that has been deallocated using the Dataset Management command is no longer deallocated when the LBA is written. Read operations do not affect the deallocation status of an LBA. The value read from a deallocated LBA shall be deterministic; specifically, the value returned by subsequent reads of that LBA shall be the same until a write occurs to that LBA.

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure or check for non-zero values in the DMRL, DMRSL, and DMSL fields to determine if the DUT supports Dataset Management. If the command is not supported this test is not applicable.
2. Configure the NVMe Host to issue a Write command to write a known data pattern to the controller.
3. Configure the NVMe Host to issue a Dataset Management command with the Attribute – Deallocate (AD) field set to '1' and specifying the same LBA range written to in step 2 to the controller.
4. Configure the NVMe Host to issue a Read command to read the same LBA range written to in step 2.

©2021 UNH–IOL

5. Configure the NVMe Host to issue another Read command to the controller in order to read the same LBA range written to in step 2.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.
2. Verify that the data returned by the controller from the Read command in step 4 is either all zeros, all ones, or the data written to the associated LBA in step 2.
3. Verify that the data returned by the controller from the Read command in step 5 is identical to the data returned by the controller from the Read command in step 4.

Case 3: NR Value is Maximum (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure or check for non-zero values in the DMRL, DMRSL, and DMSL fields to determine if the DUT supports Dataset Management. If the command is not supported this test is not applicable.
2. Configure the NVMe Host to issue a Dataset Management command with the Number of Ranges (NR) field set to all '1's'.

Observable Results:

1. Verify that the command completes successfully.

Case 4: Correct Range Deallocated (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure or check for non-zero values in the DMRL, DMRSL, and DMSL fields to determine if the DUT supports Dataset Management. If the command is not supported this test is not applicable.
2. Write a known data pattern (e.g. 'AAAA') to three consecutive LBAs.
3. Configure the NVMe Host to issue a Dataset Management command with the Attribute – Deallocate (AD) field set to '1' and specifying the middle LBA of the LBA written in step 1.
4. Perform a READ operation to each of the three LBAs.

Observable Results:

1. Verify that the Dataset Management command completes successfully.
2. Verify that the data returned for the READ command to the first and third LBAs is the known data pattern in step 1
3. Verify that the data returned from the READ operation to the second LBA is either all zeros, all ones, or the known data pattern from step 1.

Case 5: Deallocate Multiple Ranges (FYI)

The Dataset Management command may also be used to deallocate ranges of logical blocks by setting the Attribute – Deallocate (AD) field of Command Dword 11 to '1'. When the controller receives a Dataset Management command with the AD field set to '1', it may deallocate all provided ranges. If a read occurs to a deallocated range, the controller shall return all zeros, all ones, or the last data written to the associated LBA. If the deallocated or unwritten logical block error is enabled and a read occurs to a deallocated range, then the read shall fail with the Unwritten or Deallocated Logical Block status code.

An LBA that has been deallocated using the Dataset Management command is no longer deallocated when the LBA is written. Read operations do not affect the deallocation status of an LBA. The value read from a deallocated LBA shall be deterministic; specifically, the value returned by subsequent reads of that LBA shall be the same until a write occurs to that LBA.

Test Procedure:

©2021 UNH–IOL

1. Check the ONCS field of the Identify Controller Data structure or check for non-zero values in the DMRL, DMRSL, and DMSL fields to determine if the DUT supports Dataset Management. If the command is not supported this test is not applicable.
2. Configure the NVMe Host to issue a Write command to write a known data pattern to a range of LBAs in the controller.
3. Configure the NVMe Host to issue a Dataset Management command with the Attribute – Deallocate (AD) field set to '1' and specifying 3 separate smaller LBA ranges within the single larger LBA range written to in step 2 to the controller. The total capacity of the 3 smaller LBA ranges should be less than the capacity of the larger LBA range written in step 2, such that part of the larger range is not affected by the Dataset Management command.
4. Configure the NVMe Host to issue a Read command to read the same LBA range written to in step 2.
5. Configure the NVMe Host to issue another Read command to the controller in order to read the same LBA range written to in step 2.
6. Configure the NVMe Host to issue a Write command to write a known data pattern (but different than the data pattern used in step 2) to the controller specifying the same LBA range previously deallocated.
7. Configure the NVMe Host to issue a Read command to the controller in order to read the same LBA range written to in step 6.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.
2. Verify that the data returned by the controller from the Read command in step 4 is either all zeros, all ones, or the data written to the associated LBA in step 2.
3. Verify that the data returned by the controller from the Read command in step 5 is identical to the data returned by the controller from the Read command in step 4.
4. Verify that the data returned by the controller from the Read command in step 7 matches the data written in step 6.

Case 6: NSID=FFFFFFFFh (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure or check for non-zero values in the DMRL, DMRSL, and DMSL fields to determine if the DUT supports Dataset Management. If the command is not supported this test is not applicable.
2. Check the NVMe specification version supported by the DUT. If the specification is not v1.4 or higher, this test is not applicable.
3. Configure the NVMe Host to issue the Dataset Management command with NSID=FFFFFFFFh that indicates attributes for ranges of logical blocks.

Observable Results:

1. Verify that the Dataset Management Command does not complete successfully.

Case 7: ONCS Bit 2 =1 Non-MDTS Command Size Limits DMRL (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports Dataset Management. If the DSM command is not supported this test is not applicable.
2. Check the DMRL values of the DUT.
3. Transmit a Data Set Management command in which NR (Number of Ranges) exceeds the DMRL value.

Observable Results:

1. Verify that the Dataset Management Command does not return status 'Command Limit Exceeded'.

©2021 UNH–IOL

Case 8: ONCS Bit 2 =1 DSM Supported Non-MDTS Command Size Limits DMRSL (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports Dataset Management. If the DSM command is not supported this test is not applicable.
2. Check the DMRL values of the DUT.
3. Transmit a Data Set Management command in which the number of logical blocks in a single range exceeds the DMRSL value.

Observable Results:

1. Verify that the Dataset Management Command does not return status 'Command Limit Exceeded'.

Case 9: ONCS Bit 2 =1 DSM Supported Non-MDTS Command Size Limits DMSL (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports Dataset Management. If the DSM command is not supported this test is not applicable.
2. Check the DMRL values of the DUT.
3. Transmit a Data Set Management command in which the total number of logical blocks in the command exceeds the DMSL value.

Observable Results:

1. Verify that the Dataset Management Command does not return status 'Command Limit Exceeded'.

Case 10: ONCS Bit 2 =0 Non-MDTS Command Size Limits DMRL (FYI)

Test Procedure:

1. Check the ONCS Bit 2 of the Identify Controller Data structure. If ONCS Bit 2 is set to 1, this test is not applicable.
2. Check the DMRL values of the DUT.
3. Transmit a Data Set Management command in which NR (Number of Ranges) exceeds the DMRL value.

Observable Results:

1. Verify that the Dataset Management Command returns status 'Command Limit Exceeded'.

Case 11: ONCS Bit 2 =0 DSM Supported Non-MDTS Command Size Limits DMRSL (FYI)

Test Procedure:

1. Check the ONCS Bit 2 of the Identify Controller Data structure. If ONCS Bit 2 is set to 1, this test is not applicable.
2. Check the DMRL values of the DUT.
3. Transmit a Data Set Management command in which the number of logical blocks in a single range exceeds the DMRSL value.

Observable Results:

1. Verify that the Dataset Management Command returns status 'Command Limit Exceeded'.

Case 12: ONCS Bit 2 =0 DSM Supported Non-MDTS Command Size Limits DMSL (FYI)

Test Procedure:

1. Check the ONCS Bit 2 of the Identify Controller Data structure. If ONCS Bit 2 is set to 1, this test is not applicable.
2. Check the DMRL values of the DUT.

©2021 UNH–IOL

3. Transmit a Data Set Management command in which the total number of logical blocks in the command exceeds the DMSL value.

Observable Results:

1. Verify that the Dataset Management Command returns status ‘Command Limit Exceeded’.

Possible Problems: None.

©2021 UNH–IOL

Test 3.4 – Flush Command (FYI)

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

References:

[1] NVMe Base Specification 6.8, TP 4035
[1] NVMe ZNS Specification 3.3.4

Resource Requirements:

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

Last Modification: June 10, 2021

Discussion: The Flush command shall commit data and metadata associated with the specified namespace(s) to non–volatile media. The flush applies to all commands completed prior to the submission of the Flush command. The controller may also flush additional data and/or metadata from any namespace.

All command specific fields are reserved.

Test Setup: See Appendix A.

Case 1: Valid Namespace ID (FYI)

Test Procedure:

1. For each active namespace, configure the NVMe Host to issue a Flush command to the NVMe Controller specifying the NSID of the specified namespace.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.
2. If the VWC field Bit 0 of the Identify Controller data structure is cleared to 0, verify that the completion queue entry for each Flush command indicate success.
3. Verify that all received responses have all Reserved fields set to 0.

Case 2: Invalid Namespace ID (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Flush command to the NVMe Controller specifying an Invalid NSID outside of the controller specified range defined by NN (Number of Namespaces). If NN is the maximum possible value (0xFFFFFFFF) this sub test cannot be performed.

Observable Results:

1. Verify that the Flush command completes with status code Invalid Namespace or Format (0Bh).

Case 3: NSID=0xFFFFFFFF (FYI)

Test Procedure:

1. Check the VWC field of the Identify Controller Data Structure.
2. Configure the NVMe Host to issue a Flush command to the NVMe Controller specifying an NSID of 0xFFFFFFFF.

Observable Results:

1. If VWC bits 2:1 is 00b, verify that the DUT indicates support for version 1.3 or earlier of the NVMe specification in the VER field.
2. If VWC bits 2:1 is 10b, verify that controller fails the command with status code Invalid Namespace or Format.

©2021 UNH–IOL

3. If VWC bits 2:1 is 11b, verify that the Flush command completes with status success, and is applied to all namespaces attached to the controller.
4. Verify that VWC bits 2:1 is not set to 01b.

Possible Problems: None.

©2021 UNH–IOL

Test 3.5 – Read Command (FYI)

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

References:

[1] NVMe Specification 6.9, 9.4, NVMe v1.3 ECN 001
[2] NVMe ZNS Specificataion 3.3.5

Resource Requirements:

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

Last Modification: June 10, 2021

Discussion: The Read command reads data and metadata, if applicable, from the NVM controller for the LBAs indicated. The command may specify protection information to be checked as part of the read operation.

The command uses Command Dword 10, Command Dword 11, Command Dword 12, Command Dword 13, Command Dword 14, and Command Dword 15 fields. If the command uses PRPs for the data transfer, then the Metadata Pointer, PRP Entry 1, and PRP Entry 2 fields are used. If the command uses SGLs for the data transfer, then the Metadata SGL Segment Pointer and SGL Entry 1 fields are used. If supported, this test may be performed using 4k sector sizes.

Regarding the reporting of error status codes, the NVMe specification states: "The status code of the completion queue entry should indicate an Internal Error status code (if multiple error conditions exist, the lowest numerical value is returned)."

Test Setup: See Appendix A.

