UNH–IOL
NVMe Testing Service

Test Plan for NVM Command Set Conformance
Version 26.0
Target Specification: NVM Express NVM Command Set Specification 1.3 and NVM Express Base Specification 2.4
Technical Document



University of New Hampshire
Interoperability Labs

Last Updated: August 13, 2026

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

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

Table of Contents

Table of Contents 1
Modification Records 19
Acknowledgements 54
Introduction 55
References 57
Abbreviations 58
Group 1: Admin Command Set 59
Test 1.1 – Identify Command (OF, M) 60
Case 1: CNS=00h Identify Namespace Data Structure (OF, M) 60
Case 2: CNS=01h Identify Controller Data Structure (OF, M) 61
Case 3: CNS=02h Namespace List (OF, M) 61
Case 4: CNS=03h Namespace Identification Descriptor (OF, M) 61
Case 5: CNS=10h Namespace ID List (OF-FYI, M) 62
Case 6: CNS=11h Identify Allocated Namespace Data Structure Allocated NSID (OF-FYI, M) 62
Case 7: CNS=11h Identify Allocated Namespace Data Structure Unallocated NSID (OF-FYI, M) 62
Case 8: CNS=11h Identify Allocated Namespace Data Structure Invalid NSID (OF-FYI, M) 63
Case 9: CNS=12h Namespace Attached Controller List (OF-FYI, M) 63
Case 10: CNS=13h Controller List (OF-FYI, M) 63
Case 11: CNS=14h Primary Controller Data Structure (OF-FYI, FYI) 64
Case 12: CNS=15h Secondary Controller List (OF-FYI, FYI) 64
Case 13: Identify to reserved CNS Value (OF, M) 64
Case 14: CNS=16h Namespace Granularity List (OF-FYI, M) 64
Case 15: CNS=17h UUID List (OF-FYI, M) 64
Case 16: CSI Field (OF-FYI, M) 65
Case 17: CNS=1Ch I/O Command Set Data Structure (OF-FYI, FYI) 65
Case 18: NVM Command Set Not Supported (OF-FYI, FYI) 66
Case 19: CNS=05h (OF-FYI, M) 66
Case 20: CNS=05h Namespace Management Not Supported (OF-FYI, M) 66
Case 21: CNS=06h (OF-FYI, M) 66
Case 22: CNS=1Bh (OF-FYI, M) 67
Case 23: CNS=1Bh, Invalid NSID (OF-FYI, M) 67
Case 24: CNS=1Ch (OF-FYI, M) 67
Case 25: CNS=08h Namespace Management Not Supported (OF-FYI, FYI) 67
Case 26: CNS=08h UIDREUSE (OF-FYI, FYI) 67
Case 27: CNS=08h Reservation Capabilities (OF-FYI, FYI) 68
Case 28: CNS=08h Format Progress Indicator (OF-FYI, M) 68
Case 29: CNS=08h ADA Group Identifier Cleared (OF-FYI, M) 68
Case 30: CNS=08h ADA Group Identifier Valid (OF-FYI, FYI) 69
Case 31: CNS=08h ADA Group Identifier Change Notice (OF-FYI, FYI) 69
Case 32: CNS=08h Namespace Attributes (OF-FYI, FYI) 69
Case 33: CNS=08h NVM Set Identifier (OF-FYI, M) 69
Case 34: CNS=08h Endurance Group Identifier (OF-FYI, M) 70
Case 35: CNS=04h NVM Set List, No Zero Entry (OF-FYI, M) 70
Case 36: CNS=19h Endurance Group, ENDGID > ENDGIDMAX (OF-FYI, FYI) 70
Case 37: CNS=00h Identify Namespace Data Structure, with Reported field (OF-FYI, M) 70
Case 38: CNS=05h Identify Namespace Data Structure, with Reported field (OF-FYI, M) 71
Case 39: CNS=09h Namespace Capabilities, NVM Command Set (FYI, FYI) 71
Case 40: CNS=0Ah, Namespace Capabilities, IO Command Set (FYI, FYI) 72
Case 41: CNS=08h Identify Namespace Data Structure, with Reported field (OF-FYI, M) 72
Case 42: CNS=18h Domain List (FYI, FYI) 72
Case 43: CNS=19h Endurance Group List (FYI, FYI) 73
Case 44: CNS=06h, CSI=00h VERSION (OF-FYI, M) 73
Case 45: CNS=11h, NSID=FFFFFFFFh (OF-FYI, M) 73
Case 46: CNS=08h Identify Namespace Data Structure, with Non-Storage Namespace-NVMSETID (OF-FYI, FYI) 73
Case 47: CNS=08h Identify Namespace Data Structure, with Non-Storage Namespace-ENDGID (OF-FYI, FYI) 74

© 2026 UNH–IOL

Case 48: CNS=1Fh Allocated Namespace Data Structure Allocated NSID (OF-FYI, FYI) 74
Case 49: CNS=1Fh Allocated Namespace Data Structure Unallocated NSID (OF-FYI, FYI) 74
Case 50: CNS=1Fh Invalid NSID (OF-FYI, FYI) 75
Case 51: CNS=1Fh NSID=FFFFFFFFh (OF-FYI, FYI) 75
Case 52: CNS=1Bh Identify Namespace Data Structure Allocated NSID Detached (OF-FYI, FYI) 75
Case 53: Media Verification Support: CNS=01h BES=0b and CES= 0b then VERS=0b (OF-FYI, FYI) 75
Case 54: Identify I/O Command Set Specific WZSL Not Supported (OF-FYI, FYI) 76
Case 55: Maximum Unlimited Power (MUP) Field is set to 0 when the Power Limit Support (PLS) Field is set to 0 (OF-FYI, FYI) 76
Case 56: Maximum Unlimited Power (MUP) and Maximum Unlimited Power Scale (MUPS) Field Indicate Maximum Power (OF-FYI, FYI) 76
Case 57: Power Limit Scale Reserved Values (OF-FYI, FYI) 76
Case 58: Invalid Power Limit when Power Limit Value Lower than all Maximum Power Values (OF-FYI, FYI) 77
Case 59: Power States Added when Power Level Value is higher than MP (OF-FYI, FYI) 77
Case 60: Power Limit Value does not disable or enable any Power States (OF-FYI, FYI) 78
Test 1.2 – Set/Get Features Commands (OF-FYI, M) 79
Case 1: SEL = 000b (OF, M) 81
Case 2: SEL = 001b (OF, M) 82
Case 3: SEL = 010b (OF, M) 84
Case 4: SEL = 011b (OF, M) 85
Case 5: SEL = Reserved Value (OF, M) 86
Case 6: SEL = 011b Attempt to Change value indicated as Not Changeable (OF, M) 87
Case 7: NSID of FFFFFFFFh to Namespace Specific Feature (OF-FYI, M) 88
Case 8: VWC Feature (OF, M) 89
Case 9: FID 03h, NSID=FFFFFFFFh (OF-FYI, M) 90
Case 10: Controller Feature Values, NSID=0h or FFFFFFFFh (OF-FYI, M) 90
Case 11: Multiple Set Features Commands for FID 03h (OF-FYI, M) 91
Case 12: Timestamp FID 0Eh (OF-FYI, M) 91
Case 13: Get Feature Namespace Specific FID Valid NSID (OF-FYI, M) 91
Case 14: Get Feature Namespace Specific FID NSID=FFFFFFFFh (OF-FYI, M) 92
Case 15: Set Feature Namespace Specific FID NSID=FFFFFFFFh (OF-FYI, M) 92
Case 16: I/O Command Set Profile Feature (OF-FYI, FYI) 93
Case 17: I/O Command Set Combination 0h (OF-FYI, FYI) 93
Case 18: I/O Command Set Combination Not Supported (OF-FYI, FYI) 93
Case 19: Feature Value Reset to Default after Controller Level Reset (OF-FYI, M) 94
Case 20: Embedded Management Controller Address Set Feature (OF-FYI, FYI) 95
Case 21: Embedded Management Controller Address Get Feature (OF-FYI, FYI) 95
Case 22: Host Management Agent Address Set Feature (OF-FYI, FYI) 95
Case 23: Host Management Agent Address Get Feature (OF-FYI, FYI) 95
Case 24: Get Feature Performance Characteristics, Standard Performance (1Ch) (OF-FYI, FYI) 96
Case 25: Get Feature Performance Characteristics, Performance Attribute Identifier List (C0h) (OF-FYI, FYI) 96
Case 26: Get Feature Performance Characteristics, Invalid Field (1Ch) (OF-FYI, FYI) 96
Case 27: Set Feature Performance Characteristics, Invalid Field (1Ch) (OF-FYI, FYI) 97
Case 28: Set Feature Performance Characteristics, Standard Performance Attribute (1Ch) (OF-FYI, FYI) . 97
Case 29: Set Feature Performance Characteristics, Vendor Specific, USVSPA=0 (1Ch) (OF-FYI, FYI) 97
Case 30: Set Feature Performance Attribute Identifier List (C0h) (OF-FYI, FYI) 97
Case 31: Get Feature Performance Attribute Identifier List (C0h) (OF-FYI, FYI) 97
Case 32: HOSTID Resets to 0h (OF-FYI, FYI) 98
Case 33: HOSTID set to Non-Zero value (OF-FYI, FYI) 98
Case 34: HOSTID Not Saveable (OF-FYI, FYI) 98
Case 35: Temperature Hysteresis Threshold, invalid TMPSEL (OF-FYI, FYI) 99
Case 36: Temperature Hysteresis Threshold, Greater than TMPTHMH (OF-FYI, FYI) 99
Case 37: Namespace Admin Label (1Fh), SEL=000b (OF-FYI, FYI) 99
Case 38: Namespace Admin Label (1Fh), SEL=001b (OF-FYI, FYI) 100
Case 39: Namespace Admin Label (1Fh), SEL=010b (OF-FYI, FYI) 100
Case 40: Namespace Admin Label (1Fh), SEL=011b (OF-FYI, FYI) 101
Case 41: Namespace Admin Label (1Fh), SEL=Reserved Value (OF-FYI, FYI) 101
Case 42: Namespace Admin Label (1Fh) (OF-FYI, FYI) 102
Case 43: Namespace Admin Label (1Fh) Saveable (OF-FYI, M) 102