Case 1: Valid Read, LR=0, FUA=0 (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Write command to the controller in order to write a known data pattern to one LBA on the NVMe Device.
2. Configure the NVMe Host to issue a Read command to the controller in order to read from the same LBA which was written to in step 1, with LR=0, FUA=0.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.
2. Verify that the known data pattern was read correctly from the NVMe Device exactly as it was written.

Case 2: SLBA Out of Range (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to the NVMe Controller specifying CNS value 01h in order to retrieve the Identify Controller data structure.
2. Configure the NVMe Host to issue a Read command to the controller to an SLBA which is out of range of the DUT.

Observable Results:

1. Verify that the READ command completes with status code LBA Out of Range (80h).

Case 3: SLBA In Range, NLB Goes out of range (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to the NVMe Controller specifying CNS value 01h in order to retrieve the Identify Controller data structure.

©2021 UNH–IOL

2. Configure the NVMe Host to issue a Read command to the controller to an SLBA which in range of the DUT, but an NLB value that will push the Read Command past the capacity of the DUT.

Observable Results:
1. Verify that the READ command completes with status code Invalid Field (02h) when the NLB value specifies a value that exceeds MDTS.
2. If NLB is out of range and does not exceed MDTS, verify that the READ command completes with status code LBA Out of Range (0x80).

Case 4: SLBA Out of Range, NLB > MDTS (FYI)

Test Procedure:
1. Check the MDTS value reported by the DUT in the Identify Controller Data Structure. If MDTS is set to 0 (i.e. unlimited) then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to the NVMe Controller specifying CNS value 01h in order to retrieve the Identify Controller data structure.
3. Configure the NVMe Host to issue a Read command to the controller to an SLBA which is out of range of the DUT, and an NLB which is greater than MDTS.

Observable Results:
1. Verify that the READ command completes with status code of either Invalid Field (02h) or LBA out of Range (80h).

Case 5: SLBA Out of Range, but Lower Dword = 00000000 (FYI)

Test Procedure:
1. If an SLBA of FFFFFFFF00000000h is within range of the DUT, then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to the NVMe Controller specifying CNS value 01h in order to retrieve the Identify Controller data structure.
3. Configure the NVMe Host to issue a Read command to the controller with an SLBA of FFFFFFFF00000000h ,which, when read as a 64 bit value is out of range of the DUT, and an NLB which is less than MDTS.

Observable Results:
1. Verify that the READ command completes with status code LBA Out of Range (80h).

Case 6: Invalid Namespace ID (FYI)

Test Procedure:
1. Check the NN Value in the Identify Controller Data Structure. If NN is set to 0xFFFFFFFF, then this test Is not applicable.
2. Configure the NVMe Host to issue an Identify command to the NVMe Controller specifying CNS value 01h in order to retrieve the Identify Controller data structure.
3. Configure the NVMe Host to issue a Read command to the controller to the LBA 0000h.
4. Configure the NVMe Host to issue a READ command to the NVMe Controller specifying an Invalid NSID outside of the controller specified range defined by NN (Number of Namespaces).

Observable Results:
1. Verify that the READ command completes with status code Invalid Namespace or Format (0Bh).

Case 7: Invalid Namespace ID and SLBA Out of Range (FYI)

Test Procedure:
1. Check the NN field in the Identify Controller Data Structure. If NN is the maximum possible value (0xFFFFFFFF) this sub test cannot be performed.
2. Configure the NVMe Host to issue an Identify command to the NVMe Controller specifying CNS value 01h in order to retrieve the Identify Controller data structure.
3. Configure the NVMe Host to issue a Read command to the controller to the LBA 0000h.

©2021 UNH–IOL

4. Configure the NVMe Host to issue a Read command to an invalid Namespace ID with an SLBA value that is beyond the capacity of the DUT. The READ command to the NVMe Controller specifying an Invalid NSID outside of the controller specified range defined by NN (Number of Namespaces).

Observable Results:

1. Verify that the READ command completes with status code of either Invalid Namespace or Format (0Bh), or LBA Out of Range (80h).

Case 8: Valid Read, LR=0, FUA=1 (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Write command to the controller in order to write a known data pattern to one LBA on the NVMe Device.
2. Configure the NVMe Host to issue a Read command to the controller in order to read from the same LBA which was written to in step 1, with LR=0, FUA=1.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.
2. Verify that the known data pattern was read correctly from the NVMe Device exactly as it was written.

Case 9: Valid Read, LR=1, FUA=0 (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Write command to the controller in order to write a known data pattern to one LBA on the NVMe Device.
2. Configure the NVMe Host to issue a Read command to the controller in order to read from the same LBA which was written to in step 1, with LR=1, FUA=0.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.
2. Verify that the known data pattern was read correctly from the NVMe Device exactly as it was written.

Case 10: Valid Read, LR=1, FUA=1 (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Write command to the controller in order to write a known data pattern to one LBA on the NVMe Device.
2. Configure the NVMe Host to issue a Read command to the controller in order to read from the same LBA which was written to in step 1, with LR=1, FUA=1.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.
2. Verify that the known data pattern was read correctly from the NVMe Device exactly as it was written.

Case 11: NSID=FFFFFFFFh, LR=0, FUA=0 (FYI)

Test Procedure:

1. Check the NVMe specification version supported by the DUT. If the specification is not v1.4 or higher, this test is not applicable.
2. Configure the NVMe Host to issue a Read command to the controller with NSID=FFFFFFFFh in order to read from the same LBA which was written to in step 1, with LR=0, FUA=0.

Observable Results:

1. Verify that the READ command does not complete successfully.

©2021 UNH–IOL

Case 12: Read Unwritten Block DULBE=0 (FYI)

Test Procedure:

1. Configure the Testing Station to issue an Identify Command for CNS = 00h. Record the DLFEAT value.
2. Configure the Testing Station to issue a Set Feature command for the Error Recovery Feature (FID=05h) to set DULBE=0.
3. Configure the Testing Station to issue a Get Feature command for the Error Recovery Feature (FID=05h) to check that DULBE=0.
4. Configure the NVMe Host to issue a Write command to the controller in order to write a known data pattern to one LBA on the NVMe Device.
5. Configure the NVMe Host to issue a Read command for a Logical Block that is between the last written LBA ina zone and the last LBA in a zone.

Observable Results:

1. Verify that the the Read command completes with status success.
2. Verify that and the values of the data read are:
    a. all bytes cleared to 0h if bits 2:0 in the DLFEAT field are set to 001b;
    b. all bytes set to FFh if bits 2:0 in the DLFEAT field are set to 010b; or
    c. either all bytes cleared to 0h or all bytes set to FFh if bits 2:0 in the DLFEAT field are set to 000b.

Case 13: Read Unwritten Block DULBE=1 (FYI)

Test Procedure:

1. Configure the Testing Station to issue a Set Feature command for the Error Recovery Feature (FID=05h) to set DULBE=1.
2. Configure the Testing Station to issue a Get Feature command for the Error Recovery Feature (FID=05h) to check that DULBE=1.
3. Configure the NVMe Host to issue a Write command to the controller in order to write a known data pattern to one LBA on the NVMe Device.
4. Configure the NVMe Host to issue a Read command for a Logical Block that is between the last written LBA ina zone and the last LBA in a zone.

Observable Results:

1. Verify that the the Read command completes with status “Deallocated or Unwritten Logical Block” 87h.

Case 14: Zone Boundary Error (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSIO state.
2. Configure the Testing Station acting as a Host to issue a Read command to logical blocks in more than one zone.

Observable Results:

1. Verify that the Read command is aborted with a status code of “Zone Boundary Error” B8h.

Possible Problems: None.

©2021 UNH–IOL

Test 3.6 – Verify Command (FYI)

Purpose: To verify that an NVMe Controller can properly execute the Verify command, if supported.

References:

[1] NVMe Specification TP 4030
[2] NVMe ZNS Specification 3.3.6

Resource Requirements:

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

Last Modification: June 10, 2021

Discussion: The Verify command is roughly equivalent to a Read command that discards the data (and metadata) after reading. The important behavior of Verify is that errors are reported if the data (or metadata) cannot be read without expending the time and resources required to return the data (and metadata) to the host.

Test Setup: See Appendix A.

Case 1: Valid Command (FYI)

Test Procedure:

1. Check ONCS bit 7 or for a non-zero value in the VSL field to determine if the DUT supports the Verify Command. If the Verify Command is not supported, this test is not applicable.
2. Perform a Verify command of LBADS bytes. If the namespace is formatted with protection information, perform the Verify Command using the same protection information as the namespace was formatted with, and PRACT=0.
3. Repeat for all supported PRINFO values. It may be necessary to format the namespace appropriately to check all Protection Information types.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.
2. Verify that all received responses have all Reserved fields set to 0.

Case 2: PRACT = 1 (FYI)

Test Procedure:

1. Check ONCS bit 7 or for a non-zero value in the VSL field to determine if the DUT supports the Verify Command. If the Verify Command is not supported, this test is not applicable.
2. Perform a Verify command, using the same protection information as the namespace was formatted with, and with PRACT = 1.

Observable Results:

1. Verify that the Verify command completes with status Invalid Field in Command.

Case 3: NSID=FFFFFFFFh (FYI)

Test Procedure:

1. Check ONCS bit 7 or for a non-zero value in the VSL field to determine if the DUT supports the Verify Command. If the Verify Command is not supported, this test is not applicable.
2. Check the NVMe specification version supported by the DUT. If the specification is not v1.4 or higher, this test is not applicable.
3. Perform a Verify command of LBADS bytes and NSID=FFFFFFFFh. If the namespace is formatted with protection information, perform the Verify Command using the same protection information as the namespace was formatted with, and PRACT=0.

©2021 UNH–IOL

4. Repeat for all supported PRINFO values. It may be necessary to format the namespace appropriately to check all Protection Information types.

Observable Results:
1. Verify that none of the Verify commands complete successfully.

Case 4: Command Size Limits (FYI)

Test Procedure:
1. Check ONCS bit 7, if this bit is set to 1 this test is not applicable.
2. Check the VSL field in the Identify Controller Data structure. If VSL field is set to 0 this test is not applicable.
3. Perform a Verify command with an NLB value that exceeds VSL.

Observable Results:
1. Verify that the Verify command completes with status 'Invalid Field in Command'.

Case 5: Zone Boundary Error (FYI)

Test Procedure:
1. Configure the Testing Station to put the DUT into the ZSIO state.
2. Configure the Testing Station acting as a Host to issue a Verify command to logical blocks in more than one zone.

Observable Results:
1. Verify that the Verify command is aborted with a status code of "Zone Boundary Error" B8h.

Possible Problems: None.

©2021 UNH–IOL

Test 3.7 – Write (FYI)

Purpose: To verify that an NVMe ZNS capable controller can properly process a Write command.

References:

[1] NVMe Specification 6.14, 9.4

Resource Requirements:

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

Last Modification: June 10, 2021

Discussion: The Write command writes data and metadata, if applicable, to the NVM controller for the logical blocks indicated. The host may also specify protection information to include as part of the operation.

The command uses Command Dword 10, Command Dword 11, Command Dword 12, Command Dword 13, Command Dword 14, and Command Dword 15 fields. If the command uses PRPs for the data transfer, then the Metadata Pointer, PRP Entry 1, and PRP Entry 2 fields are used. If the command uses SGLs for the data transfer, then the Metadata SGL Segment Pointer and SGL Entry 1 fields are used. If supported, this test may be performed using 4k sector sizes.

Regarding the reporting of error status codes, the NVMe specification states: "The status code of the completion queue entry should indicate an Internal Error status code (if multiple error conditions exist, the lowest numerical value is returned)."

Test Setup: See Appendix A.

Case 1: Valid Write, LR=0, FUA=0 (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Write command to the controller in order to write a known data pattern to one LBA on the NVMe Device, with LR=0, FUA=0.
2. Configure the NVMe Host to issue a Read command to the controller in order to read from the same LBA which was written to in step 1.
3. Verify that all received responses have all Reserved fields set to 0.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.
2. Verify that the known data pattern was read correctly from the NVMe Device exactly as it was written.

Case 2: SLBA Out of Range (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to the NVMe Controller specifying CNS value 01h in order to retrieve the Identify Controller data structure.
2. Configure the NVMe Host to issue a Write command with a known data pattern to an SLBA which is out of range of the DUT.

Observable Results:

1. Verify that the Write command completes with status code LBA Out of Range (80h).

Case 3: SLBA In Range, NLB Goes out of range (FYI)

©2021 UNH–IOL

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to the NVMe Controller specifying CNS value 01h in order to retrieve the Identify Controller data structure.
2. Configure the NVMe Host to issue a Write command with a known data pattern to an SLBA which in range of the DUT, but an NLB value that will push the Write Command past the capacity of the DUT.

Observable Results:

1. Verify that the WRITE command completes with status code Invalid Field (02h) when the NLB value specifies a value that exceeds MDTS.
2. If NLB is out of range and does not exceed MDTS, verify that the WRITE command completes with status code LBA Out of Range (0x80) or Capacity Exceeded (0x81).

Case 4: SLBA Out of Range, NLB > MDTS (FYI)

Test Procedure:

1. Check the MDTS value reported by the DUT in the Identify Controller Data Structure. If MDTS is set to 0 (i.e. unlimited) then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to the NVMe Controller specifying CNS value 01h in order to retrieve the Identify Controller data structure.
3. Configure the NVMe Host to issue a Write command with a known data pattern to an SLBA which is out of range of the DUT, and an NLB which is greater than MDTS.

Observable Results:

1. Verify that the Write command completes with status code of either Invalid Field (02h) or LBA out of Range (80h).

Case 5: SLBA Out of Range, but Lower Dword = 00000000 (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to the NVMe Controller specifying CNS value 01h in order to retrieve the Identify Controller data structure.
2. Configure the NVMe Host to issue a Write command with a known data pattern with an SLBA of FFFFFFFF00000000h ,which, when read as a 64 bit value is out of range of the DUT, and an NLB which is less than MDTS.

Observable Results:

1. Verify that the Write command completes with status code LBA Out of Range (80h).

Case 6: Invalid Namespace ID (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to the NVMe Controller specifying CNS value 01h in order to retrieve the Identify Controller data structure.
2. Configure the NVMe Host to issue a Write command with a known data pattern, to an invalid Namespace ID. The Write command to the NVMe Controller should specify an Invalid NSID outside of the controller specified range defined by NN (Number of Namespaces). If NN is the maximum possible value (0xFFFFFFFF) this sub test cannot be performed.

Observable Results:

1. Verify that the Write command completes with status code Invalid Namespace or Format (0Bh).

Case 7: Invalid Namespace ID and SLBA Out of Range (FYI)

Test Procedure:

©2021 UNH–IOL

1. Check the NN field in the Identify Controller Data Structure. If NN is the maximum possible value (0xFFFFFFFF) this sub test cannot be performed.
2. Configure the NVMe Host to issue an Identify command to the NVMe Controller specifying CNS value 01h in order to retrieve the Identify Controller data structure.
3. Configure the NVMe Host to issue a Write command with a known data pattern, to an invalid Namespace ID with an SLBA value that is beyond the capacity of the DUT. The Write command to the NVMe Controller should specify an Invalid NSID outside of the controller specified range defined by NN (Number of Namespaces).

Observable Results:

1. Verify that the Write command completes with status code of either Invalid Namespace or Format (0Bh), or LBA Out of Range (80h).

Case 8: Valid Write, LR=0, FUA=1 (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Write command with a known data pattern, to the controller in order to write a known data pattern to one LBA on the NVMe Device, with LR=0, FUA=1.
2. Configure the NVMe Host to issue a Read command to the controller in order to read from the same LBA which was written to in step 1.
3. Verify that all received responses have all Reserved fields set to 0.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.
2. Verify that the known data pattern was read correctly from the NVMe Device exactly as it was written.

Case 9: Valid Write, LR=1, FUA=0 (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Write command with a known data pattern, to the controller in order to write a known data pattern to one LBA on the NVMe Device, with LR=1, FUA=0.
2. Configure the NVMe Host to issue a Read command to the controller in order to read from the same LBA which was written to in step 1.
3. Verify that all received responses have all Reserved fields set to 0.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.
2. Verify that the known data pattern was read correctly from the NVMe Device exactly as it was written.

Case 10: Valid Write, LR=1, FUA=1 (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Write command with a known data pattern, to the controller in order to write a known data pattern to one LBA on the NVMe Device, with LR=1, FUA=1.
2. Configure the NVMe Host to issue a Read command to the controller in order to read from the same LBA which was written to in step 1.
3. Verify that all received responses have all Reserved fields set to 0.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.
2. Verify that the known data pattern was read correctly from the NVMe Device exactly as it was written.

©2021 UNH–IOL

Case 11: NSID=FFFFFFFFh, LR=0, FUA=0 (FYI)

Test Procedure:

1. Check the NVMe specification version supported by the DUT. If the specification is not v1.4 or higher, this test is not applicable.
2. Configure the NVMe Host to issue a Write command to the controller with NSID=FFFFFFFFh in order to write a known data pattern to one LBA on the NVMe Device, with LR=0, FUA=0.

Observable Results:

1. Verify that the Write command did not complete successfully.

Case 12: Write Exceeds Maximum Active Resources (FYI)

Test Procedure:

1. Configure the Testing Station to issue an Identify Command for CNS 05h, I/O Command Set Specific Identify Namespace Data Structures and record the Maximum Active Resources value (MAR) and the Maximum Open Resources value (MOR). If the MAR field is set to FFFFFFFFh this test is not applicable.
2. Configure the Testing Station to put the DUT into the ZSE state using a Zone Management Send command with Action of Reset.
3. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
4. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Open Zone and Select All=1, and a valid SLBA. Repeat this step for different zones until the number of active zones is 1 less than the MAR value. If opening a zone will exceed the MOR value, some zones may be closed with a Zone Management Send command with action Close Zone, which will decrease the Open Resources count but maintain the Active Resources count.
5. Configure the Testing Station acting as a Host to send a Write command for a new zone.

Observable Results:

1. Verify that the Write command is aborted with a status code of “Too Many Active Zones” BDh.

Case 13: Write Exceeds Maximum Open Resources (FYI)

Test Procedure:

1. Configure the Testing Station to issue an Identify Command for CNS 05h, I/O Command Set Specific Identify Namespace Data Structures and record the Maximum Open Resources value (MOR). If the MOR field is set to FFFFFFFFh this test is not applicable.
2. Configure the Testing Station to put the DUT into the ZSE state using a Zone Management Send command with Action of Reset.
3. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
4. Configure the Testing Station acting as a Host to issue a Write command with a valid SLBA. Repeat this step to different zones until the number of implicitly opened zones exceeds the MOR value.

Observable Results:

1. If MAR is greater than MOR, verify that the Write command which causes the number of Open Resources to exceed the value specified by the Maximum Open Resources field is aborted with a status code of Too Many Open Zones BEh.
2. If MAR is equal to MOR, verify that the Write command which causes the number of Open Resources to exceed the value specified by the Maximum Open Resources field is aborted with a status code of Too Many Active Zones BDh.

Possible Problems: None.

©2021 UNH–IOL

Test 3.8 – Write Uncorrectable Command (FYI)

Purpose: To verify that an NVMe Controller can properly execute the Write Uncorrectable command.

References:

[1] NVMe Specification 6.15
[2] NVMe ZNS Specification 3.3.8

Resource Requirements:

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

Last Modification: June 10, 2021

Discussion: The Write Uncorrectable command is used to mark a range of logical blocks as invalid. When the specified logical blocks(s) are read after this operation, a failure is returned with Unrecovered Read Error status. To clear the invalid logical block status, a write operation is performed for those logical blocks.

The fields used are Command Dword 10, Command Dword 11, and Command Dword 12 fields. All other command specific fields are reserved.

Test Setup: See Appendix A.

Case 1: SLBA In Range, NLB Valid (FYI)

Test Procedure:

1. Check the ONCS field or check for a non-zero value in the WUSL field to determine if the DUT supports the Write Uncorrectable command. If the DUT does not support the Write Uncorrectable command this test is not applicable.
2. Configure the NVMe Host to issue a Write Uncorrectable command to a particular LBA on the NVMe Device. Both the SLBA and NLB should be valid.
3. Configure the NVMe Host to issue a Read command for the LBA on which the Write Uncorrectable command was performed.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.
2. Verify that the completion queue entry returned for the Read command indicates *Unrecovered Read Error* status.

Case 2: SLBA Out of Range, NLB Valid (FYI)

Test Procedure:

1. Check the ONCS field or check for a non-zero value in the WUSL field to determine if the DUT supports the Write Uncorrectable command. If the DUT does not support the Write Uncorrectable command this test is not applicable.
2. Configure the NVMe Host to issue a Write Uncorrectable command to a particular LBA on the NVMe Device. The SLBA should be for a value out of range for the DUT. NLB should be valid.
3. Configure the NVMe Host to issue a Read command for the LBA on which the Write Uncorrectable command was performed.

Observable Results:

1. Verify that that the Write Uncorrectable Command returns status code LBA Out of Range 80h.
2. Verify that the completion queue entry returned for the first Read command indicates Status Code LBA out of Range and does not return status code Unrecovered READ error.

©2021 UNH–IOL

Case 3: SLBA Out of Range, NSID Invalid (FYI)

Test Procedure:

1. Check the ONCS field or check for a non-zero value in the WUSL field to determine if the DUT supports the Write Uncorrectable command. If the DUT does not support the Write Uncorrectable command this test is not applicable.
2. Check the NN value reported by the DUT in the Identify Controller Data Structure. If NN is the maximum possible value (0xFFFFFFFF) this sub test cannot be performed.
3. Configure the NVMe Host to issue a Write Uncorrectable command to a particular LBA on the NVMe Device. The SLBA should be for a value out of range for the DUT, and the NSID should be invalid. The Write Uncorrectable command to the NVMe Controller should specify an Invalid NSID outside of the controller specified range defined by NN (Number of Namespaces).
4. Configure the NVMe Host to issue a Read command for the LBA on which the Write Uncorrectable command was performed.

Observable Results:

1. Verify that the Write Uncorrectable command completes with status code Invalid Namespace or Format (0Bh), and not LBA Out of Range (80h).
2. Verify that the completion queue entry returned for the first Read command indicates Status Code Invalid Namespace or Format and does not return status code Unrecovered READ error.