© 2026 UNH–IOL

Case 44: Reservation Persistence (83h), not Saveable (OF-FYI, FYI) 103
Case 45: Get Feature LBA Range Type (03h) Data Structure (OF-FYI, M) 103
Case 46: Get Feature Host Metadata, GDHM (FYI) 103
Case 47: Feature Values after Controller Level Reset for Features that are not Saveable and are Persistent: (OF-FYI) 104
Case 48: NSID != 0 on Feature values that apply to NVM Subsystem Scope, Domain Scope, or Endurance Group Scope: (OF-FYI) 104
Case 49: Configurable Device Personality Get Features CDP with Unsupported Personality Identifier (FYI) 105
Case 50: Set Features with Invalid All Personalities Request (FYI) 105
Case 51: Host Metadata Descriptor Limit (FYI) 105
Test 1.3 – Get Log Page Command (OF, M) 106
Case 1: Supported LIDs (OF, M) 108
Case 2: Unsupported Vendor Specific LIDs (OF, M) 109
Case 3: Reserved LIDs (OF, M) 109
Case 4: NUMD/MDTS Conflict (OF, M) 110
Case 5: Get Error Information after Error (OF, M) 110
Case 6: SMART Temperature Threshold (OF, M) 110
Case 7: Data Units Read Count – Compare (OF, M) 111
Case 8: Data Units Written (FYI, M) 111
Case 9: Power Cycle Count (M) 112
Case 10: NUMD Greater than Log Page Conflict (FYI) 112
Case 11: Telemetry Host Initiated Valid Offset Create=1 (OF-FYI, M) 113
Case 12: Telemetry Host Initiated Valid Offset Create=0 (OF-FYI, M) 113
Case 13: Telemetry Host Initiated Invalid Offset (OF-FYI, M) 114
Case 14: Telemetry Controller Initiated Valid Offset (OF-FYI, M) 114
Case 15: Telemetry Controller Initiated Invalid Offset (OF-FYI, M) 114
Case 16: Telemetry Host Initiated Valid 0 Offset Create=1 (OF-FYI, M) 115
Case 17: Data Units Written Does not Increment for Write Uncorrectable (OF, M) 115
Case 18: Persistent Event Log (OF-FYI, M) 115
Case 19: Data Units Read Count – Verify (OF-FYI, M) 116
Case 20: Data Units Read Count – Read Only (OF, M) 116
Case 21: Data Units Written Does not Increment for Write Zeroes (OF, M) 117
Case 22: Invalid LPOL offset (OF-FYI, M) 117
Case 23: Invalid LPOU offset (OF-FYI, M) 118
Case 24: Domain Identifier (OF-FYI, M) 119
Case 25: Feature Identifiers Supports and Effects Log Page (OF-FYI, M) 120
Case 26: NVMe-MI Commands Supported and Effects Log Page (OF-FYI, M) 120
Case 27: Persistent Event Log Log Revision (OF-FYI, M) 120
Case 28: Persistent Event Log Generation Number (OF-FYI, M) 120
Case 29: Log Page Offset, Offset Type=0 (FYI, M) 121
Case 30: Log Page Offset, Offset Type=1 (FYI, FYI) 122
Case 31: Log Page Offset, Invalid Field in Command (FYI, M) 123
Case 32: LPOL, OT=0 (FYI, M) 124
Case 33: Invalid LPOU & LPOL, OT=0 (OF-FYI, M) 125
Case 34: Invalid LPOU & LPOL, OT=1 (OF-FYI, M) 126
Case 35: Invalid LPOU & LPOL, OT=1, IOS=0 (OF-FYI, M) 127
Case 36: Telemetry Host-initiated, Not Extended (OF-FYI, M) 128
Case 37: Telemetry Host-Initiated, bit 6=1, ETDAS=0 (OF-FYI, M) 129
Case 38: Telemetry Host-Initiated, bit 6=1, ETDAS=1 (OF-FYI, FYI) 129
Case 39: Telemetry Host-Initiated, Last Block field (OF-FYI, FYI) 129
Case 40: Telemetry Controller-initiated, LPA bit 6=0 (OF-FYI, M) 129
Case 41: Telemetry Controller-initiated, LPA bit 6=1 (OF-FYI, M) 130
Case 42: Telemetry Controller-initiated, LPA bit 6=1, ETDAS=0 (OF-FYI, M) 130
Case 43: Telemetry Host-initiated, LPA bit 6=1, ETDAS=0 (OF-FYI, M) 130
Case 44: Telemetry Controller-initiated, LPA bit 6=1 (OF-FYI, M) 130
Case 45: Telemetry Controller-Initiated, LPA bit 6=1, ETDAS=1 (OF-FYI, FYI) 131
Case 46: Media Unit Status Invalid Field in Command (OF-FYI, FYI) 131
Case 47: Supported Capacity Configuration List, Invalid Field in Command (OF-FYI, FYI) 131
Case 48: Predictable Latency Event Aggregate, Invalid Field in Command (OF-FYI, FYI) 131
Case 49: Media Unit Status, Selected Configuration=0 (OF-FYI, FYI) 132
Case 50: Media Unit Status, Channel Ids ordered (OF-FYI, FYI) 132

© 2026 UNH–IOL

Case 51: Media Unit Status, Domain Identifier=0 (OF-FYI, FYI) 132
Case 52: Media Unit Status, ENDGID <= ENDGIDMAX (OF-FYI, FYI) 132
Case 53: Media Unit Status, NVMSETID <= NSETIDMAX (OF-FYI, FYI) 133
Case 54: Media Unit Status, same Endurance Group, same Capacity Adjustment Factor (OF-FYI, FYI) .. 133
Case 55: Media Unit Status, Percentage Used (OF-FYI, FYI) 133
Case 56: Media Unit Status, Channel Identifiers Offset (OF-FYI, FYI) 134
Case 57: Support Capacity Configuration Lists, Capacity Configuration Identifier (OF-FYI, FYI) 134
Case 58: Support Capacity Configuration Lists, Endurance Group Configuration Identifier (OF-FYI, FYI) 134
Case 59: Support Capacity Configuration Lists, Multiple Domains (OF-FYI, FYI) 134
Case 60: Support Capacity Configuration Lists, NVM Set Identifier (OF-FYI, FYI) 135
Case 61: Support Capacity Configuration Lists, Channel Identifier (OF-FYI, FYI) 135
Case 62: Support Capacity Configuration Lists, Endurance Group ENDGID (OF-FYI, FYI) 135
Case 63: Support Capacity Configuration Lists, NVM Set Identifiers (OF-FYI, FYI) 135
Case 64: Support Capacity Configuration Lists, NVM Set EGSET (OF-FYI, FYI) 135
Case 65: Channel Configuration Descriptor, NVM Set EGSET (OF-FYI, FYI) 136
Case 66: Channel Configuration Descriptor, MUDL=0 (OF-FYI, FYI) 136
Case 67: NVM Capacity Model with Endurance Group support and no NVM Set support (OF-FYI, FYI) 136
Case 68: NVM Capacity Model, Endurance Group not supported (OF-FYI, FYI) 136
Case 69: Get Media Unit Status log page, ANA enabled (OF-FYI, FYI) 137
Case 70: Supported Capacity Configuration List log page, ANA enabled (OF-FYI, FYI) 137
Case 71: Telemetry Host-Initiated, Establish Context and Read 512 header support (OF-FYI, M) 137
Case 72: Telemetry Host-Initiated, MCDS and Data Area size (OF-FYI, FYI) 138
Case 73: Telemetry Host-Initiated, Generation Number (OF-FYI, FYI) 138
Case 74: Management Address List (OF-FYI, FYI) 138
Case 75: Get Error Information enhancement (OF-FYI, FYI) 139
Case 76: Format NVM Completion Event, Persistent Event Log (OF-FYI, FYI) 139
Case 77: Supported Capacity Configuration List (11h) appear ascending order and once (OF-FYI, FYI) . 139
Case 78: NSID Specified on Controller or NVM Subsystem Scope Log Pages (OF-FYI, M) 139
Case 79: NSID Controller or NVM Subsystem SMART/Health Log Page (OF-FYI, M) 141
Case 80: Command Supported Effects Log page – Cancel Command Not Supported (M) 141
Case 81: Command Supported Effects Log Page – Cancel Command Indicates Support (M) 141
Case 82: Error Information Log Page NVM Command Set Identifier (OF-FYI, FYI) 142
Case 83: Get Log Page Commands Ignore CSI field or if CC.CSS is Set to 110b (OF-FYI, M) 142
Case 84: Command Supported Effects Log page – Command Submission and Execution Relaxations (OF-FYI, M) 142
Case 85: Command Supported Effects Log page – Command Not Supported (OF-FYI, M) 143
Case 86: Get Log Page Command 0Dh, Persistent Event Log PIT Field != 0 (OF-FYI, M) 143
Case 87: Persistent Event Log Page (LID 0Dh) - Sanitize Media Verification Event (Event Type 0Eh) (OF-FYI, FYI) 143
Case 88: Persistent Event Log Page (LID 0Dh) - Sanitize Completion Event (Event Type 0Ah) (OF-FYI, FYI) 144
Case 89: Sanitize Status (Log Page Identifier 81h) Unallocated NSID (OF-FYI, FYI) 144
Case 90: Sanitize Status Log Page Target-Specific Field Validations (OF-FYI, FYI) 145
Case 91: Sanitize Namespace Status List Log Page Contents Verification (OF-FYI, FYI) 145
Test 1.4 – Create/Delete I/O Submission and Completion Queue Commands (M) 147
Case 1: Basic Operation (M) 147
Case 2: Create I/O Completion Queue with QID=0h, exceeds Number of Queues reported, Identifier Already in Use (M) 147
Case 3: Delete I/O Completion Queue before deleting Corresponding Submission Queue (M) 148
Case 4: Create I/O Completion Queue with Invalid Queue Size (M) 148
Case 5: Create I/O Submission Queue with Invalid Queue Size (M) 148
Case 6: Create I/O Submission Queue Physically Contiguous (M) 149
Case 7: Create I/O Submission Queue Invalid CQID of 0h (M) 149
Case 8: Create I/O Completion Queue Invalid Interrupt Vector (M) 149
Case 9: Create I/O Submission Queue Invalid CQID Outside Supported Range (M) 149
Case 10: Create I/O Submission Queue Invalid CQID within supported range, but Queue not created (M) 150
Case 11: Set Feature After Queues Created (M) 150
Test 1.5 – Abort Command (M) 151

© 2026 UNH–IOL

Case 1: Abort I/O Command (M) 151
Case 2: Abort Admin Command (M) 151
Case 3: Abort I/O Command - Immediate Abort Not Performed Bit (FYI) 152
Case 4: Abort Admin Command – Immediate Abort Not Performed Bit (FYI) 152
Test 1.6 – Format NVM Command (OF, M) 154
Case 1: Valid LBAF, SES=000b (OF, M) 154
Case 2: Valid LBAF, SES=001b (OF-FYI, M) 154
Case 3: Valid LBAF, SES=010b (OF-FYI, M) 155
Case 4: Valid LBAF, SES=111b (reserved value) (OF-FYI, M) 155
Case 5: Unsupported LBAF, SES=000b (OF-FYI, M) 156
Case 6: Unsupported LBAF, SES=111b (reserved value) (OF-FYI, M) 157
Case 7: Valid LBAF, SES=000b, PI is non-zero (M) 157
Case 8: Valid LBAF, SES=000b, PI=Type 3 (OF-FYI, FYI) 158
Case 9: NSID=FFFFFFFFh, no attached namespaces, SES=000 (OF-FYI, FYI) 159
Case 10: NSID=FFFFFFFFh, no attached namespaces, SES!=000 (OF-FYI, FYI) 159
Case 11: NSID=FFFFFFFFh Supported (OF-FYI, M) 160
Case 12: NSID=FFFFFFFFh Not Supported (OF-FYI, FYI) 160
Case 13: Valid LBAFEE Value (OF-FYI, FYI) 160
Case 14: LBA Format & Namespace Granularity (LBAFEE=1h, & ELBAS=1h) (OF-FYI, FYI) 161
Case 15: LBA Formats & Namespace Granularity (LBAFEE=0h) (OF-FYI, FYI) 161
Case 16: Format (LBAFEE=0h & ELBAS=1h) (OF-FYI, FYI) 161
Case 17: Format on Namespace that is Write Protected (OF-FYI, FYI) 162
Case 18: Format with an I/O Command in Progress (OF-FYI, M) 162
Case 19: I/O Command, with a Format Command in Progress (OF-FYI, M) 162
Case 20: Formatting Namespaces to non-zero STS when in a different format & LBAFEE = 1 (OF-FYI) .163
Test 1.7 – Asynchronous Events (OF, M) 164
Case 1: Asynchronous Event Request Command (M) 164
Case 2: Outstanding Commands Aborted after Reset (OF, M) 165
Case 3: Clearing Events (M) 165
Case 4: Masking Events (OF, M) 166
Case 5: Asynchronous Events not Cleared (OF-FYI, M) 167
Case 6: Asynchronous Event Sanitize Operation Entered Media Verification State (OF-FYI, FYI) 167
Case 7: Sanitize Operation Completed (OF-FYI, M) 167
Case 8: Sanitize Operation Unexpected Deallocation (OF-FYI, M) 168
Case 9: Sanitize Clears Events (OF-FYI, FYI) 169
Case 10: Attached and Allocated Namespace Attribute Notice Asynchronous Event (OF-FYI, FYI) 169
Case 11: Controller Data Queue Tail Pointer Event Triggered (FYI) 170
Test 1.8 – Get Feature Select (M) 171
Case 1: Get Feature Select (M) 171
Test 1.9 – Feature Saved Across Reset (M) 173
Case 1: Feature Saved Across Reset (M) 173
Test 1.10 – Device Self-test Short Operation (OF-FYI, M) 175
Case 1: Namespace Test Action = 00000000h, STC=1h (OF-FYI, M) 175
Case 2: Namespace Test Action = 00000001h-FFFFFFFEh, STC=1h (OF-FYI, M) 175
Case 3: Namespace Test Action = FFFFFFFFh, STC=1h (OF-FYI, M) 176
Case 4: Namespace Test Action = Invalid Namespace, STC=1h (OF-FYI, M) 177
Case 5: Namespace Test Action = Inactive Namespace, STC=1h (OF-FYI, M) 177
Case 6: Namespace Test Action = 00000000h, STC=1h and 1h (OF-FYI, M) 177
Case 7: Namespace Test Action = 00000001h-FFFFFFFEh, STC=1h and 1h (OF-FYI, M) 178
Case 8: Namespace Test Action = FFFFFFFFh, STC=1h, Second DST (OF-FYI, M) 179
Test 1.11 – Device Self-test Extended Operation (OF-FYI, M) 180
Case 1: Namespace Test Action = 00000000h, STC=2h (OF-FYI, M) 180
Case 2: Namespace Test Action = 00000001h-FFFFFFFEh, STC=2h (OF-FYI, M) 180
Case 3: Namespace Test Action = FFFFFFFFh, STC=2h (OF-FYI, M) 181
Case 4: Namespace Test Action = Invalid Namespace, STC=2h (OF-FYI, M) 182
Case 5: Namespace Test Action = Inactive Namespace, STC=2h (OF-FYI, M) 182
Case 6: Namespace Test Action = 00000000h, STC=2h and 2h (OF-FYI, M) 182
Case 7: Namespace Test Action = 00000001h-FFFFFFFEh, STC=2h and 2h (OF-FYI, M) 183
Case 8: Namespace Test Action = FFFFFFFFh, STC=2h, Second DST (OF-FYI, M) 184
Test 1.12 – Abort Device Self-test Short Operation (OF-FYI, M) 185
Case 1: Namespace Test Action = 00000000h, STC=1h (OF-FYI, M) 185