Case 4: SLBA Out of Range, but Lower Dword = 00000000 (FYI)

Test Procedure:

1. Check the ONCS field or check for a non-zero value in the WUSL field to determine if the DUT supports the Write Uncorrectable command. If the DUT does not support the Write Uncorrectable command this test is not applicable.
2. Configure the NVMe Host to issue a Write Uncorrectable command to a particular LBA on the NVMe Device. The SLBA should be FFFFFFFF00000000h ,which, when read as a 64 bit value is out of range of the DUT, and an NLB which is less than MDTS.
3. Configure the NVMe Host to issue a Read command for the LBA on which the Write Uncorrectable command was performed.

Observable Results:

1. Verify that the Write Uncorrectable command completes with status code LBA Out of Range (80h).
2. Verify that the completion queue entry returned for the first Read command indicates Status Code LBA out of Range and does not return status code Unrecovered READ error.

Case 5: NLB greater than MDTS and Non-MDTS Command Size Limits Not Supported (FYI)

Test Procedure:

1. Check the WUSL field. If this field is set to a non-zero value this test is not applicable.
2. Check the ONCS field to determine if the DUT supports the Write Uncorrectable command. If the DUT does not support the Write Uncorrectable command this test is not applicable.
3. Check the MDTS value reported by the DUT in the Identify Controller Data Structure. If MDTS is set to 0 (unlimited) then this test case is not applicable.
4. Configure the NVMe Host to issue a Write Uncorrectable command to a particular LBA on the NVMe Device. NLB should be greater than MDTS.
5. Configure the NVMe Host to issue Read commands for the LBAs on which the Write Uncorrectable command was performed. Ensure that the NVMe Host issues Read commands to all LBAs affected by the Write Uncorrectable command.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status of Successful for the Write Uncorrectable command. The DUT is expected to ignore the NLB/MDTS conflict since MDTS does not affect the Write Uncorrectable command.

©2021 UNH–IOL

2. Verify that the completion queue entry returned for the Read commands indicate *Unrecovered Read Error* status.

Case 6: NSID=FFFFFFFFh, SLBA In Range, NLB Valid (FYI)

Test Procedure:

1. Check the ONCS field or check for a non-zero value in the WUSL field to determine if the DUT supports the Write Uncorrectable command. If the DUT does not support the Write Uncorrectable command this test is not applicable.
2. Check the NVMe specification version supported by the DUT. If the specification is not v1.4 or higher, this test is not applicable.
3. Configure the NVMe Host to issue a Write Uncorrectable command with NSID=FFFFFFFFh to a particular LBA on the NVMe Device. Both the SLBA and NLB should be valid.

Observable Results:

1. Verify that the Write Uncorrectable command does not complete successfully.

Case 7: NLB greater than WUSL and Non-MDTS Command Size Limits Supported (FYI)

Test Procedure:

1. Check the WUSL value reported by the DUT in the Identify Controller Data Structure. If WUSL is set to 0 (unlimited) then this test case is not applicable.
2. Configure the NVMe Host to issue a Write Uncorrectable command to a particular LBA on the NVMe Device. NLB should be greater than WUSL.

Observable Results:

1. Verify that after the completion of the Write Uncorrectable command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status of Invalid Field in command. for the Write Uncorrectable command.

Case 8: Zone Boundary Error (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSIO state.
2. Configure the Testing Station acting as a Host to issue a Write Uncorrectable command to logical blocks in more than one zone.

Observable Results:

1. Verify that the Write Uncorrectable command is aborted with a status code of “Zone Boundary Error” B8h.

Case 9: Zone is Full (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSF state.
2. Configure the Testing Station acting as a Host to issue a Write Uncorrectable command to an LBA in the zone which is in ZSF state.

Observable Results:

1. Verify that the Write Uncorrectable command is aborted with a status code of “Zone is Full” B9h.

Case 10: Zone Invalid Write (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSIO state.
2. Configure the Testing Station acting as a Host to issue a Write Uncorrectable command to an LBA different than the write pointer.

©2021 UNH–IOL

Observable Results:

1. Verify that the Write Uncorrectable command is aborted with a status code of "Zone Invalid Write" BCh.

Case 11: Write Uncorrectable Exceeds Maximum Active Resources (FYI)

Test Procedure:

1. Configure the Testing Station to issue an Identify Command for CNS 05h, I/O Command Set Specific Identify Namespace Data Structures and record the Maximum Active Resources value (MAR) and the Maximum Open Resources value (MOR). If the MAR field is set to FFFFFFFFh this test is not applicable.
2. Configure the Testing Station to put the DUT into the ZSE state using a Zone Management Send command with Action of Reset.
3. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
4. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Open Zone and Select All=1, and a valid SLBA. Repeat this step for different zones until the number of active zones is 1 less than the MAR value. If opening a zone will exceed the MOR value, some zones may be closed with a Zone Management Send command with action Close Zone, which will decrease the Open Resources count but maintain the Active Resources count.
5. Configure the Testing Station acting as a Host to send a Write Uncorrectable command for a new zone.

Observable Results:

1. Verify that the Write Uncorrectable command is aborted with a status code of "Too Many Active Zones" BDh.

Case 12: Write Uncorrectable Exceeds Maximum Open Resources (FYI)

Test Procedure:

1. Configure the Testing Station to issue an Identify Command for CNS 05h, I/O Command Set Specific Identify Namespace Data Structures and record the Maximum Open Resources value (MOR). If the MOR field is set to FFFFFFFFh this test is not applicable.
2. Configure the Testing Station to put the DUT into the ZSE state using a Zone Management Send command with Action of Reset.
3. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
4. Configure the Testing Station acting as a Host to issue a Write Uncorrectable command with a valid SLBA. Repeat this step to different zones until the number of implicitly opened zones exceeds the MOR value.

Observable Results:

1. If MAR is greater than MOR, verify that the Write Uncorrectable command which causes the number of Open Resources to exceed the value specified by the Maximum Open Resources field is aborted with a status code of Too Many Open Zones BEh.
2. If MAR is equal to MOR, verify that the Write Uncorrectable command which causes the number of Open Resources to exceed the value specified by the Maximum Open Resources field is aborted with a status code of Too Many Active Zones BDh.

Possible Problems:

©2021 UNH–IOL

Test 3.9 – Write Zeroes Command (FYI)

Purpose: To verify that an NVMe Controller can properly execute the Write Zeroes command.

References:

[1] NVMe Specification 6.16

Resource Requirements:

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

Last Modification: November 20, 2018

Discussion: The Write Zeroes command is used to set a range of logical blocks to zero. After successful completion of this command, the value returned by subsequent reads of logical blocks in this range shall be zeroes until a write occurs to this LBA range. The metadata for this command shall be all zeroes and the protection information is updated based on CDW12.PRINFO.

The fields used are Command Dword 10, Command Dword 11, Command Dword 12, Command Dword 14, and Command Dword 15 fields.

Test Setup: See Appendix A.

Case 1: SLBA In Range, NLB Valid, LR=0, FUA=0 (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data Structure or check for a non-zero value in the WZSL field to determine if the Write Zeroes command is supported. If the Write Zeroes Command is not supported then this test is not applicable.
2. Configure the NVMe Host to issue a Write Zeroes command to a specific LBA on the NVMe Device.
3. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA for which the Write Zeroes command was performed.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.
2. Verify that the data returned by the controller for the Read command after the Write Zeroes command returns data of zeroes only.

Case 2: SLBA Out of Range, NLB Valid (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data Structure or check for a non-zero value in the WZSL field to determine if the Write Zeroes command is supported. If the Write Zeroes Command is not supported then this test is not applicable.
2. Configure the NVMe Host to issue a Write Zeroes command to the controller specifying SLBA which is out of range for the DUT.

Observable Results:

1. Verify that that the Write Zeroes Command returns status code LBA Out of Range 80h.

Case 3: SLBA Out of Range, NSID Invalid (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data Structure or check for a non-zero value in the WZSL field to determine if the Write Zeroes command is supported. If the Write Zeroes Command is not supported then this test is not applicable.

©2021 UNH–IOL

2. Check the NN value reported by the DUT in the Identify Controller Data Structure. If NN is the maximum possible value (0xFFFFFFFF) this sub test cannot be performed.
3. Configure the NVMe Host to issue a Write Zeroes command to the controller specifying SLBA which is out of range for the DUT, and an invalid NSID. The Write Zeroes command to the NVMe Controller should specify an Invalid NSID outside of the controller specified range defined by NN (Number of Namespaces).

Observable Results:

1. Verify that the Write Zeroes command completes with status code or either Invalid Namespace or Format (0Bh or LBA Out of Range (80h).

Case 4: SLBA Out of Range, but Lower Dword = 00000000 (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data Structure or check for a non-zero value in the WZSL field to determine if the Write Zeroes command is supported. If the Write Zeroes Command is not supported then this test is not applicable.
2. Configure the NVMe Host to issue a Write Zeroes command to a particular LBA on the NVMe Device. The SLBA should be FFFFFFFF00000000h ,which, when read as a 64 bit value is out of range of the DUT, and an NLB which is less than MDTS.

Observable Results:

1. Verify that the Write Zeroes command completes with status code LBA Out of Range (80h).

Case 5: NLB greater than MDTS and Non-MDTS Command Size Limits Not Supported (FYI)

Test Procedure:

1. Check the WZSL field. If this field is set to a non-zero value this test is not applicable.
2. Check the ONCS field of the Identify Controller Data Structure. If the Write Zeroes Command is not support then this test is not applicable.
3. Check the MDTS value reported in the Identify Controller Data Structure, If MDTS is set to 0 (unlimited) then this test case is not applicable.
4. Configure the NVMe Host to issue a Read command to the controller specifying a given LBA.
5. Configure the NVMe Host to issue a Write Zeroes command the tsame LBA on the NVMe Device. NLB should be greater than MDTS.
6. Configure the NVMe Host to issue Read commands to the controller specifying the same LBAs for which the Write Zeroes command was performed.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command which should be ‘Successful’.
2. Verify that the data returned by the controller for the Read commands after the Write Zeroes command returns all 0’s.

Case 6: SLBA In Range, NLB Valid, LR=0, FUA=1 (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data Structure or check for a non-zero value in the WZSL field to determine if the Write Zeroes command is supported. If the Write Zeroes Command is not supported then this test is not applicable.
2. Configure the NVMe Host to issue a Read command to the controller specifying a given LBA.
3. Configure the NVMe Host to issue a Write Zeroes command to the same LBA on the NVMe Device.
4. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA for which the Write Zeroes command was performed.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.

©2021 UNH–IOL

2. Verify that the data returned by the controller for the Read command after the Write Zeroes command returns data of zeroes only.

Case 7: SLBA In Range, NLB Valid, LR=1, FUA=0 (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data Structure or check for a non-zero value in the WZSL field to determine if the Write Zeroes command is supported. If the Write Zeroes Command is not supported then this test is not applicable.
2. Configure the NVMe Host to issue a Read command to the controller specifying a given LBA.
3. Configure the NVMe Host to issue a Write Zeroes command the same LBA on the NVMe Device.
4. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA for which the Write Zeroes command was performed.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.
2. Verify that the data returned by the controller for the Read command after the Write Zeroes command returns data of zeroes only.

Case 8: SLBA In Range, NLB Valid, LR=1, FUA=1 (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data Structure or check for a non-zero value in the WZSL field to determine if the Write Zeroes command is supported. If the Write Zeroes Command is not supported then this test is not applicable.
2. Configure the NVMe Host to issue a Read command to the controller specifying a given LBA.
3. Configure the NVMe Host to issue a Write Zeroes command to the same LBA on the NVMe Device.
4. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA for which the Write Zeroes command was performed.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command.
2. Verify that the data returned by the controller for the Read command after the Write Zeroes command returns data of zeroes only.

Case 9: PRCHK is Non Zero (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data Structure or check for a non-zero value in the WZSL field to determine if the Write Zeroes command is supported. If the Write Zeroes Command is not supported then this test is not applicable.
2. Check for support for End to End Data Protection by checking the DPC field in the Identify Namespace Data Structure. If the DUT does not support End to End Data Protection, this test is not applicable.
3. Perform a Format NVM command to enable the namespace to use end to end data protection. End to End data protection must align with the support indicated in the DPC field.
4. Configure the NVMe Host to issue a Read command to the controller specifying a given LBA.
5. Configure the NVMe Host to issue a Write Zeroes command to the same LBA on the NVMe Device with the PRCHK value within the PRINFO field set to a non-zero value.
6. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA for which the Write Zeroes command was performed.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command, and that the Write Zeroes command completes with Status 02h Invalid Field or Status 81h Invalid Protection Information.

©2021 UNH–IOL

2. Verify that the data returned by the controller for the second Read command after the Write Zeroes command returns the same data returned in response to the first Read command. This may not be observable if the first Read command returns data of all zeroes.

Case 10: NSID=FFFFFFFFh, SLBA In Range, NLB Valid, LR=0, FUA=0 (FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data Structure or check for a non-zero value in the WZSL field to determine if the Write Zeroes command is supported. If the Write Zeroes Command is not supported then this test is not applicable.
2. Check the NVMe specification version supported by the DUT. If the specification is not v1.4 or higher, this test is not applicable.
3. Configure the NVMe Host to issue a Read command to the controller specifying a given LBA.
4. Configure the NVMe Host to issue a Write Zeroes command with NSID=FFFFFFFFh to the controller specifying the same LBA on the NVMe Device which was previously read from.
5. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA for which the Write Zeroes command was performed.

Observable Results:

1. Verify that the Write Zeroes command did not complete successfully.
2. Verify that the data returned by the controller for the second Read command after the Write Zeroes command returns the same data returned in response to the first Read command. This may not be observable if the first Read command returns data of all zeroes.

Case 11: NLB greater than WZSL and Non-MDTS Command Size Limits Supported (FYI)

Test Procedure:

1. Check the WZSL field. If this field is set to 0h this test is not applicable.
2. Check the ONCS field of the Identify Controller Data Structure. If the Write Zeroes Command is not supported then this test is not applicable.
3. Check the WZSL value reported in the Identify Controller Data Structure, If WZSL is set to 0 (unlimited) this test case is not applicable.
4. Configure the NVMe Host to issue a Write Zeroes command with NLB greater than WZSL.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command of 'Invalid Field in Command'.

Case 12: Zone Boundary Error (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSIO state.
2. Configure the Testing Station acting as a Host to issue a Write Zeroes command to logical blocks in more than one zone.

Observable Results:

1. Verify that the Write Zeroes command is aborted with a status code of "Zone Boundary Error" B8h.

Case 13: Zone is Full (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSF state.
2. Configure the Testing Station acting as a Host to issue a Write Zeroes command to an LBA in the zone which is in ZSF state.

Observable Results:

1. Verify that the Write Zeroes command is aborted with a status code of "Zone is Full" B9h.

©2021 UNH–IOL

Case 14: Zone Invalid Write (FYI)

Test Procedure:

1. Configure the Testing Station to put the DUT into the ZSIO state.
2. Configure the Testing Station acting as a Host to issue a Write Zeroes command to an LBA different than the write pointer.

Observable Results:

1. Verify that the Write Zeroes command is aborted with a status code of "Zone Invalid Write" BCh.

Case 15: Write Zeroes Exceeds Maximum Active Resources (FYI)

Test Procedure:

1. Configure the Testing Station to issue an Identify Command for CNS 05h, I/O Command Set Specific Identify Namespace Data Structures and record the Maximum Active Resources value (MAR) and the Maximum Open Resources value (MOR). If the MAR field is set to FFFFFFFFh this test is not applicable.
2. Configure the Testing Station to put the DUT into the ZSE state using a Zone Management Send command with Action of Reset.
3. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
4. Configure the Testing Station acting as a Host to send a Zone Management Send command with the action of Open Zone and Select All=1, and a valid SLBA. Repeat this step for different zones until the number of active zones is 1 less than the MAR value. If opening a zone will exceed the MOR value, some zones may be closed with a Zone Management Send command with action Close Zone, which will decrease the Open Resources count but maintain the Active Resources count.
5. Configure the Testing Station acting as a Host to send a Write Zeroes command for a new zone.

Observable Results:

1. If MAR is greater than MOR, verify that the Write Zeroes command which causes the number of Open Resources to exceed the value specified by the Maximum Open Resources field is aborted with a status code of Too Many Open Zones BEh.
2. If MAR is equal to MOR, verify that the Write Zeroes command which causes the number of Open Resources to exceed the value specified by the Maximum Open Resources field is aborted with a status code of Too Many Active Zones BDh.

Case 16: Write Zeroes Exceeds Maximum Open Resources (FYI)

Test Procedure:

1. Configure the Testing Station to issue an Identify Command for CNS 05h, I/O Command Set Specific Identify Namespace Data Structures and record the Maximum Open Resources value (MOR). If the MOR field is set to FFFFFFFFh this test is not applicable.
2. Configure the Testing Station to put the DUT into the ZSE state using a Zone Management Send command with Action of Reset.
3. Confirm the current Zone state by querying the Zone Descriptor using a Zone Management Receive.
4. Configure the Testing Station acting as a Host to issue a Write Zeroes command with a valid SLBA. Repeat this step to different zones until the number of implicitly opened zones exceeds the MOR value.

Observable Results:

1. Verify that the Write Zeroes command which causes the number of Open Resources to exceed the value specified by the Maximum Open Resources field is aborted with a status code of Too Many Open Zones BEh.

Possible Problems: None.

©2021 UNH–IOL

Group 4: Reservations Commands

Overview:

This section describes a method for performing conformance verification for NVMe ZNS products implementing Reservations.

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).

©2021 UNH–IOL

Test 4.1 – Reservation Report Command (FYI)

Purpose: To determine if an NVMe Controller properly reports the status of a reservation when processing a Reservation Report command.

References:

NVMe Specification 8.8, 5.14.1.15, 6.13, NVMe v1.3 ECN 003

Resource Requirements:

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

Last Modification: June 10, 2021

Discussion: A host may determine the current reservation status associated with a namespace by executing a Reservation Report command.

The Reservation Report command returns a Reservation Status data structure to memory that describes the registration and reservation status of a namespace.

The size of the Reservation Status data structure is a function of the number of controllers in the NVM Subsystem that are associated with hosts that are registrants of the namespace (i.e., there is a Registered Controller data structure for each such controller).

The command uses Command Dword 10. If the command uses PRPs for the data transfer, then PRP Entry 1 and PRP Entry 2 fields are used. If the command uses SGLs for the data transfer, then the SGL Entry 1 field is used. All other command specific fields are reserved.

Test Setup: See Appendix A.

Case 1: No Registrants (FYI)

Test Procedure:

1. Check the ONCS field to determine of the controller supports reservations. If the controller does not support reservations then this test is not applicable.
2. For each NVMe Controller in the NVM Subsystem:
    a. Configure the NVMe Host to issue a Set Features command with the Host Identifier feature to the NVMe Controller in order to set its Host Identifier for that controller.
    b. For each active namespace attached to the NVMe Controller:
        i. Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
        ii. Configure the NVMe Host to issue a Get Features command with the Reservation Persistence feature to the NVMe Controller.
3. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated Completion Queue indicating the status for the command.
2. Verify that the Reservation Type (RTYPE) field of the Reservation Status data structure is set to 0 to indicate that no reservation is held on the namespace.
3. Verify that the Number of Registered Controllers (REGCTL) field of the Reservation Status data structure is set to 0 to indicate that no hosts are registrants of the namespace.

©2021 UNH–IOL

4. Verify that the Persist Through Power Loss State (PTPLS) field of the Reservation Status data structure is set to the same value of the Persist Through Power Loss (PTPL) field of Command Dword 0 of the completion queue entry for the Get Features command with Reservation Persistence feature.
5. Verify that the controller supports the Host Identifier feature.

Case 2: Host is a Registrant (FYI)

Test Procedure:

1. Check the ONCS field to determine of the controller supports reservations. If the controller does not support reservations then this test is not applicable.
2. For each NVMe Controller in the NVM Subsystem:
    a. Configure the NVMe Host to issue a Set Features command with the Host Identifier feature to the NVMe Controller in order to set its Host Identifier for that controller.
    b. For each active namespace attached to the NVMe Controller:
        i. Configure the NVMe Host to issue a Reservation Register command with the Reservation Register Action (RREGA) field set to 000b (i.e. Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field to the NVMe Controller for the namespace in order to make the host a registrant of that namespace, and perform a Reservation Acquire to that namespace.
        ii. Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
        iii. Configure the NVMe Host to issue an Identify command specifying CNS value 01h to the NVMe Controller in order to receive back the Identify Controller data structure for that controller.
3. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated Completion Queue indicating the status for the command.
2. Verify that the Generation (GEN) field of the Reservation Status data structure is incremented for each Reservation Register command which the host issues to an NVMe Controller.
3. Verify that the Reservation Type (RTYPE) field of the Reservation Status data structure is set to not 0 to indicate that a reservation is held on the namespace.
4. Verify that the Number of Registered Controllers (REGCTL) field of the Reservation Status data structure is set to the number of controllers for which the host has set its Host Identifier for to indicate the number of controllers associated with the host and that the host is a registrant of the namespace.
5. Verify that the number of Registered Controller data structures returned as part of the Reservation Status data structure is exactly equal to the value stored in the REGCTL field.
6. Verify that the Controller ID (CNTLID) field of the Registered Controller data structures matches the CNTLID field in the Identify Controller data structure for that controller.
7. Verify that the Reservation Status (RCSTS) field of the Registered Controller data structures have bit 0 cleared to not '0' to indicate that the host associated with the controller holds a reservation on the namespace.
8. Verify that the Host Identifier (HOSTID) field of the Registered Controller data structures is set to the same value which the host set for its Host Identifier in the Set Features command. Verify that the Reservation Key (RKEY) field of the Registered Controller data structures is set to the same value which the host set for its reservation key in the Reservation Register command.
9. Verify that the controller supports the Host Identifier feature.

Case 3: 64 Bit Host Identifier (FYI)

Test Procedure:

1. Check the ONCS field to determine of the controller supports reservations. If the controller does not support reservations then this test is not applicable.