© 2026 UNH–IOL

Case 2: Test Action = 00000001h-FFFFFFFEh, STC=1h (OF-FYI, M) 185
Case 3: Test Action = FFFFFFFFh, STC=1h (OF-FYI, M) 186
Case 4: DST with specific NSID Aborted by Format NVM Command with specific NSID (OF-FYI, M) 187
Case 5: Aborted by Controller Level Reset (OF-FYI, M) 187
Case 6: Aborted by Sanitize Operation (OF-FYI, M) 188
Case 7: DST with specific NSID Aborted by Format NVM Command with NSID = FFFFFFFFh (OF-FYI, M) 188
Case 8: DST NSID=FFFFFFFFh Aborted by Format NVM Command with NSID = FFFFFFFFh (OF-FYI, M) 189
Test 1.13 – Abort Device Self-test Extended Operation (OF-FYI, M) 190
Case 1: Namespace Test Action = 00000000h, STC=2h (OF-FYI, M) 190
Case 2: Test Action = 00000001h-FFFFFFFEh, STC=2h (OF-FYI, M) 190
Case 3: Test Action = FFFFFFFFh, STC=2h (OF-FYI, M) 191
Case 4: DST with specific NSID Aborted by Format NVM Command with specific NSID (OF-FYI, M) 192
Case 5: Aborted by Sanitize Operation (OF-FYI, M) 192
Case 6: DST with specific NSID Aborted by Format NVM Command with NSID = FFFFFFFFh (OF-FYI, M) 193
Case 7: DST NSID=FFFFFFFFh Aborted by Format NVM Command with NSID = FFFFFFFFh (OF-FYI, M) 193
Test 1.14 – NVMe-MI Send/Receive (OF-FYI, M) 195
Case 1: Request and Response (OF-FYI, M) 195
Test 1.15 – Directive Receive Identify (OF-FYI, FYI) 196
Case 1: Valid Receive (OF-FYI, FYI) 196
Case 2: Receive with NSID=FFFFFFFFh (OF-FYI, FYI) 196
Test 1.16 – Directive Send Enable Directive (OF-FYI, FYI) 197
Case 1: Valid Send (OF-FYI, FYI) 197
Case 2: Send to Enable Identify (OF-FYI, FYI) 197
Case 3: Send to Enable Unsupported Directive (OF-FYI, FYI) 198
Case 4: Shared Stream Writes (OF-FYI, FYI) 198
Case 5: Directive Send Release Resources updates NSSA (OF-FYI, FYI) 198
Case 6: Namespace Deletion updates NSSA (OF-FYI, FYI) 199
Case 7: Format NVM to Namespace updates NSSA (OF-FYI, FYI) 199
Case 8: Send Vendor Specific Directive, not supported (OF-FYI, FYI) 199
Test 1.17 – Sanitize Command (OF-FYI, FYI) 201
Case 1: Sanitize Supported (OF-FYI, M) 201
Case 2: Sanitize Not Supported (OF-FYI, M) 201
Case 3: Sanitize Config FID Not Savable (OF-FYI, FYI) 201
Case 4: Sanitize Config NDI=1, NODRM=1 (OF-FYI, M) 202
Case 5: Sanitize Config NDI=1, NODRM=0 (OF-FYI, M) 202
Case 6: Sanitize In Progress SPROG (OF-FYI, M) 202
Case 7: Sanitize Not In Progress SPROG (OF-FYI, M) 202
Case 8: Sanitize Completes Succesfully, then Sanitize with Action = 001b (OF-FYI, M) 203
Case 9: Prohibited Commands Aborted when Sanitize In Progress SPROG (OF-FYI, M) 203
Case 10: Sanitize on Write Protected Namespace (OF-FYI, FYI) 203
Case 11: Media Verification - Media Verification Supported (OF-FYI, FYI) 204
Case 12: Media Verification - SPROG=FFFFh, SSI=5b (OF-FYI, FYI) 204
Case 13: Media Verification - NVM Subsystem Reset Exits Media Verification State (OF-FYI, FYI) 205
Case 14: Global Data Erase (OF-FYI, M) 205
Case 15: Sanitize Operation Status - Sanitized (SOS=001b) (OF-FYI, M) 205
Case 16: Sanitize Operation Status - Sanitizing (SOS=010b) (OF-FYI, M) 206
Case 17: Sanitize Operation Status - Sanitized Unexpected Deallocate (OF-FYI, M) 206
Case 18: Sanitize Command Dword 10 Information (SCDW10) (OF-FYI, M) 206
Case 19: Sanitize Operation Status (SOS!=011b) - Failure State=0b (OF-FYI, FYI) 207
Case 20: Sanitize Command - VERS=0b and EMVS=1b (OF-FYI, FYI) 207
Case 21: Media Verification - EMVS=1b and SANACT=011b (OF-FYI, FYI) 207
Case 22: Media Verification - EMVS=1b and NDAS=1b (OF-FYI, FYI) 207
Case 23: Media Verification - Not in Media Verification State, SANACT=101b (OF-FYI, FYI) 208
Case 24: Sanitize Persists Across Controller Level Reset (OF-FYI, M) 208
Case 25: Media Verification - Sanitizing (OF-FYI, FYI) 208

© 2026 UNH–IOL

Case 26: SANACT field not set to Exit Media Verification (101b) (OF-FYI, FYI) 209
Case 27: Media Verification - Deallocates All Media Allocated for User Data (OF-FYI, FYI) 209
Case 28: Media Verification - Read with PRCHK > 0h (OF-FYI, FYI) 209
Case 29: Sanitize State Machine (OF-FYI, FYI) 210
Case 30: Sanitize Command Aborts While Sanitization Target is a Controller in a Suspended State (FYI) .211
Case 31: Sanitize Execution Behavior with EMVS Bit Cleared (EMVS = 0b) (OF-FYI, FYI) 211
Case 32: EMVS Bit Ignored When SANACT Does Not Start a Sanitize Namespace Operation (OF-FYI, FYI) 212
Case 33: Sanitize command Option and Rejection Constraints (OF-FYI, FYI) 212
Case 34: Sanitize Namespace Interruption and Media Verification Canceled Tracking (OF-FYI, FYI) 212
Case 35: Sanitize Command Abort and Restriction in Media Verification State (OF-FYI, FYI) 213
Test 1.18 – Virtualization Management Command (OF-FYI, FYI) 214
Case 1: Valid Virtualization Management Command, RT=000b (OF-FYI, FYI) 214
Case 2: Valid Virtualization Management Command, RT=001b (OF-FYI, FYI) 214
Case 3: Invalid CNTLID (OF-FYI, FYI) 215
Test 1.19 – Security Receive (OF-FYI, FYI) 216
Case 1: SECP = 00h (OF-FYI, M) 216
Case 2: SECP = Unsupported Value (OF-FYI, M) 216
Test 1.20 – Security Send (OF-FYI, FYI) 217
Case 1: SECP = Unsupported Value (OF-FYI, M) 217
Test 1.21 – Command and Feature Lockdown (OF-FYI, FYI) 218
Case 1: Lockdown a not prohibited Opcode or Feature Identifier (OF-FYI, M) 218
Case 2: Lockdown OACS check for all controllers (OF-FYI, M) 218
Case 3: Lockdown with incorrect Interface Field (OF-FYI, FYI) 218
Case 4: Lockdown with incorrect Interface Field and Scope (OF-FYI, M) 219
Case 5: Lockdown Get Log Page, no Management Endpoint (OF-FYI, FYI) 219
Case 6: Lockdown Get Log Page, Length field cleared (OF-FYI, FYI) 219
Case 7: Lockdown Command and Feature Identifier List (OF-FYI, M) 219
Case 8: Lockdown Change Aborted (OF-FYI, M) 220
Case 9: Lockdown persists until power cycle (OF-FYI, M) 220
Case 10: Lockdown persists until next Lockdown Command (OF-FYI, M) 220
Test 1.22 – Capacity Management (OF-FYI, FYI) 222
Case 1: Capacity Management command, Invalid Field (OF-FYI, FYI) 222
Case 2: Media Unit Configuration Selection create Endurance and NVM Set (OF-FYI, FYI) 222
Case 3: Media Unit Configuration Selection delete Endurance and NVM Set (OF-FYI, FYI) 223
Case 4: Media Unit Configuration Selection, No Change (OF-FYI, FYI) 223
Case 5: Media Unit Configuration Selection, Invalid Field in Command (OF-FYI, FYI) 223
Case 6: Create Endurance Group (OF-FYI, FYI) 224
Case 7: Create Endurance Group, Insufficient Capacity (OF-FYI, FYI) 224
Case 8: Delete Endurance Group, Invalid Field in Command (OF-FYI, FYI) 224
Case 9: Create NVM Set (OF-FYI, FYI) 224
Case 10: Create NVM Set, Insufficient Capacity (OF-FYI, FYI) 225
Case 11: Create NVM Set, Element Identifier=0 (OF-FYI, FYI) 225
Case 12: Delete NVM Set, Invalid Field in Command (OF-FYI, FYI) 225
Case 13: Fixed and/or Variable Capacity Management supported (OF-FYI, FYI) 225
Case 14: Fixed Capacity Management Support (OF-FYI, FYI) 226
Case 15: Variable Capacity Management Support (OF-FYI, FYI) 226
Test 1.23 – Cancel Command (FYI) 227
Case 1: Cancel I/O Command – Single Command Cancel (FYI) 227
Case 2: Cancel I/O Command – Invalid Command ID (FYI) 227
Case 3: Cancel I/O Command – Incorrect SQID (FYI) 228
Case 4: Cancel I/O Command – Single Command Cancel, Different Namespace (FYI) 228
Case 5: Cancel I/O Command – Single Command Cancel, Command not Aborted (FYI) 228
Case 6: Cancel I/O Command – Multiple Command Cancel (FYI) 229
Case 7: Cancel I/O Command – Multiple Command Cancel, CID not equal FFFFh (FYI) 229
Case 8: Cancel I/O Command – Multiple Command Cancel, Different Namespace (FYI) 230
Test 1.24 – Keep Alive Timer (OF-FYI, FYI) 231
Case 1: KATO - KAS (OF-FYI, FYI) 231
Case 2: KATO – Keep Alive Timer Clear to 0h (OF-FYI, FYI) 231
Case 3: Keep Alive Timer Maximum Keep Alive Timeout (OF-FYI, FYI) 231
Case 4: KATO Command Cleanup Within Quiesce Time (OF-FYI, FYI) 232