©2021 UNH–IOL

2. Determine if the Controller supports the Host Identifier feature identifier (81h).
3. For each NVMe Controller in the NVM Subsystem:
    a. Configure the NVMe Host to issue a Set Features command with the Host Identifier feature to the NVMe Controller in order to register a 64 bit Host Identifier for that controller.
    b. For each active namespace attached to the NVMe Controller:
        i. Configure the NVMe Host to issue a Reservation Register command with the Reservation Register Action (RREGA) field set to 000b (i.e. Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field to the NVMe Controller for the namespace in order to make the host a registrant of that namespace, and perform a Reservation Acquire to that namespace.
        ii. Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller with the Extended Data Structure bit set to 1 in Command Dword 11.
4. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

Observable Results:
1. Verify that the controller aborts the Reservation Report command with the status code of Host Identifier Inconsistent Format.
2. Verify that the controller supports the Host Identifier feature.

Case 4: 128 Bit Host Identifier (FYI)

Test Procedure:
1. Check the ONCS field to determine of the controller supports reservations. If the controller does not support reservations then this test is not applicable.
2. Check the CTRATT field to determine if the controller supports 128 bit Host identifiers. If the controller does not support 128 bit host identifiers then this test is not applicable.
3. Determine if the Controller supports the Host Identifier feature identifier (81h).
4. For each NVMe Controller in the NVM Subsystem:
    a. Configure the NVMe Host to issue a Set Features command with the Host Identifier feature to the NVMe Controller in order to register a 128 bit Host Identifier for that controller.
    b. For each active namespace attached to the NVMe Controller:
        i. Configure the NVMe Host to issue a Reservation Register command with the Reservation Register Action (RREGA) field set to 000b (i.e. Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field to the NVMe Controller for the namespace in order to make the host a registrant of that namespace, and perform a Reservation Acquire to that namespace.
        ii. Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller with the Extended Data Structure bit set to 0 in Command Dword 11.
5. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

Observable Results:
1. Verify that the controller supports the Host Identifier feature.
2. Verify that the controller aborts the Reservation Report command with the status code of Host Identifier Inconsistent Format.

Case 5: Dynamic Controller Not Associated with Host (FYI)

Test Procedure:
1. Check the ONCS field to determine of the controller supports reservations. If the controller does not support reservations then this test is not applicable.
2. For each NVMe Controller in the NVM Subsystem:

©2021 UNH–IOL

   a. Check that the controller is a dynamic controller (as defined in the NVMe-oF specification). If the controller is not a dynamic controller then this test is not applicable.
   b. Configure the NVMe Host to issue a Reservation Report command to the dynamic NVMe Controller.
3. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

Observable Results:
1. Verify that the controller sets the Controller ID field to FFFFh.

Possible Problems: None.

©2021 UNH–IOL

Test 4.2 – Reservation Registration (FYI)

Purpose: To determine if an NVMe Controller properly supports registering hosts via the Reservation Register command.

References:

NVMe Specification 8.8 , 5.14.1.15, 6.11

Resource Requirements:

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

Last Modification: June 10, 2021

Discussion: Prior to establishing a reservation on a namespace, a host shall become a registrant of that namespace by registering a reservation key. Registering a reservation key with a namespace creates an association between a host and a namespace. A host need only register on a single controller in order to become a registrant of the namespace on all controllers in the NVM Subsystem that have access to the namespace and are associated with the host.

A host registers a reservation key by executing a Reservation Register command on the namespace with Reservation Register Action (RREGA) field set to 000b (i.e., Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field.

The Reservation Register command uses Command Dword 10 and a Reservation Register data structure in memory. If the command uses PRPs for the data transfer, then the PRP Entry 1 and PRP Entry 2 fields are used. If the command uses SGLs for the data transfer, then the SGL Entry 1 field is used. All other command specific fields are reserved.

Test Setup: See Appendix A.

Case 1: Basic Operation (FYI)

Test Procedure:

1. Check the ONCS field to determine of the controller supports reservations. If the controller does not support reservations then this test is not applicable.
2. For each NVMe Controller in the NVM Subsystem:
    a. Configure the NVMe Host to issue a Set Features command with the Host Identifier feature to the NVMe Controller in order to set its Host Identifier for that controller.
    b. For each shared namespace attached to the NVMe Controller:
        i. Configure the NVMe Host to issue a Reservation Register command with the Reservation Register Action (RREGA) field set to 000b (i.e. Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field to the NVMe Controller for the namespace in order to make the host a registrant of that namespace.
        ii. Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
3. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated Completion Queue indicating the status for the command.
2. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the Reservation Report command indicates that the host was successfully registered to the namespace.

Case 2: Re-registration (FYI)

©2021 UNH–IOL

Test Procedure:

1. Check the ONCS field to determine of the controller supports reservations. If the controller does not support reservations then this test is not applicable.
2. For each NVMe Controller in the NVM Subsystem:
    a. Configure the NVMe Host to issue a Set Features command with the Host Identifier feature to the NVMe Controller in order to set its Host Identifier for that controller.
    b. For each shared namespace attached to the NVMe Controller:
        i. Configure the NVMe Host to issue a Reservation Register command with the Reservation Register Action (RREGA) field set to 000b (i.e. Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field to the NVMe Controller for the namespace in order to make the host a registrant of that namespace.
        ii. Configure the NVMe Host to issue an additional Reservation Register command with Register Reservation Key action and the same reservation key to the NVMe Controller.
        iii. Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
        iv. Configure the NVMe Host to issue an additional Reservation Register command with Register Reservation Key action and a different reservation key to the NVMe Controller.
        v. Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
3. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated Completion Queue indicating the status for the command.
2. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the Reservation Report command indicates that the host was successfully registered to the namespace.
3. Verify that the completion queue entry for the final Reservation Register command (with the different reservation key) indicates status Reservation Conflict.
4. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the final Reservation Report command indicates that the reservation key for the host was not changed.

Case 3: Replace Registration Key (FYI)

A host that is a registrant of a namespace may replace its existing reservation key by executing a Reservation Register command on the namespace with the RREGA field set to 010b (i.e., Replace Reservation Key), supplying the current reservation key in the Current Reservation Key (CRKEY) field, and the new reservation key in the NRKEY field. If the contents of the CRKEY field do no match the key currently associated with the host, then the Reservation Register command shall be aborted with status of Reservation Conflict. A host may replace its reservation key without regard to its registration status or current reservation key value by setting the Ignore Existing Key (IEKEY) bit to '1' in the Reservation Register command. Setting the IEKEY bit to '1' causes the Reservation Register command to succeed regardless of the value of the CRKEY field (i.e., the current reservation key is not checked).

Replacing a reservation key has no effect on any reservation that may be held on the namespace.

Test Procedure:

1. Check the ONCS field to determine of the controller supports reservations. If the controller does not support reservations then this test is not applicable.
2. For each NVMe Controller in the NVM Subsystem:
    a. Configure the NVMe Host to issue a Set Features command with the Host Identifier feature to the NVMe Controller in order to set its Host Identifier for that controller.
    b. For each shared namespace attached to the NVMe Controller:
        i. Configure the NVMe Host to issue a Reservation Register command with the Reservation Register Action (RREGA) field set to 000b (i.e. Register Reservation Key) and supplying

©2021 UNH–IOL

a reservation key in the New Reservation Key (NRKEY) field to the NVMe Controller for the namespace in order to make the host a registrant of that namespace.

ii. Configure the NVMe Host to issue a Reservation Register command with the RREGA field set to 010b (i.e. Replace Reservation Key), supplying the current reservation key in the Current Reservation Key (CRKEY) field, and a new reservation key in the NRKEY field to the NVMe Controller for the namespace in order to associate a new reservation key with the registrant of the namespace.

iii. Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.

iv. Configure the NVMe Host to issue a Reservation Register command with the Replace Reservation Key action, supplying any key value that is not the current reservation key in the CRKEY field, and a new reservation key in the NRKEY field to the NVMe Controller. Also, set the RType to 2 using the Reservation Acquire Command.

v. Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.

vi. Configure the NVMe Host to issue a Reservation Acquire command, setting the Reservation Acquire Action (RACQA) field to 000b (Acquire), and supplying the current reservation key associated with the host in the Current Reservation Key (CRKEY) field to the NVMe Controller in order for the host to acquire a reservation on the namespace.

vii. Configure the NVMe Host to issue a Reservation Register command with the Register Reservation Key action, supplying a new reservation key in the New Reservation Key (NRKEY) field, and setting the IEKEY bit to '1' to the NVMe Controller for the namespace in order to make the host a registrant of that namespace with a different reservation key.

viii. Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.

3. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated Completion Queue indicating the status for the command.
2. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the Reservation Report command after the second Reservation Registration command indicates that the host successfully changed its reservation key.
3. Verify that the completion queue entry for the third Reservation Register command (with the invalid current reservation key) indicates status Reservation Conflict.
4. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the Reservation Report command after the third Reservation Register command indicates that the reservation key associated with the host did not change.
5. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the Reservation Report command after the fourth Reservation Register command (with the IEKEY set) indicates that the host successfully changed its reservation key and that it still holds the reservation it previously acquired.

Case 4: Multiple Hosts (FYI) Dual Port Devices Only

There are no restrictions on the reservation key value used by hosts with different Host Identifiers. For example, multiple hosts may all register the same reservation key value.

This test requires there to be at least two controllers with a shared namespace in the NVM Subsystem.

Test Procedure:

1. Check the ONCS field to determine of the controller supports reservations. If the controller does not support reservations then this test is not applicable.
2. Configure NVMe Host 1 to issue a Set Features command with the Host Identifier feature to NVMe Controller 1 in order to set its Host Identifier for that controller.

©2021 UNH–IOL

3. Configure NVMe Host 2 to issue a Set Features command with the Host Identifier feature to NVMe Controller 2 with a different Host Identifier than NVMe Host 1.
4. Configure NVMe Host 1 to issue a Reservation Register command with the Reservation Register Action (RREGA) field set to 000b (i.e. Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field to NVMe Controller 1 for the shared namespace in order to make the host a registrant of that namespace.
5. Configure NVMe Host 2 to issue a Reservation Register command with the Register Reservation Key action and supplying the same reservation key that NVMe Host 1 used in the NRKEY field to NVMe Controller 2 for the shared namespace in order to make the host a registrant of that namespace.
6. Configure NVMe Host 1 to issue a Reservation Report command to NVMe Controller 1.
7. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated Completion Queue indicating the status for the command.
2. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the Reservation Report command indicates that both hosts were successfully registered to the namespace.

Case 5: Reservation Persistence (FYI)

Test Procedure:

1. Check the ONCS field to determine of the controller supports reservations. If the controller does not support reservations then this test is not applicable.
2. For each NVMe Controller in the NVM Subsystem, configure the NVMe Host to issue a Set Features command with the Host Identifier feature to the NVMe Controller in order to set its Host Identifier for that controller.
3. For each shared namespace attached to the NVMe Controller:
    a. Configure the NVMe Host to issue a Set Features command with the Reservation Persistence feature in order to set PTPL to 0 for that Namespace. If this feature is not changeable then this test is not applicable.
    b. Configure the NVMe Host to issue a Get Features command for the Reservation Persistence feature. Verify that the previously supplied value was returned.
    c. Configure the NVMe Host to issue a Reservation Register command with the Reservation Register Action (RREGA) field set to 000b (i.e. Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field to the NVMe Controller for the namespace in order to make the host a registrant of that namespace, and set CPTPL to 00b (No change to PTPL state).
    d. Configure the NVMe Host to issue a Get Features command for the Reservation Persistence feature. Verify that the previously supplied value for PTPL (0) was returned.
    e. Configure the NVMe Host to issue a Reservation Register command with the Reservation Register Action (RREGA) field set to 000b (i.e. Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field to the NVMe Controller for the namespace in order to make the host a registrant of that namespace, and set CPTPL to 11b (Set PTPL state to ‘1’).
    f. Configure the NVMe Host to issue a Get Features command for the Reservation Persistence feature. Verify that the value returned for PTPL is 1.
    g. Configure the NVMe Host to issue a Reservation Register command with the Reservation Register Action (RREGA) field set to 000b (i.e. Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field to the NVMe Controller for the namespace in order to make the host a registrant of that namespace, and set CPTPL to 10b (Set PTPL state to ‘0’).
    h. Configure the NVMe Host to issue a Get Features command for the Reservation Persistence feature. Verify that the value returned for PTPL is 0.
    i. Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
4. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

©2021 UNH–IOL

Observable Results:

1. Verify that in steps 3b, 3e, and 3i, the DUT returned a value of 0 for the PTPL value in the Get Feature response.
2. Verify that in step 3g, the DUT returned a value of 1 for the PTPL value in the Get Feature response.

Possible Problems: None.

©2021 UNH–IOL

Test 4.3 – Acquiring a Reservation (FYI)

Purpose: To determine if an NVMe Controller properly allows acquisition of reservations via the Reservation Register command.

References:

NVMe Specification 8.8, 5.14.1.15, 6.10

Resource Requirements:

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

Last Modification: June 10, 2021

Discussion: In order for a host to obtain a reservation on a namespace, it shall be a registrant of that namespace. A registrant obtains a reservation by executing a Reservation Acquire command, setting the Reservation Acquire Action (RACQA) field to 000b (Acquire), and supplying the current reservation key associated with the host to the Current Reservation Key (CRKEY) field.

The Reservation Acquire command uses Command Dword 10 and a Reservation Acquire data structure in memory. If the command uses PRPs for the data transfer, then the PRP Entry 1 and PRP Entry 2 fields are used. If the command uses SGLs for the data transfer, then the SGL Entry 1 field is used. All other command specific fields are used.

Test Setup: See Appendix A.

Case 1: Basic Operation (FYI)

Test Procedure:

1. Check the ONCS field to determine of the controller supports reservations. If the controller does not support reservations then this test is not applicable.
2. For each NVMe Controller in the NVM Subsystem:
    a. Configure the NVMe Host to issue a Set Features command with the Host Identifier feature to the NVMe Controller in order to set its Host Identifier for that controller.
    b. For each active namespace attached to the NVMe Controller:
        i. Configure the NVMe Host to issue a Reservation Register command with the Reservation Register Action (RREGA) field set to 000b (i.e. Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field to the NVMe Controller for the namespace in order to make the host a registrant of that namespace.
        ii. Configure the NVMe Host to issue an Identify command specifying CNS value 00h to the NVMe Controller in order to receive back the Identify Namespace data structure for the namespace.
        iii. For each reservation type supported by the namespace based on the RESCAP field of the Identify Namespace data structure for the namespace:
            1. Configure the NVMe Host to issue a Reservation Acquire command, setting the Reservation Acquire Action (RACQA) field to 000b (Acquire), supplying the current reservation key associated with the host in the Current Reservation Key (CRKEY) field, and setting the Reservation Type (RTYPE) field to the reservation type to the NVMe Controller in order for the host to acquire a reservation on the namespace with the reservation type.
            2. Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
            3. Configure the NVMe Host to issue a Reservation Release command, setting the Reservation Release Action (RRELA) field to 000b (i.e. Release), setting the Reservation Type (RTYPE) field to the type of the reservation being released, and supplying the current reservation key associated with the host in the Current

©2021 UNH–IOL

Reservation Key (CRKEY) field to the NVMe Controller in order to release the reservation held by the host.

3. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated Completion Queue indicating the status for the command.
2. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the Reservation Report command indicates that the host successfully acquired the reservation with the associated reservation type.

Case 2: Error Conditions (FYI)

If the CRKEY value does not match that used by the registrant to register with the namespace or the host is not a registrant, the command shall be aborted with status Registration Conflict. A Host may acquire a reservation without regard to its current reservation key value by setting the Ignore Existing Key (IEKEY) bit to '1' in the command.

If a reservation holder attempts to obtain a reservation of a different type on a namespace for which it is already the reservation holder, then the command shall be aborted with status Reservation Conflict. It is not an error for a reservation holder to attempt to obtain a reservation of the same type on a namespace for which it is already the reservation holder.

Test Procedure:

1. Check the ONCS field to determine of the controller supports reservations. If the controller does not support reservations then this test is not applicable.
2. For each NVMe Controller in the NVM Subsystem:
    a. Configure the NVMe Host to issue a Set Features command with the Host Identifier feature to the NVMe Controller in order to set its Host Identifier for that controller.
    b. For each active namespace attached to the NVMe Controller:
        i. RES_ACQ1: Configure the NVMe Host to issue a Reservation Acquire command, setting the Reservation Acquire Action (RACQA) field to 000b (Acquire), supplying a random reservation key in the Current Reservation Key (CRKEY) field, and setting the Reservation Type (RTYPE) field to a reservation type supported by the namespace to the NVMe Controller.
        ii. RES_REP1: Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
        iii. Configure the NVMe Host to issue a Reservation Register command with the Reservation Register Action (RREGA) field set to 000b (i.e. Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field to the NVMe Controller for the namespace in order to make the host a registrant of that namespace.
        iv. RES_ACQ2: Configure the NVMe Host to issue a Reservation Acquire command, setting the Acquire action, supplying any reservation key other than the reservation key currently associated with the host in the CRKEY field, and setting the RTYPE field to a reservation type supported by the namespace to the NVMe Controller.
        v. RES_REP2: Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
        vi. RES_ACQ3: Configure the NVMe Host to issue a Reservation Acquire command, setting the Acquire action, setting the IEKEY bit to '1', and setting the RTYPE field to a reservation type supported by the namespace to the NVMe Controller in order for the host to acquire a reservation on the namespace.
        vii. RES_REP3: Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
        viii. RES_ACQ4: Configure the NVMe Host to issue a Reservation Acquire command, setting the Acquire action, supplying the reservation key currently associated with the host in the CRKEY field, and setting the RTYPE field to a different reservation type supported by the

©2021 UNH–IOL

namespace than the one used for the current reservation held by the host to the NVMe Controller.

ix. RES_REP4: Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.

x. RES_ACQ5: Configure the NVMe Host to issue a Reservation Acquire command, setting the Acquire action, supplying the reservation key currently associated with the host in the CRKEY field, and setting the RTYPE field to the same reservation type used for the current reservation held by the host to the NVMe Controller.

xi. RES_REP5: Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.

3. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated Completion Queue indicating the status for the command.
2. Verify that the completion queue entry for RES_ACQ1 indicates status Reservation Conflict.
3. Verify that the Reservation Status data structure returned by the NVMe Controller after completing RES_REP1 indicates that the host does not hold any reservations.
4. Verify that the completion queue entry for RES_ACQ2 indicates status Reservation Conflict.
5. Verify that the Reservation Status data structure returned by the NVMe Controller after completing RES_REP2 indicates that the host does not hold any reservations.
6. Verify that the completion queue entry for RES_ACQ3 indicates status "Invalid Field in Command".
7. Verify that the Reservation Status data structure returned by the NVMe Controller after completing RES_REP3 indicates that the host does not hold any reservations.
8. Verify that the completion queue entry for RES_ACQ4 indicates status Reservation Conflict.
9. Verify that the Reservation Status data structure returned by the NVMe Controller after completing RES_REP4 indicates that the host still holds its reservation with the namespace with the same reservation type. Verify that the Reservation Status data structure returned by the NVMe Controller after completing RES_REP5 indicates that the host still holds its reservation with the namespace with the same reservation type.

Case 3: Multiple Hosts (FYI) Dual Port Devices Only

Only one reservation is allowed at a time on a namespace. If a registrant attempts to obtain a reservation on a namespace that already has a reservation holder, then the command is aborted with status Reservation Conflict.

This test requires there to be at least two controllers with a shared namespace in the NVM Subsystem.

Test Procedure:

1. Check the ONCS field to determine of the controller supports reservations. If the controller does not support reservations then this test is not applicable. If the DUT is a single port device then this test is not applicable.
2. Configure NVMe Host 1 to issue a Set Features command with the Host Identifier feature to NVMe Controller 1 in order to set its Host Identifier for that controller.
3. Configure NVMe Host 2 to issue a Set Features command with the Host Identifier feature to NVMe Controller 2 with a different Host Identifier than NVMe Host 1.
4. Configure NVMe Host 1 to issue a Reservation Register command with the Reservation Register Action (RREGA) field set to 000b (i.e. Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field to NVMe Controller 1 for the shared namespace in order to make the host a registrant of that namespace.
5. Configure NVMe Host 2 to issue a Reservation Register command with the Register Reservation Key action and supplying a reservation key in the NRKEY field to NVMe Controller 2 for the shared namespace in order to make the host a registrant of that namespace.
6. Configure NVMe Host 1 to issue a Reservation Acquire command, setting the Reservation Acquire Action (RACQA) field to 000b (Acquire), supplying the reservation key currently associated with NVMe Host 1 in

©2021 UNH–IOL

the Current Reservation Key (CRKEY) field, and setting the Reservation Type (RTYPE) field to a reservation type supported by the namespace to NVMe Controller 1.

7. Configure NVMe Host 2 to issue a Reservation Acquire command with the Acquire action, supplying the reservation key currently associated with NVMe Host 2 in the CRKEY field, and setting the Reservation Type (RTYPE) field to a reservation type supported by the namespace to NVMe Controller 2.
8. Configure NVMe Host 2 to issue a Reservation Report command to NVMe Controller 2.
9. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated Completion Queue indicating the status for the command.
2. Verify that the completion queue entry for the Reservation Acquire command sent by NVMe Host 2 indicates status Reservation Conflict.
3. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the Reservation Report command indicates that NVMe Host 1 holds a reservation on the namespace and that NVMe Host 2 does not.

Possible Problems: None.

©2021 UNH–IOL

Test 4.4 – Releasing a Reservation (FYI)

Purpose: To determine if an NVMe Controller properly releases reservations.

References:

NVMe Specification 8.8 , 5.14.1.15, 6.12

Resource Requirements:

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

Last Modification: June 10, 2021

Discussion: A host releases a reservation by executing a Reservation Release command, setting the Reservation Release Action (RRELA) field to 000b (i.e. Release), setting the Reservation Type (RTYPE) field to the type of the reservation being released, and supplying the current reservation key associated with the host in the Current Reservation Key (CRKEY) field.

The Reservation Release command uses Command Dword 10 and a Reservation Release data structure in memory. If the command uses PRPs for the data transfer, then the PRP Entry 1 and PRP Entry 2 fields are used. If the command uses SGLs for the data transfer, then the SGL Entry 1 field is used. All other command specific fields are reserved.

Reservation release due to preemption or a registration clear is verified in subsequent tests.

Test Setup: See Appendix A.