© 2026 UNH–IOL

Case 5: Command Based Keep Alive Timer Restart (OF-FYI, FYI) 232
Test 1.25 – Rotational Media (OF-FYI, FYI) 234
Case 1: Rotational Media Support Requirements (OF-FYI, FYI) 234
Case 2: Rotational Media Log Page 16h and Endurance Group Log Page Requirements (OF-FYI, FYI) . 234
Case 3: Rotational Media Log Page 16h During a Sanitize Operation (OF-FYI, FYI) 235
Case 4: FID 1Ah, Spinup Control on Rotational Media Persistence (OF-FYI, FYI) 235
Case 5: Rotational Media Log Page Initial Spinup After Subsystem Reset (OF-FYI, FYI) 236
Case 6: Rotational Media FID 1Ah (OF-FYI, M) 236
Test 1.26 – Device Self-test Host-Initiated Refresh Operation (OF-FYI, FYI) 237
Case 1: Device Self-test Command with STC=3h (OF-FYI, FYI) 237
Case 2: Device Self-Test Options SDSO (OF-FYI, M) 237
Case 3: Host-Initiated Operation Aborted (OF-FYI, FYI) 238
Group 2: NVM Command Set 240
Test 2.1 – Compare Command (OF-FYI, M) 241
Case 1: Valid SLBA (OF-FYI, M) 241
Case 2: SLBA Out of Range (OF-FYI, M) 241
Case 3: SLBA In Range, NLB Goes out of range (OF-FYI, M) 242
Case 4: SLBA Out of Range, NLB > MDTS (OF-FYI, M) 242
Case 5: SLBA Out of Range, but Lower Dword = 00000000 (OF-FYI, M) 242
Case 6: Invalid Namespace ID (OF-FYI, M) 242
Case 7: PRCHK Non-zero (OF-FYI, FYI) 243
Case 8: NSID=FFFFFFFFh (OF-FYI, M) 243
Test 2.2 – Dataset Management Command (OF-FYI, M) 244
Case 1: Basic Operation (OF, M) 244
Case 2: Deallocate (OF, M) 245
Case 3: NR Value is Maximum (OF-FYI, M) 245
Case 4: Correct Range Deallocated (OF, M) 246
Case 5: Deallocate Multiple Ranges (OF, M) 246
Case 6: NSID=FFFFFFFFh (OF-FYI, M) 247
Case 7: ONCS Bit 2 =1 Non-MDTS Command Size Limits DMRL (OF-FYI, FYI) 247
Case 8: ONCS Bit 2 =1 DSM Supported Non-MDTS Command Size Limits DMRSL (OF-FYI, M) 247
Case 9: ONCS Bit 2 =1 DSM Supported Non-MDTS Command Size Limits DMSL (OF-FYI, M) 247
Case 10: ONCS Bit 2 =0 Non-MDTS Command Size Limits DMRL (OF-FYI, M) 248
Case 11: ONCS Bit 2 =0 DSM Supported Non-MDTS Command Size Limits DMRSL (OF-FYI, M) 248
Case 12: ONCS Bit 2 =0 DSM Supported Non-MDTS Command Size Limits DMSL (OF-FYI, M) 248
Test 2.3 – Read Command (OF-FYI, M) 249
Case 1: Valid Read, LR=0, FUA=0 (OF, M) 249
Case 2: SLBA Out of Range (OF, M) 249
Case 3: SLBA In Range, NLB Goes out of range (OF, M) 249
Case 4: SLBA Out of Range, NLB > MDTS (OF, M) 250
Case 5: SLBA Out of Range, but Lower Dword = 00000000 (OF, M) 250
Case 6: Invalid Namespace ID (OF, M) 250
Case 7: Invalid Namespace ID and SLBA Out of Range (OF, M) 250
Case 8: Valid Read, LR=0, FUA=1 (OF, M) 251
Case 9: Valid Read, LR=1, FUA=0 (OF, M) 251
Case 10: Valid Read, LR=1, FUA=1 (OF, M) 251
Case 11: NSID=FFFFFFFFh, LR=0, FUA=0 (OF-FYI, M) 252
Case 12: Type 1 Protection, Invalid ILBRT (OF-FYI, FYI) 252
Case 13: Type 1 Protection, Invalid EILBRT (OF-FYI, FYI) 252
Test 2.4 – Write Command (OF-FYI, M) 253
Case 1: Valid Write, LR=0, FUA=0 (OF, M) 253
Case 2: SLBA Out of Range (OF, M) 253
Case 3: SLBA In Range, NLB Goes out of range (OF, M) 253
Case 4: SLBA Out of Range, NLB > MDTS (OF, M) 254
Case 5: SLBA Out of Range, but Lower Dword = 00000000 (OF, M) 254
Case 6: Invalid Namespace ID (OF, M) 254
Case 7: Invalid Namespace ID and SLBA Out of Range (OF, M) 254
Case 8: Valid Write, LR=0, FUA=1 (OF, M) 255
Case 9: Valid Write, LR=1, FUA=0 (M) 255
Case 10: Valid Write, LR=1, FUA=1 (OF, M) 255
Case 11: NSID=FFFFFFFFh, LR=0, FUA=0 (OF, M) 256

© 2026 UNH–IOL

Test 2.5 – Write Uncorrectable Command (OF, M) 257
Case 1: SLBA In Range, NLB Valid (OF, M) 257
Case 2: SLBA Out of Range, NLB Valid (OF, M) 257
Case 3: SLBA Out of Range, NSID Invalid (OF, M) 258
Case 4: SLBA Out of Range, but Lower Dword = 00000000 (OF, M) 258
Case 5: NLB greater than MDTS and Non-MDTS Command Size Limits Not Supported (OF, M) 258
Case 6: NSID=FFFFFFFFh, SLBA In Range, NLB Valid (OF-FYI, M) 259
Case 7: NLB greater than WUSL and Non-MDTS Command Size Limits Supported (OF-FYI, FYI) 259
Test 2.6 – Flush Command (OF, M) 260
Case 1: Valid Namespace ID (OF, M) 260
Case 2: Invalid Namespace ID (OF, M) 260
Case 3: NSID=0xFFFFFFFF (OF, M) 260
Case 4: Flush when VWC Not Present (OF-FYI, M) 261
Case 5: FDP Identify Controller, VWCP=1 & VCE=0, Persistent Reset (OF-FYI, M) 261
Case 6: Flush With Volatile Write Cache and Power Cycle (M) 261
Test 2.7 – Write Zeroes Command (OF, M) 263
Case 1: SLBA In Range, NLB Valid, LR=0, FUA=0 (OF, M) 263
Case 2: SLBA Out of Range, NLB Valid (OF, M) 263
Case 3: SLBA Out of Range, NSID Invalid (OF, M) 264
Case 4: SLBA Out of Range, but Lower Dword = 00000000 (OF, M) 264
Case 5: NLB greater than MDTS and Non-MDTS Command Size Limits Not Supported (OF, M) 264
Case 6: SLBA In Range, NLB Valid, LR=0, FUA=1 (OF, M) 265
Case 7: SLBA In Range, NLB Valid, LR=1, FUA=0 (OF, M) 265
Case 8: SLBA In Range, NLB Valid, LR=1, FUA=1 (OF, M) 265
Case 9: PRCHK is Non Zero (OF-FYI, M) 266
Case 10: NSID=FFFFFFFFh, SLBA In Range, NLB Valid, LR=0, FUA=0 (OF-FYI, M) 266
Case 11: NLB greater than WZSL and Non-MDTS Command Size Limits Supported (OF-FYI, M) 267
Case 12: Namespace Zeroes Supported Deallocates Non-Specified LBAs (OF-FYI, FYI) 267
Case 13: WZDSL Greater than WZSL (OF-FYI, FYI) 267
Case 14: Namespace Zeroes Requires Deallocate (OF-FYI, FYI) 268
Case 15: Write Zeroes with Namespace Zeroes Ignores SLBA (OF-FYI, FYI) 268
Case 16: Write Zeroes with Namespace Zeroes Ignores SLBA (OF-FYI, FYI) 268
Case 17: Write Zeroes Command with Deallocate Data Size > Write Zeroes Command Data Size with no Deallocate (OF-FYI, FYI) 269
Test 2.8 – Atomicity Parameters (OF, M) 270
Case 1: NVM Command Set Supported (OF, M) 270
Case 2: NVM Command Set Not Supported (OF-FYI, FYI) 270
Test 2.9 – AWUN/NAWUN (M) 272
Case 1: Atomic Boundaries Not Supported (NABSN/NABSPF = 0) (M) 272
Case 2: Atomic Boundaries Supported (NABSN != 0) (M) 272
Case 3: Multiple Atomicity, Atomic Boundaries Supported (NABSN != 0) (FYI) 273
Test 2.10 – AWUPF/NAWUPF (IP) 274
Test 2.11 – Verify Command (OF-FYI, M) 275
Case 1: Valid Command (OF-FYI, FYI) 275
Case 2: PRACT = 1 (OF-FYI, FYI) 275
Case 3: NSID=FFFFFFFFh (OF-FYI, M) 276
Case 4: Command Size Limits (OF-FYI, FYI) 276
Test 2.12 – Fused Operations (OF-FYI, FYI) 277
Case 1: Supported Fused Operation (OF-FYI, M) 277
Case 2: Fused Operations Not Supported (OF-FYI, M) 277
Case 3: Missing Fused Command (OF-FYI, FYI) 277
Case 4: LBA Range Mismatch (OF-FYI, M) 278
Case 5: Unsupported Fused Operation (OF-FYI, M) 278
Case 6: Command Aborted due to Failed Fused Command (OF-FYI, M) 278
Test 2.13 – Deallocate and Allocate (OF-FYI, FYI) 280
Case 1: Deallocate via Dataset Management then READ, DULBE=0 (OF-FYI, M) 280
Case 2: Deallocate via Dataset Management, DULBE=1 (OF-FYI, FYI) 280
Case 3: Allocate via Write (OF-FYI, M) 281
Case 4: Allocate via Write Uncorrectable (OF-FYI, FYI) 281
Case 5: Allocate via Write Zeroes (OF-FYI, M) 281
Case 6: Deallocate via Dataset Management then COPY, DULBE=0 (OF-FYI, FYI) 282