Case 1: Release with Reservation Release Command (FYI)

Test Procedure:

1. Check the ONCS field to determine of the controller supports reservations. If the controller does not support reservations then this test is not applicable.
2. For each NVMe Controller in the NVM Subsystem:
    a. Configure the NVMe Host to issue a Set Features command with the Host Identifier feature to the NVMe Controller in order to set its Host Identifier for that controller.
    b. For each active namespace attached to the NVMe Controller:
        i. Configure the NVMe Host to issue a Reservation Register command with the Reservation Register Action (RREGA) field set to 000b (i.e. Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field to the NVMe Controller for the namespace in order to make the host a registrant of that namespace.
        ii. Configure the NVMe Host to issue a Reservation Acquire command, setting the Reservation Acquire Action (RACQA) field to 000b (Acquire), supplying the current reservation key associated with the host in the Current Reservation Key (CRKEY) field, and setting the Reservation Type (RTYPE) field to a reservation type supported by the namespace to the NVMe Controller in order for the host to acquire a reservation on the namespace with the reservation type.
        iii. Configure the NVMe Host to issue a Reservation Release command, setting the Reservation Release Action (RRELA) field to 000b (i.e. Release), setting the Reservation Type (RTYPE) field to the type of the reservation being released, and supplying the current reservation key associated with the host in the Current Reservation Key (CRKEY) field to the NVMe Controller in order to release the reservation held by the host.
        iv. Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
3. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

©2021 UNH–IOL

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated Completion Queue indicating the status for the command.
2. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the Reservation Report command indicates that the host successfully released the reservation.

Case 2: Reservation Release Command Error Conditions (FYI)

If the CRKEY value does not match that used by the registrant to register with the namespace, the command shall be aborted with status Registration Conflict. A host may release a reservation without regard to its current reservation key value by setting the Ignore Existing Key (IEKEY) bit to '1' in the command. If the RTYPE field does not match the type of the current reservation, then the command shall be completed with status Invalid Field in Command.

Test Procedure:

1. Check the ONCS field to determine of the controller supports reservations. If the controller does not support reservations then this test is not applicable.
2. For each NVMe Controller in the NVM Subsystem:
    a. Configure the NVMe Host to issue a Set Features command with the Host Identifier feature to the NVMe Controller in order to set its Host Identifier for that controller.
    b. For each active namespace attached to the NVMe Controller:
        i. Configure the NVMe Host to issue a Reservation Register command with the Reservation Register Action (RREGA) field set to 000b (i.e. Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field to the NVMe Controller for the namespace in order to make the host a registrant of that namespace.
        ii. Configure the NVMe Host to issue a Reservation Acquire command, setting the Reservation Acquire Action (RACQA) field to 000b (Acquire), supplying the current reservation key associated with the host in the Current Reservation Key (CRKEY) field, and setting the Reservation Type (RTYPE) field to a reservation type supported by the namespace to the NVMe Controller in order for the host to acquire a reservation on the namespace with the reservation type.
        iii. RES_REL1: Configure the NVMe Host to issue a Reservation Release command, setting the Reservation Release Action (RRELA) field to 000b (i.e. Release), setting the Reservation Type (RTYPE) field to the type of the reservation being released, and supplying any reservation key other than the current reservation key associated with the host in the Current Reservation Key (CRKEY) field to the NVMe Controller.
        iv. RES_REP1: Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
        v. RES_REL2: Configure the NVMe Host to issue a Reservation Release command with the Release action, setting the RTYPE field to any reservation type other than the type of the reservation being released, and supplying the current reservation key associated with the host in the CRKEY field to the NVMe Controller.
        vi. RES_REP2: Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
        vii. RES_REL3: Configure the NVMe Host to issue a Reservation Release command with the Release action, setting the Reservation Type (RTYPE) field to the type of the reservation being released, supplying the correct reservation key that is associated with the host in the CRKEY field, and setting the IEKEY bit to '1' to the NVMe Controller in order to release the reservation held by the host.
        viii. RES_REP3: Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
        ix. RES_REL4: Configure the NVMe Host to issue a Reservation Release command with the Release action, setting the Reservation Type (RTYPE) field to the type of the reservation being released, supplying the correct reservation key that is associated with the host in the CRKEY field, and setting the IEKEY bit to '0' to the NVMe Controller in order to release the reservation held by the host.

©2021 UNH–IOL

x. RES_REP4: Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.

xi.

3. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated Completion Queue indicating the status for the command.
2. Verify that the completion queue entry for RES_REL1 indicates status Reservation Conflict.
3. Verify that the Reservation Status data structure returned by the NVMe Controller after completing RES_REP1 indicates that the host still holds a reservation with the namespace.
4. Verify that the completion queue entry for RES_REL2 indicates status Invalid Field in Command.
5. Verify that the Reservation Status data structure returned by the NVMe Controller after completing RES_REP2 indicates that the host still holds a reservation with the namespace.
6. Verify that the Reservation Status data structure returned by the NVMe Controller after completing RES_REP3 indicates that the host did not release the reservation, and the Reservation Release command completed with status Invalid Field in Command.
7. Verify that the Reservation Status data structure returned by the NVMe Controller after completing RES_REP4 indicates that the host successfully released the reservation.

Case 3: Multiple Hosts (FYI) Dual Port Devices Only

An attempt by a registrant to release a reservation using the Reservation Release command in the absence of a reservation held on the namespace or when the host is not the reservation holder shall cause the command to complete successfully, but shall have no effect on the controller or namespace.

This test requires there to be at least two controllers with a shared namespace in the NVM Subsystem.

Test Procedure:

1. Check the ONCS field to determine of the controller supports reservations. If the controller does not support reservations then this test is not applicable. If the DUT is a single port device then this test is not applicable.
2. Configure NVMe Host 1 to issue a Set Features command with the Host Identifier feature to NVMe Controller 1 in order to set its Host Identifier for that controller.
3. Configure NVMe Host 2 to issue a Set Features command with the Host Identifier feature to NVMe Controller 2 with a different Host Identifier than NVMe Host 1.
4. Configure NVMe Host 1 to issue a Reservation Register command with the Reservation Register Action (RREGA) field set to 000b (i.e. Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field to NVMe Controller 1 for the shared namespace in order to make the host a registrant of that namespace.
5. Configure NVMe Host 2 to issue a Reservation Register command with the Register Reservation Key action and supplying a reservation key in the NRKEY field to NVMe Controller 2 for the shared namespace in order to make the host a registrant of that namespace.
6. Configure NVMe Host 1 to issue a Reservation Release command, setting the Reservation Release Action (RRELA) field to 000b (i.e. Release), setting the Reservation Type (RTYPE) field to any reservation type, and supplying the current reservation key associated with NVMe Host 1 in the Current Reservation Key (CRKEY) field to NVMe Controller 1.
7. Configure NVMe Host 1 to issue a Reservation Acquire command, setting the Reservation Acquire Action (RACQA) field to 000b (Acquire), supplying the reservation key currently associated with NVMe Host 1 in the Current Reservation Key (CRKEY) field, and setting the Reservation Type (RTYPE) field to a reservation type supported by the namespace to NVMe Controller 1.
8. Configure NVMe Host 2 to issue a Reservation Release command with the Release action, setting the RTYPE field to the type of the reservation which NVMe Host 1 holds, and supplying the current reservation key associated with NVMe Host 1 in the CRKEY field to NVMe Controller 2.
9. Configure NVMe Host 2 to issue a Reservation Report command to NVMe Controller 2.

©2021 UNH–IOL

10. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

Observable Results:
1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated Completion Queue indicating the status for the command.
2. Verify that the completion queue entry for the Reservation Release command sent by NVMe Host 1 indicates status Success.
3. Verify that the completion queue entry for the Reservation Release command sent by NVMe Host 2 indicates status Success.
4. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the Reservation Report command indicates that NVMe Host 1 holds a reservation on the namespace and that NVMe Host 2 does not.

Case 4: Release Due to Unregister (FYI)

If a host is the last remaining reservation holder (i.e. the Reservation Type is Write Exclusive - All Registrants or Exclusive Access - All Registrants) or is the only reservation holder, then the reservation is released when the host unregisters.

Test Procedure:
1. Check the ONCS field to determine of the controller supports reservations. If the controller does not support reservations then this test is not applicable.
2. For each NVMe Controller in the NVM Subsystem:
    a. Configure the NVMe Host to issue a Set Features command with the Host Identifier feature to the NVMe Controller in order to set its Host Identifier for that controller.
    b. For each active namespace attached to the NVMe Controller:
        i. Configure the NVMe Host to issue a Reservation Register command with the Reservation Register Action (RREGA) field set to 000b (i.e. Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field to the NVMe Controller for the namespace in order to make the host a registrant of that namespace.
        ii. Configure the NVMe Host to issue a Reservation Acquire command, setting the Reservation Acquire Action (RACQA) field to 000b (Acquire), supplying the current reservation key associated with the host in the Current Reservation Key (CRKEY) field, and setting the Reservation Type (RTYPE) field to 01h (Write Exclusive Access) to the NVMe Controller in order for the host to acquire a reservation on the namespace.
        iii. Configure the NVMe Host to issue a Reservation Register command with the Unregister Reservation Key action and supplying its current reservation key in the CRKEY field to the NVMe Controller for the namespace in order to unregister the host as a registrant of that namespace.
        iv. Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
3. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

Observable Results:
1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated Completion Queue indicating the status for the command.
2. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the Reservation Report command indicates that the host was successfully unregistered from the namespace and that the reservation held by the host on the namespace was released.

Possible Problems: None.

©2021 UNH–IOL

Appendix A: TEST SETUP

Refer to the NVMe Base Specification test plan for descriptions of valid test setups.

Appendix B: ZONE STATE TRANSITIONS REFERENCE

There are scenarios where in test procedures it is necessary to put a device into a particular state in the Zone State Machine before beginning the test procedure. Some test cases described in this document assume the Zone to be in a particular state when the test begins. The following outlines how a test procedure can move from one Zone State to any other Zone state before beginning a prescribed test procedure. This material references section 2.4 of the ZNS command set specification.

Starting Zone State	Intended Finishing Zone State	Procedure
ZSE	ZSIO	Perform a Write Operation to that zone.
	ZSEO	Perform a Zone Management Send command with action of Open Zone.
	ZSC	Perform a Zone Management Send command with action of Set Zone Descriptor Extension.
	ZSF	Perform a Zone Management Send command with action of Finish Zone.
ZSIO	ZSE	Perform a Zone Management Send command with action of Reset Zone.
	ZSEO	Perform a Zone Management Send command with action of Open Zone.
	ZSC	Perform a Zone Management Send command with action of Close Zone.
	ZSF	Perform a Zone Management Send command with action of Finish Zone.
ZSEO	ZSE	Perform a Zone Management Send command with action of Reset Zone.
	ZSC	Perform a Zone Management Send command with action of Close Zone.
	ZSF	Perform a Zone Management Send command with action of Finish Zone.
ZSC	ZSE	Perform a Zone Management Send command with action of Reset Zone.
	ZSIO	Perform a Write Operation to that zone.
	ZSEO	Perform a Zone Management Send command with action of Open Zone.
	ZSF	Perform a Zone Management Send command with action of Finish Zone.
ZSF	ZSE	Perform a Zone Management Send command with action of Reset Zone.

©2021 UNH–IOL

ZSRO	ZSO	Perform a Zone Management Send command with action of Offline Zone.

©2021 UNH–IOL