© 2026 UNH–IOL

Test 2.14 – Copy (OF-FYI, FYI) 283
Case 1: Copy Command Supported (OF-FYI, FYI) 283
Case 2: MSSRL Exceeded (OF-FYI, FYI) 283
Case 3: MCL Exceeded (OF-FYI, FYI) 283
Case 4: MSRC Exceeded (OF-FYI, FYI) 284
Case 5: Descriptor Format Not Supported (OF-FYI, FYI) 284
Case 6: PRACT Mismatch (OF-FYI, FYI) 284
Case 7: Copy to Read Only Media (OF-FYI, FYI) 284
Case 8: Copy Descriptor Formats 2h and 3h Disabled (OF-FYI, FYI) 285
Case 9: Copy Descriptor Formats 1h and 3h with Destination NSID Formatted to Use 16b Guard Protection (OF-FYI, FYI) 285
Case 10: Copy Descriptor Formats 0h and 2h with Destination NSID Formatted to Use 32b Guard Protection (OF-FYI, FYI) 285
Case 11: Copy Descriptor Formats 0h and 2h with Destination NSID Formatted to Use 64b Guard Protection (OF-FYI, FYI) 286
Case 12: Copy Command with Unusable Source Namespace Identifiers (OF-FYI, FYI) 286
Case 13: Copy Command Copy Descriptor Format 2h and 3h with Overlapping I/O Range (OF-FYI, FYI) 286
Case 14: NVM Copy Single Atomicity Support Requirements (OF-FYI, FYI) 287
Case 15: NVM Copy Single Atomicity Verification (OF-FYI, FYI) 287
Case 16: Cross Namespace Copy without Protection Information (OF-FYI, FYI) 287
Case 17: Cross Namespace Copy with PRACT Mismatched (OF-FYI, FYI) 288
Case 18: Cross Namespace Copy with LBADs Mismatched (OF-FYI, FYI) 288
Case 19: Cross Namespace Copy with DPS Mismatched (OF-FYI, FYI) 289
Case 20: Cross Namespace Copy with PIFA Mismatched (OF-FYI, FYI) 289
Case 21: Cross Namespace Copy with Logical Block Storage Tag Mask Mismatched (OF-FYI, FYI) 290
Case 22: Cross Namespace Copy, One Namespace without Protection Information to a Namespace with Protection Information (OF-FYI, FYI) 291
Case 23: Cross Namespace Copy, One Namespace with Protection Information to a Namespace Without Protection Information (OF-FYI, FYI) 291
Case 24: Cross Namespace Copy, Metadata Size on Destination Namespace is Larger than the PI Size of the Namespace (OF-FYI, FYI) 292
Case 25: Cross Namespace Copy, No Reachability Association (OF-FYI, FYI) 292
Group 3: NVM Features 294
Test 3.1 – Metadata Handling (OF-FYI, M) 295
Case 1: Extended LBA (OF-FYI, M) 295
Case 2: Separate Buffer (OF-FYI, M) 295
Test 3.2 – End–to–end Data Protection (M) 297
Case 1: Write Command Processing (M) 297
Case 2: Read Command Processing (M) 298
Case 3: DPC=0 PRACT ignored by Controller (M) 298
Case 4: DPC=0 PRCHK ignored by Controller (M) 299
Case 5: DPC= 0 & STS=0 Storage Tag ignored by Controller (M) 299
Case 6: Metadata size=0 MSET ignored by Controller (M) 299
Case 7: Qualified Protection Information Format Support (FYI) 300
Case 8: Qualified protection Information Format Supported, STMLA=001b (FYI) 300
Case 9: 16b Guard Protection Information Storage Tag Mask Requirement (FYI) 301
Case 10: Storage Tag Masking Level Attributes LBSTM Formatting (FYI) 301
Case 11: STMLA Value Check, QPIFS=0 (M) 301
Case 12: STMLA Value Check, PIF not equal to 11b (M) 302
Case 13: Qualified Protection Information Not Supported (M) 302
Test 3.3 – Power Management (OF, M) 304
Case 1: Relative Write Latency (OF, M) 304
Case 2: Relative Write Throughput (OF, M) 305
Case 3: Relative Read Latency (OF, M) 305
Case 4: Relative Read Throughput (OF, M) 305
Case 5: Power Management Feature (OF, M) 306
Test 3.4 – Host Memory Buffer (M) 307
Case 1: Proper Structure (M) 307
Case 2: Configuration (M) 307

© 2026 UNH–IOL

Case 3: Reset Persistent (M) 308
Case 4: Enable HMB when Already Enabled (M) 308
Case 5: Disable HMB when Already Disabled (M) 309
Case 6: HMB Restrict Non-Operational Power State Access Enable (FYI) 309
Case 7: HMB Restrict Non-Operational Power State Access Not Supported (FYI) 310
Case 8: HMB Restrict Non-Operational Power State CQE Verification (FYI) 310
Test 3.5 – Replay Protected Memory Block (IP) 311
Case 1: RPMB Operations (IP) 311
Case 2: Authentication Key Programming (IP) 312
Case 3: Read Write Counter Value (IP) 312
Case 4: Authenticated Data Write (IP) 312
Case 5: Authenticated Data Read (IP) 313
Test 3.6 – IO Determinism (OF-FYI, FYI) 315
Case 1: Predictable Latency Mode Supported (OF-FYI, FYI) 315
Case 2: Predictable Latency Mode Not Supported (OF, M) 315
Case 3: Predictable Latency Mode Not Enabled (OF-FYI, FYI) 316
Test 3.7 – Namespace Write Protection (OF-FYI, FYI) 317
Case 1: Enable and Disable Write Protection (OF, FYI) 317
Test 3.8 – Persistent Memory Region (IP, IP) 319
Case 1: PMR Persistence Across Reset (IP, IP) 319
Case 2: PMR Persistence Across Disable and Enable (IP, IP) 319
Case 3: PMRMSC L and U CBA Persistence Across Reset (IP, IP) 320
Test 3.9 – Rebuild Assist via Get LBA Status (OF-FYI, FYI) 321
Case 1: Get LBA Status (OF-FYI, M) 321
Case 2: Reporting Allocated LBA Supported (OF-FYI, FYI) 322
Case 3: Descriptors not Returned When no LBAs are Allocated (OF-FYI, FYI) 322
Case 4: Get LBA Status - RALBAS=1, ATYPE=2 Reports User Data Changes (FYI) 323
Test 3.10 – Improved Performance Parameters (OF-FYI, FYI) 324
Case 1: Unaligned Write Command (OF-FYI, FYI) 324
Case 2: Unaligned Write Uncorrectable Command (OF-FYI, FYI) 324
Case 3: Unaligned Write Zeroes Command (OF-FYI, FYI) 325
Case 4: Unaligned Dataset Management Command (OF-FYI, FYI) 325
Case 5: Performance Valid Values (OF-FYI, M) 325
Case 6: Optimal Write Performance Value Fields Supported (OF-FYI, M) 326
Case 7: Optimal Read Performance Ignored, OPTRPERF=1 (OF-FYI, FYI) 326
Case 8: Optimal Read Performance, OPTRPERF=0 (OF-FYI, M) 326
Test 3.11 – Controller Memory Buffer (IP, IP) 328
Case 1: CMB Supported (IP, IP) 328
Test 3.12 – NVM Sets (OF-FYI, FYI) 329
Case 1: NVM Sets Supported (OF-FYI, FYI) 329
Case 2: RLL Supported in NVM Set (OF-FYI, FYI) 329
Case 3: Endurance Group Info Log Matches SMART/Health Log Summary (OF-FYI, M) 330
Case 4: Endurance Group Identifier = 0 (OF-FYI, M) 330
Case 5: Endurance Group Identifier Does not Exist for Set Feature (OF-FYI, M) 330
Case 6: Endurance Group Identifier Does not Exist for Get Feature (OF-FYI, M) 331
Case 7: Endurance Group Critical Warnings Configuration with Reserved Field (OF-FYI, M) 331
Case 8: Endurance Groups not Supported (OF-FYI, M) 331
Case 9: NVM Sets Supported (OF-FYI, M) 331
Case 10: Endurance Group Info Log Matches SMART/Health Log Summary (OF-FYI, M) 332
Case 11: NVM Set List, no zero Entry (OF-FYI, M) 332
Test 3.13 – Read Recovery Level (OF-FYI, FYI) 333
Case 1: Proper RLL Levels Supported (OF-FYI, FYI) 333
Case 2: NVM Sets Not Supported (OF-FYI, FYI) 333
Test 3.14 – Asymmetric Namespace Access Reporting (OF-FYI, FYI) 335
Case 1: ANA Log Page RGO=0 (OF-FYI, FYI) 335
Case 2: ANA Log Page RGO=1 (OF-FYI, FYI) 335
Case 3: Single AER for Namespace Attach Change Event (OF-FYI, FYI) 335
Case 4: Single AER for Namespace Deletion Change Event (OF-FYI, FYI) 336
Case 5: Single AER for Namespace Detachment Change Event (OF-FYI, FYI) 336
Case 6: No Namespace attached, Get Log Page ANA (OF-FYI, FYI) 336
Case 7: Asymmetric Namespace Access - IOS=1, OT=1 (OF-FYI, FYI) 337

© 2026 UNH–IOL

Test 3.15 – Namespace Granularity (OF-FYI, FYI) 338
Case 1: Create Namespace with all capacity Allocated (OF-FYI, M) 338
Case 2: Create Namespace with not all capacity Allocated (OF-FYI, FYI) 338
Test 3.16 – SQ Associations (OF-FYI, FYI) 340
Case 1: Associated Features Supported (OF-FYI, FYI) 340
Test 3.17 – Transport SGL Data Block Descriptor (OF-FYI) 341
Case 1: Correct Descriptor Format (OF-FYI) 341
Case 2: Incorrect Descriptor Format (OF-FYI) 341
Case 3: SGL Bit Bucket Descriptor Type (OF-FYI) 342
Case 4: Keyed SGL Data Block Descriptor Type (OF-FYI) 342
Case 5: Data Transfer Size Exceeds Total Metadata SGL Length (IP) 342
Case 6: Data Transfer Size Exceeds Total SGL Length (OF-FYI, FYI) 343
Group 4: Controller Registers 344
Test 4.1 – Offset 00h: CAP – Memory Page Size Maximum (MPSMAX) (OF, M) 345
Test 4.2 – Offset 00h: CAP – Memory Page Size Minimum (MPSMIN) (OF, M) 346
Test 4.3 – Offset 00h: CAP – Command Sets Supported (CSS) (OF, M) 347
Test 4.4 – Offset 00h: CAP – Doorbell Stride (DSTRD) (OF, M) 348
Test 4.5 – Offset 00h: CAP – Timeout (TO) (OF, M) 349
Test 4.6 – Offset 00h: CAP – Arbitration Mechanism Supported (AMS) (OF, M) 350
Test 4.7 – Offset 00h: CAP – Contiguous Queues Required (CQR) (OF, M) 351
Test 4.8 – Offset 00h: CAP – Maximum Queue Entries Supported (MQES) (OF, M) 352
Test 4.9 – Offset 0Ch–10h: INTMS – Interrupt Mask Set and INTMC – Interrupt Mask Clear (OF, M) 353
Test 4.10 – Offset 14h: CC – I/O Completions Queue Entry Size (IOCQES) (OF, M) 354
Test 4.11 – Offset 14h: CC – I/O Submission Queue Entry Size (IOSQES) (OF, M) 355
Test 4.12 – Offset 14h: CC – Shutdown Notification (SHN) (OF, M) 356
Test 4.13 – Offset 14h: CC – Arbitration Mechanism Selected (AMS) (OF, M) 358
Test 4.14 – Offset 14h: CC – I/O Command Set Selected (CSS) (OF, M) 359
Test 4.15 – Offset 14h: CC – Enable (EN) (OF, M) 360
Test 4.16 – Offset 1Ch: CSTS – Shutdown Status (SHST) (OF, M) 361
Test 4.17 – Offset 1Ch: CSTS – Controller Fatal Status (CFS) (OF, M) 363
Test 4.18 – Offset −08h: CAP – Version (VS) (OF, M) 364
Test 4.19 – Offset 1Ch: Controller Processing Paused (PP) (FY-OF, M) 365
Test 4.20 – Offset 68h: Controller Ready Independent of Media Timeout (CRIMT) (FYI-OF, M) 366
Test 4.21 – Offset 00h: NVM Subsystem Shutdown Enhancements (FYI-OF, FYI) 367
Case 1: NVM Subsystem Shutdown Enhancements VER > 2.0 (OF-FYI, FYI) 367
Case 2: NVM Subsystem Shutdown Enhancements Supported (FYI) 367
Case 3: NVM Subsystem Shutdown Enhancements Not Supported (M) 367
Case 4: Controller Level Reset - ST=0, SHST=10b (FYI) 367
Case 5: Controller Level Reset Doesn’t Affect Shutdown Status (FYI) 368
Case 6: Shutdown Type Remains Set Until NVM Subsystem Reset (FYI) 368
Case 7: CLR Not Initiated by NVM Subsystem Reset Does Not Abort NVM Subsystem Shutdown (FYI) 369
Group 5: System Memory Structure 370
Test 5.1 – Page Base Address and Offset (PBAO) (OF, M) 371
Test 5.2 – Completion Queue Entry (M) 372
Test 5.3 – Status Field Definition (OF, M) 373
Test 5.4 – Generic Command Status Definition (OF-FYI, M) 374
Case 1: Read Success (OF-FYI, M) 374
Case 2: Invalid Opcode Error (OF-FYI, M) 375
Case 3: Invalid Field Error (OF-FYI, M) 375
Case 4: Invalid Namespace or Format Error (OF-FYI, M) 375
Case 5: Command Sequence Error (OF-FYI, M) 375
Case 6: LBA Out of Range (OF-FYI, M) 375
Test 5.5 – Command Specific Errors Definition (OF-FYI, M) 377

© 2026 UNH–IOL

Case 1: Abort Command Limit Exceeded (OF-FYI, M) 379
Case 2: Asynchronous Event Request Limit Exceeded (OF-FYI, M) 379
Case 3: Invalid Firmware Slot (OF-FYI, M) 379
Case 4: Feature Identifier Not Saveable (OF-FYI, M) 379
Case 5: Feature Not Changeable (OF-FYI, M) 380
Case 6: Feature Not Namespace Specific IV=1 (OF-FYI, M) 380
Case 7: Overlapping Range (OF-FYI, M) 380
Test 5.6 – Media and Data Integrity Errors Definition (OF-FYI, M) 382
Case 1: Media and Data Integrity Errors Definition (OF-FYI, M) 383
Group 6: Controller Architecture 384
Test 6.1 – Controller Level Reset – Controller Reset (PCIeC 6.3) (OF, M) 385
Test 6.2 – Self-Reported Drive Power (OF, M) 386
Case 1: Interval Power Measurement Value Set to Non-Zero Value When Required by Related Fields (OF-FYI, FYI) 386
Case 2: Power Histogram Descriptors in Ascending Order (OF-FYI, FYI) 386
Case 3: Power Measurement Action Field Start Measurement Clears Number Power Histogram Descriptors (OF-FYI, FYI) 386
Case 4: Power Measurement Log Page Validation (OF-FYI, FYI) 387
Case 5: Product of Sample Rate Scale and Sample Rate Value <= 1 Second (OF-FYI, FYI) 387
Case 6: Power Threshold Exceeded Triggers Asynchronous Event (OF-FYI, FYI) 388
Case 7: Starting Power Measurement Clears and Changes Stop Measurement Time Remaining Field (OF-FYI, FYI) 388
Case 8: Power Measurement Timer SMTR Decrements Over Time (OF-FYI, FYI) 388
Case 9: Stop Measurement Time Greater Than The Maximum (OF-FYI, FYI) 389
Case 10: Recording Power Measurements Change Values in Log Page (OF-FYI, FYI) 389
Test 6.3 – Controller Level Reset – Controller Reset (PCIeC 6.3) (OF, M) 390
Group 7: Reservations 391
Test 7.1 – Reservation Report Command (OF-FYI, M) 392
Case 1: No Registrants (OF-FYI, M) 392
Case 2: Host is a Registrant (OF-FYI, M) 393
Case 3: 64 Bit Host Identifier (OF-FYI, FYI) 393
Case 4: 128 Bit Host Identifier (OF-FYI, FYI) 394
Case 5: Dynamic Controller Not Associated with Host (OF-FYI, FYI) 395
Test 7.2 – Reservation Registration (OF-FYI, M) 396
Case 1: Basic Operation (OF-FYI, M) 396
Case 2: Re-registration (OF-FYI, M) 397
Case 3: Replace Registration Key (OF-FYI, M) 397
Case 4: Multiple Hosts Dual Port Devices Only (FYI) 398
Case 5: Reservation Persistence (OF-FYI, FYI) 399
Case 6: Controller ID field FFFFh when Registrant Namespace is Detached (OF-FYI, FYI) 400
Test 7.3 – Unregistering (OF-FYI, M) 401
Case 1: Unregistering with Reservation Register Command (OF-FYI, M) 401
Case 2: Unregistering due to Preemption Dual Port Devices Only (OF-FYI, FYI) 402
Test 7.4 – Acquiring a Reservation (OF-FYI, M) 403
Case 1: Basic Operation (OF-FYI, M) 403
Case 2: Error Conditions (OF-FYI, M) 404
Case 3: Multiple Hosts Dual Port Devices Only (OF-FYI, FYI) 405
Test 7.5 – Releasing a Reservation (OF-FYI, M) 407
Case 1: Release with Reservation Release Command (OF-FYI, M) 407
Case 2: Reservation Release Command Error Conditions (OF-FYI, M) 408
Case 3: Multiple Hosts Dual Port Devices Only (OF-FYI, FYI) 409
Case 4: Release Due to Unregister (OF-FYI, M) 410
Test 7.6 – Preempting a Reservation (OF-FYI, FYI) 412
Case 1: Write Exclusive - All Registrants or Exclusive Access - All Registrants Dual Port Devices Only (OF-FYI, FYI) 412
Case 2: Other Registration Types Dual Port Devices Only (OF-FYI, FYI) 413
Case 3: Self-preemption Dual Port Devices Only (OF-FYI, FYI) 415
Case 4: Preempt and Abort Dual Port Devices Only (OF-FYI, FYI) 416
Case 5: Preempt Attempt when CRKEY does not Match Dual Port Devices Only (OF-FYI, FYI) 416

© 2026 UNH–IOL

Test 7.7 – Clearing a Reservation (OF-FYI, M) 418
Case 1: Basic Operation with Reservation Release Command (OF-FYI, M) 418
Case 2: Error Conditions (OF-FYI, M) 419
Test 7.8 – Command Behavior with Different Reservation Types (OF-FYI, M) 421
Case 1: Write Exclusive Dual Port Devices Only (OF-FYI, M) 421
Case 2: Exclusive Access Dual Port Devices Only (OF-FYI, M) 422
Case 3: Write Exclusive - Registrants Only or Write Exclusive - All Registrants Dual Port Devices Only (OF-FYI, M) 423
Case 4: Exclusive Access - Registrants Only or Exclusive Access - All Registrants Dual Port Devices Only (OF-FYI, M) 424
Test 7.9 – Reservation Notification Log Page (OF-FYI, FYI) 426
Case 1: Retrieve Log (OF-FYI, FYI) 426
Case 2: Empty Log (OF-FYI, FYI) 427
Case 3: Wrapped Log Count (OF-FYI, FYI) 427
Group 8: Namespace Management 429
Test 8.1 – Namespace Management Identify Command (PCIeC 9.1) (OF-FYI, M) 430
Case 1: CNS 10h & 11h – Namespace Lists (OF, M) 430
Case 2: CNS 12h – Controller List – Controllers Attached to a Namespace (OF, M) 430
Case 3: CNS 13h – Controller List – All Controllers (OF-FYI, M) 431
Case 4: Common Namespace Data Structure (OF, M) 431
Case 5: NSID Uniqueness (OF, M) 431
Case 6: Namespace management not supported (OF, M) 432
Case 7: Namespace Management, OAES bit 8 (OF-FYI, M) 432
Case 8: Namespace Identifier with Invalid Namespace or Format (OF-FYI, M) 432
Test 8.2 – Namespace Management Command (PCIeC 9.2) (OF-FYI, M) 433
Case 1: Namespace Creation – Exceed Number Supported (OF-FYI, M) 433
Case 2: Namespace Deletion (OF-FYI, M) 434
Case 3: Namespace Creation – Insufficient Capacity (OF-FYI, M) 434
Case 4: Namespace Deletion when NSID=FFFFFFFFh (OF-FYI, M) 435
Case 5: Creation with NVMSETID & ENDGID Cleared to Zero (OF-FYI, M) 436
Case 6: Creation with NVMSETID=0 & valid ENDGID (OF-FYI, M) 436
Case 7: Creation with NVMSETID=non-zero value & ENDGID=0 (OF-FYI, FYI) 437
Case 8: Creation with valid NVMSETID existing in ENDGID (OF-FYI, FYI) 437
Case 9: Creation with valid existing in ENDGID, Invalid NVMSETID (OF-FYI, FYI) 437
Case 10: Creation of Namespace, LBAFEE=0h & ELBAS=1h (OF-FYI, FYI) 437
Test 8.3 – Namespace Attachment Command (PCIeC 9.3) (OF-FYI, M) 439
Case 1: Namespace Attachment (OF-FYI, M) 439
Case 2: Namespace Detachment (OF-FYI, M) 440
Case 3: Namespace Attachment, Incorrect Command Set (OF-FYI, FYI) 440
Case 4: Namespace Attachment, Command Set not Enabled (OF-FYI, FYI) 440
Case 5: MAXDNA Namespace Attachment Limit Exceeded (OF-FYI, FYI) 441
Case 6: MAXCNA Namespace Attachment Limit Exceeded (OF-FYI, FYI) 441
Test 8.4 – Dispersed Namespaces (OF-FYI, FYI) 442
Case 1: Get Log LID 17h, Invalid namespaces (OF-FYI, FYI) 442
Case 2: Get Log LID 17h, Non-Dispersed Namespaces (OF-FYI, FYI) 442
Case 3: Get Log LID 17h NQN matches ID CTRL (OF-FYI, FYI) 442
Case 4: DISNSRS 0h sent to Dispersed Namespace (OF-FYI, FYI) 443
Group 9: Flexible Data Placement 444
Test 9.1 – FDP Configuration Log Page, Log Identifier 20h (OF-FYI, FYI) 445
Case 1: FDP Configuration Log Page, Version (OF-FYI, FYI) 445
Case 2: FDP Configuration Log Page, FDPVWC=1 (OF-FYI, FYI) 445
Case 3: FDP Configuration Log Page, NRG>1 & RGIF not 0 (OF-FYI, FYI) 445
Case 4: FDP Configuration Log Page, NRG not 0 (OF-FYI, FYI) 445
Case 5: FDP Configuration Log Page, NRUH not 0 (OF-FYI, FYI) 446
Case 6: FDP Configuration Log Page, MAXPIDS<NRG*NRUH (OF-FYI, FYI) {#case-9-1-6 } 446
Case 7: FDP Configuration Log Page, Number of Namespaces Supported (OF-FYI, FYI) 446
Case 8: FDP Configuration Log Page, 8-byte Boundary Cleared to 0 (OF-FYI, FYI) 446
Test 9.2 – Reclaim Unit Handle Usage Log Page, Log Identifier 21h (OF-FYI, FYI) 448
Case 1: Reclaim Unit Handle Usage Log Page, Invalid NSID (OF-FYI, FYI) 448
Case 2: Reclaim Unit Handle Usage Log Page, FDP Disabled (OF-FYI, FYI) 448

© 2026 UNH–IOL

Case 3: Reclaim Unit Handle Usage Log Page, NRUH (OF-FYI, FYI) 448
Case 4: Reclaim Unit Handle Usage Log Page, Reclaim Unit Handle Attributes (OF-FYI, FYI) 448
Test 9.3 – FDP Statistics Log Page, Log Identifier 22h (OF-FYI, FYI) 450
Case 1: FDP Statistics Log Page, FDP Disabled (OF-FYI, FYI) 450
Case 2: FDP Statistics Log Page, Invalid NSID (OF-FYI, FYI) 450
Case 3: FDP Statistics Log Page, Statistics Reset to 0h (OF-FYI, FYI) 450
Case 4: FDP Statistics Log Page, HBMW (OF-FYI, FYI) 451
Case 5: FDP Statistics Log Page, MBE (OF-FYI, FYI) 451
Test 9.4 – FDP Events Log Page, Log Identifier 23h (OF-FYI, FYI) 452
Case 1: FDP Events Log Page, Events Disabled (OF-FYI, FYI) 452
Case 2: FDP Events Log Page, Invalid NSID (OF-FYI, FYI) 452
Case 3: FDP Events Log Page, FDP Disabled (OF-FYI, FYI) 452
Case 4: FDP Events Log Page, N=0h (OF-FYI, FYI) 452
Case 5: FDP Events Log Page, N Increments (OF-FYI, FYI) 453
Case 6: FDP Events Log Page, Invalid Placement Identifier Event (03h) (OF-FYI, FYI) 453
Case 7: FDP Events Log Page, Order of Occurrence (OF-FYI, FYI) 454
Case 8: FDP Events Log Page, events Discarded (OF-FYI, FYI) 454
Case 9: FDP Events Log Page, FDPET (OF-FYI, FYI) 455
Test 9.5 – FDP Identify Controller, Log CNS 01h (OF-FYI, FYI) 456
Case 1: FDP Identify Controller, FDPS & FCM (OF-FYI, FYI) 456
Test 9.6 – FDP Namespace Management Operations (OF-FYI, FYI) 457
Case 1: FDP Namespace Creation, NPHNDLS>NRUH or 128 (OF-FYI, FYI) 457
Case 2: FDP Namespace Creation, NPHNDLS=NRUH or 128 (OF-FYI, FYI) 457
Case 3: FDP Namespace Creation, NPHNDLS=0h (OF-FYI, FYI) 457
Case 4: FDP Namespace Creation, NPHNDLS=0h (OF-FYI, FYI) 458
Case 5: FDP Namespace Creation, Controller Specified RUH in Placement List (OF-FYI, FYI) 458
Case 6: FDP Namespace Creation, Duplicate Placement Handles (OF-FYI, FYI) 459
Case 7: FDP Namespace Creation, Duplicate Placement Handle > NRUH (OF-FYI, FYI) 459
Case 8: FDP Namespace Creation, Shared Placement Handle, Different FLBAS (OF-FYI, FYI) 459
Test 9.7 – FDP Writes (OF-FYI, FYI) 461
Case 1: FDP Write Command FDPE=1h, DTYPE=2h, Invalid DSPEC (OF-FYI, FYI) 461
Case 2: FDP Write Command FDPE=1h, DTYPE=0h, and DSPEC=0h (OF-FYI, FYI) 461
Case 3: FDP Write Command FDPE=1h, DTYPE=2h, and Valide Placement Identifier (OF-FYI, FYI) . 461
Case 4: FDP Write Command FDPE=1h, DTYPE=2h, and Invalid DSPEC (OF-FYI, FYI) 462
Case 5: FDP Write Uncorrectable Command FDPE=1h, DTYPE=0h, and DSPEC=0h (OF-FYI, FYI) . . 463
Case 6: FDP Write Uncorrectable Command FDPE=1h, DTYPE=2h, and valid DSPEC (OF-FYI, FYI) .463
Case 7: FDP Write Zeros Command FDPE=1h, DTYPE=2h, and Invalid DSPEC (OF-FYI, FYI) 464
Case 8: FDP Write Zeros Command FDPE=1h, DTYPE=0h, and DSPEC=0h (OF-FYI, FYI) 464
Case 9: FDP Write Zeros Command FDPE=1h, DTYPE=2h, and Valid DSPEC (OF-FYI, FYI) 465
Test 9.8 – FDP NVM Format Operations (OF-FYI, FYI) 466
Case 1: FDP Format NVM – Shared Placement Handle, Unique FLBAS (OF-FYI, FYI) 466
Test 9.9 – I/O Management Receive Management Operation (1h) (OF-FYI, FYI) 467
Case 1: FDP I/O Management Receive (MO=1h) (OF-FYI, FYI) 467
Case 2: FDP I/O Management Receive (MO=1h), FDP Disabled (OF-FYI, FYI) 467
Case 3: FDP I/O Management Receive (MO=1h), Invalid Namespace (OF-FYI, FYI) 467
Case 4: FDP I/O Management Receive (MO=1h), Extra Zeros (OF-FYI, FYI) 467
Test 9.10 – I/O Management Send Management Operation (1h) (OF-FYI, FYI) 469
Case 1: FDP I/O Management Send (MO=1h), FDP Disabled (OF-FYI, FYI) 469
Case 2: FDP I/O Management Send (MO=1h), NPIC>MAXPIDS (OF-FYI, FYI) 469
Case 3: FDP I/O Management Send (MO=1h), NPID>NRG*NRUH (OF-FYI, FYI) {#case-9-10-3 } . . . 469
Test 9.11 – FDP Enable Flexible Data Feature (1Dh) (OF-FYI, FYI) 470
Case 1: FDP Feature (FID 1Dh) (OF-FYI, FYI) 470
Case 2: FDP Feature (FID 1Dh), Existing Namespace (OF-FYI, FYI) 470
Case 3: FDP Feature (FID 1Dh), SV=0 (OF-FYI, FYI) 471
Case 4: FDP Enabled, Streams Directive & Directive Operation 01h (OF-FYI, FYI) 471
Test 9.12 – FDP Events Feature Identifier (1Eh) (OF-FYI, FYI) 472
Case 1: FDP Events Feature – Events Enabled (OF-FYI, FYI) 472
Case 2: FDP Events Feature – Invalid Namespace (OF-FYI, FYI) 472
Case 3: FDP Events Feature – FDP Disabled (OF-FYI, FYI) 472
Case 4: FDP Events Feature – Events Enabled (OF-FYI, FYI) 473

© 2026 UNH–IOL

Test 9.13 – FDP Data Placement Directive (OF-FYI, FYI) 474
Case 1: FDP Data Placement Directive (OF-FYI, FYI) 474
Case 2: Data Placement Directive, Persistent Across Reset (OF-FYI, FYI) 474
Case 3: Data Placement Directive, NSID=FFFFFFFFh (OF-FYI, FYI) 474
Case 4: Data Placement Directive, FDP Disabled (OF-FYI, FYI) 475
Case 5: Data Placement Directive, Invalid Operation (OF-FYI, FYI) 475
Test 9.14 – RUH Scope Reporting (OF-FYI, FYI) 476
Group 10: Reachability Groups 477
Test 10.1 – Reachability Groups (OF-FYI, FYI) 478
Case 1: Reachability Groups Log with RGO = 1h (OF-FYI, FYI) 478
Case 2: Reachability Groups Log Change Count (OF-FYI, FYI) 478
Case 3: Reachability Groups Log no Namespaces Attached (OF-FYI, FYI) 478
Case 4: Reachability Association Log with RGO = 1h (OF-FYI, FYI) 479
Case 5: Reachability Association Log no Namespaces Attached (OF-FYI, FYI) 479
Case 6: Reachability Association Log Change Count (OF-FYI, FYI) 479
Case 7: Reachability Group Identifier When Reachability Group Reporting is Not Supported (OF-FYI, M) 480
Group 11: Boot Partition 481
Test 11.1 – Boot Partition Write Protection (OF-FYI, FYI) 482
Case 1: Valid Values (OF-FYI, M) 482
Case 2: Set Feature, RPMB Enabled (OF-FYI, FYI) 483
Case 3: Set Feature, Feature Not Changeable (OF-FYI, FYI) 483
Case 4: Get Feature, RPMB Enabled (OF-FYI, FYI) 483
Case 5: Boot Partitions & RPMB Not Supported (OF-FYI, FYI) 484
Case 6: Boot Partitions & RPMB Supported (OF-FYI, FYI) 484
Case 7: RPMB Boot Partition Write Protection Not Supported (OF-FYI, FYI) 484
Case 8: Enable While in Write Locked Until Power Cycle State (OF-FYI, FYI) 485
Case 9: Get Feature While RPMB Enabled (OF-FYI, FYI) 485
Group 12: PCIe Infrastructure for Live Migration 486
Test 12.1 – Host Managed Live Migration Support (FYI) 487
Case 1: HMLMS Support Valid Values (OF-FYI, FYI) 487
Case 2: CNS=01h Host Managed Live Migration Support (OF-FYI, M) 487
Test 12.2 – Host Managed Live Migration: Migration Receive (FYI) 488
Case 1: Migration Receive command - Management Operation=0h (FYI) 488
Case 2: Migration Receive command - Offset Lower Dword Aligned (FYI) 488
Case 3: Migration Receive command - Management Operation=0h, CSUP=0h, CSUP=1h (FYI) 488
Case 4: Migration Receive command - Management Operation=0h, Invalid CSVI (FYI) 488
Case 5: Migration Receive command - Management Operation=0h, CSVI=0h (FYI) 489
Case 6: Migration Receive command -Management Operation=0h, Invalid CSUUIDI (FYI) 489
Case 7: Controller State Data Structure Version Support and Check (FYI) 489
Case 8: Controller State Data Structure NVMe Controller State Size Verification (FYI) 489
Test 12.3 – Host Managed Live Migration: Migration Send (FYI) 491
Case 1: Migration Send Command - Management Operation=0h, DUDMQ=1h (FYI) 491
Case 2: Migration Send Command - Management Operation=0h, Command Processing (FYI) 491
Case 3: Migration Send command - Management Operation=0h, Controller Already Suspended (FYI) .. 491
Case 4: Migration Send command - Management Operation=1h, Controller Not Suspended (FYI) 492
Case 5: Migration Send command - Management Operation=2h, Controller State Not Verified or Committed (FYI) 492
Case 6: Migration Send command - Management Operation=2h, Controller Not Suspended or Enabled (FYI) 492
Case 7: Migration Send command - Management Operation=2h, SEQIND (FYI) 493
Case 8: Migration Send command - Management Operation=2h, SEQIND=02h (FYI) 493
Case 9: Migration Send command - Management Operation=2h, CSOL not Dword Aligned (FYI) 493
Case 10: Migration Send command - Management Operation=2h, NUMD=0h (FYI) 493
Case 11: Migration Send command - Management Operation=2h, Controller State Offset > NUMD (FYI) 493
Case 12: Migration Send with Invalid Controller State Version Index (CSVI) in CDW11 (FYI) 494
Case 13: Migration Send Command with Mismatched Controller State Version Index (CSVI) and NVMe Controller State Size (NVMECSS) (FYI) 494

© 2026 UNH–IOL

Case 14: Migration Send with Invalid CSUUIDI (FYI) 494
Case 15: Set Controller State via Migration Send Command when NVMECSS != 0 and IOSQs/IOCQs Exist (FYI) 495
Case 16: Migration Send Command Set Controller State Operation with Unexpected Vendor Specific Size (FYI) 495
Case 17: Migration Send Command Set Controller State with No Format Indices Specified (FYI) 496
Test 12.4 – Controller Data Queue Command (FYI) 497
Case 1: Controller Data Queue command - MCMR Exceeded (FYI) 497
Case 2: Controller Data Queue Command - NMCMR Exceeded (FYI) 497
Case 3: Controller Data Queue Command – MNSUDMQ Exceeded (FYI) 497
Case 4: Controller Data Queue command - Invalid PRP Offset (FYI) 498
Case 5: Controller Data Queue command - PRP Entry 1 Invalid Memory (FYI) 498
Case 6: Controller Data Queue command - Invalid CDQSIZE (FYI) 498
Case 7: Controller Data Queue command - Equal CQS and CNTLID (FYI) 498
Case 8: Controller Data Queue command - Exceed MCUDMQ (FYI) 499
Case 9: Controller Data Queue command - CDQ ID does not exist (FYI) 499
Case 10: Log User Data Changes - Controller Data Queue Entry Phase Tag (FYI) 499
Case 11: Log User Data Changes - Controller Reset Stops Logging (FYI) 500
Case 12: Log User Data Changes - ESA = 000b, LBACIR = 000b (FYI) 500
Case 13: ESA Values 000-011b, 111b (FYI) 500
Case 14: Empty CDQ with Invalid Head Pointer Field (FYI) 501
Case 15: Tail Pointer Not Associated with the Controller Data Queue (FYI) 501
Case 16: Specifying a Non-Existent Controller Data Queue (FYI) 502
Test 12.5 – Host Managed Live Migration: Track Send (FYI) 503
Case 1: Tracked Memory Changes - Reported Memory Range Granularity Alignment (FYI) 503
Case 2: Log User Data Changes - Tracking After LACT=1b, No Tracking After LACT=0b (FYI) 503
Case 3: Log User Data Changes - Already Logging Controller Data Queue (FYI) 504
Case 4: Log User Data Changes - CDQID Associated with Suspended Controller (FYI) 504
Case 5: Tracked Memory Changes - Memory Range Tracking Granularity Bounds (FYI) 505
Case 6: Tracked Memory Changes - RNMRTD > CMMRTD (FYI) 505
Case 7: Tracked Memory Changes - RNMRTD > NMMRTD (FYI) 505
Case 8: Tracked Memory Changes - Already Tracking Controller (FYI) 506
Case 9: Track Memory Changes - MRTLL=1b, Descriptor LEN is a power of 2 (FYI) 506
Case 10: Track Memory Changes - TACT=0h, Track Memory Changes Data Structure Ignored (FYI) 506
Case 11: Track Memory Changes - Tracking After TACT=1b, No Tracking After TACT=0b (FYI) 506
Case 12: Track Memory Changes - Controller is Suspended (FYI) 507
Case 13: Track Memory Changes - Incorrect Version (FYI) 507
Case 14: Track Memory Changes - Descriptor SADDR not aligned to RMRTG (FYI) 507
Case 15: Log User Data Changes - Track Send Clears Controller Data Queue Full Error Event (FYI) 508
Case 16: Log User Data Changes - LBACIR=10b, NSID & NLB Cleared to 0h (FYI) 509
Case 17: Log User Data Changes - Controller Data Queue Doesn't Exist (FYI) 509
Test 12.6 – Host Managed Live Migration: Track Receive (FYI) 510
Case 1: Track Receive Command VER=0h (FYI) 510
Case 2: Track Receive Command MTR=0b (FYI) 510
Case 3: Track Receive Operations Clear Previously Tracked Memory (FYI) 510
Group 13: Configurable Device Personality 512
Test 13.1 – Personality Freeze and Unfreeze Behavior (FYI) 513
Case 1: Changing a Frozen Personality Maintains Values (OF-FYI, FYI) 513
Case 2: Feature Not Changeable When Personality Requirements Not Met (OF-FYI, FYI) 513
Case 3: Set Features with PERFS = 1 and CHPS = 1 Returns Feature Not Changeable (OF-FYI, FYI) 514
Case 4: Freeze and Unfreeze All Personalities With PERID=FFh (OF-FYI, FYI) 514
Case 5: Unfreezing Personality with Unsupported Authentication Method Remains Frozen (OF-FYI, FYI) 515
Case 6: Security Personality – Attribute Change Logged in Persistent Event Log (OF-FYI, FYI) 515
Case 7: CDP Programmable Key – Pre-Existing Key (OF-FYI, FYI) 516
Case 8: CDP Programmable Key – Key Not Programmed After Personality Switch (OF-FYI, FYI) 516
Case 9: CDP Programmable Key – Host Key Not Purged After Reset or Sanitize (OF-FYI, FYI) 517
Case 10: CDP Programmable Key – Improper HMAC Size (OF-FYI, FYI) 517
Case 11: CDP Programmable Key – Personality Unfreeze Request Purges Nonce (OF-FYI, FYI) 518
Case 12: Security Personality – Security Protocol Status (OF-FYI, FYI) 519
Case 13: Security Personality – SSP=0h (OF-FYI, FYI) 519

© 2026 UNH–IOL

Case 14: Security Receive – Security Receive without Security Send (OF-FYI, FYI) 520
Case 15: CDP Authentication – CDP Request REQDFV=0h, DFHL=4h (OF-FYI, FYI) 520
Case 16: CDP Authentication – CDP Response RESPDFV=0h, DFHL=4h, RESPDFS=07h (OF-FYI, FYI) 520
Case 17: CDP Authentication – Derived HMAC Key Programming Responses (OF-FYI, FYI) 520
Case 18: CDP Authentication – CLR Purges Nonce (OF-FYI, FYI) 521
Test 13.2 – Device Personalities Log Page Behavior (FYI) 522
Case 1: PEL entry creation with PERID set to 00h (OF-FYI, FYI) 522
Case 2: Personality 0 is Manufacturing Default (OF-FYI, FYI) 522
Case 3: Version 0h Device Personality 1Dh (OF-FYI, FYI) 522
Case 4: Personality changed effects PSCUDE and UDCC (OF-FYI, FYI) 523
Case 5: PMDSS Clears and Sets (OF-FYI, FYI) 523
Test 13.3 – Pending State Verifications (FYI) 525
Case 1: Sanitize Prohibited During Pending Personality Change (OF-FYI, FYI) 525
Case 2: CDP Modification Restrictions during Pending Change (OF-FYI, FYI) 525
Case 3: CDP Get Features All Personalities with Pending Change (OF-FYI, FYI) 526
Group 14: Sanitize Per Namespace 527
Test 14.1 – Sanitize Namespace Command (OF-FYI, FYI) 528
Case 1: Delete Namespace Aborted with Sanitize Namespace in Progress (OF-FYI, FYI) 528
Case 2: Namespace Data Erased (NDE) (OF-FYI, FYI) 528
Case 3: Sanitize Namespace - Overwrite Passes Completed (OPC) (OF-FYI, FYI) 529
Case 4: Sanitize Namespace - Sanitize Never Started (SOS 000b) (OF-FYI, FYI) 529
Case 5: Sanitize Namespace Concurrent Operations Limit Validation (OF-FYI, FYI) 530
Case 6: Sanitize Namespace NSID Target Validation and Rejection Constraints (OF-FYI, M) 530
Case 7: Sanitize Namespace command Option and Rejection Constraints (OF-FYI, FYI) 530
Case 8: Memory Namespace Sanitize Data Zeroing and Action Equivalence (OF-FYI, FYI) 531
Appendix A: Default Test Setup 532
Appendix B: Notes on Test Procedures 533
Appendix C: Test Tools 534
Appendix D: NVMe Integrators List Requirements 535

© 2026 UNH–IOL

Modification Records

2013 May 21 (Version 1.0)

David Woolf: Changed version of test suite to 1.0 after completion of NVMe Plugfest.

2013 October 22 (Version 1.1 DRAFT)

David Woolf: Updates to tests 2.1 and 2.4.

2013 December 16 (Version 1.1 DRAFT)

David Woolf: Updates to tests 1.1, 1.2, 1.3, 1.5, 2.1, 2.2, 2.3, 2.4, 2.5, 4.3, 4.4, 4.5, 4.6, 4.7, and 4.8. Most of the changes involve clarifying or removing observable results that could not be easily observed.

2013 December 19 (Version 1.1)

David Woolf: Added Appendix C with information about potential test tools. Removed references to conformance testing of NVMe Hosts, as this is not currently a requirement for including NVMe Hosts on the NVMe Integrators List.

2014 May 27 (Version 1.1)

David Woolf: Edited procedures to test 1.2 and 4.5 to make them easier to perform.

2014 June 5 (Version 1.1)

David Woolf: Corrected test numbering in Group 5 and Group 6. Clarified that tests 3.1, 3.2, and 4.6 are optional. Corrected reference in test 5.3.

2014 July 7 (Version 1.1)

David Woolf: Added test 6.4.

2014 July 10 (Version 1.1)

David Woolf: Added test 2.7, modified procedure to test 2.1.

2014 July 14 (Version 1.1b)

David Woolf: Change specification references to 1.1b revision of specification.

2014 August 14 (Version 1.1b)

David Woolf: Added Group 7 tests for Reservations.

2014 September 18 (Version 1.1b)

David Woolf: Added notes Group 7 tests for Reservations to indicate that these tests are only applicable to device which support reservations.

2014 September 29 (Version 1.1b)

Mike Bogochow: Updated references and fixed errata.

2014 October 16 (Version 1.1b)

David Woolf: Clarification on Mandatory and Optional tests in Group 2.

2015 April 9 (Version 1.2)

David Woolf: Prepared document for 1.2 revisions.

2015 April 13 (Version 1.2)

David Woolf: Added tests 8.1 and 8.2.

2015 June 24 (Version 1.2.1)

Mike Bogochow: Fixed errata and updated references, procedures, and observable results for all tests in groups 1–6.

© 2026 UNH–IOL

2015 November 23 (Version 1.2.1)

Mike Bogochow: Added Group 9 Tests.

2016 March 2 (Version 1.2.2)

Mike Bogochow: Fixed errata, clarified language, and updated procedures and observable results of various existing tests.

2016 March 21 (Version 1.3.0)

Mike Bogochow: Rewrote Groups 7 and 8. Added “Test 1.7 - Asynchronous Events”. Added “Test 1.4 - Case 2: Full Queue Condition”. Fixed Purpose wording for most tests.

2016 May 19 (Version 6.0 r01)

David Woolf:

Adopted new document numbering scheme, rather than using numbers that are close to the current specification release, test suites will follow a numbering scheme based on the Integrators List version, which is incremented with each plugfest.

2016 May 24 (Version 6.0 r02)

David Woolf:

Added tags to titles of FYI tests to help readers determine what tests were mandatory versus FYI. Added Appendix outlining which tests were Mandatory, Optional (dependent on feature support), and FYI. Previously this information was in the NVMe Integrators List Policy Document.

2016 May 26 (Version 6.0 r03)

David Woolf:

Added requirement to tests 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 2.3, 2.4, 2.6 that all Reserved fields be checked and that they are set to 0.

2016 June 9 (Version 6.0 r04)

David Woolf:

Added tests 1.8,1.9, 2.8, 2.9, 3.4, 3.5, 4.18, 8.3.

2016 June 13 (Version 6.0 r05)

David Woolf:

Added tests 2.10, 10.1. Added Case 2 to test 2.9 to address the case where Atomic Boundaries are supported.

2016 June 27 (Version 6.0 r06)

David Woolf:

Added new subtests to 1.1, 1.2, 1.3, 1.6, 2.1, 2.3, 2.4, 2.5, 2.6, 2.7. Added checking of CSTS.NSSRO to test 6.4.

2016 June 28 (Version 6.0 r07)

David Woolf:

Added new subtests to 1.4. Added Mandatory, Optional, FYI designations to all Tests and Cases within Tests. Edited Appendix D for a clearer description of the meaning of Mandatory, Optional, and FYI test designations and NVMe Integrators List requirements.

2016 July 5 (Version 6.0 r08)

David Woolf:

Added Case 9 in Test 1.4, added Case 11 in Test 2.3, added Case 11 in Test 2.4. Added Case 4 and 5 to Test 2.2.

2016 July 7 (Version 6.0 r09)

David Woolf:

Replaced “Optional” indication with “Mandatory if Supported” in all test titles and Appendix D. Flagged ‘Test 1.6 – Format NVM’ as “Mandatory if Supported”, previously it had been indicated

© 2026 UNH–IOL

as “Mandatory”. Replaced ‘optional’ with “Mandatory if Supported” in the ‘Possible Problems’ section of several tests.

2016 July 7 (Version 6.0 r10)

David Woolf:

Adjusted procedure of Case 2 of test 2.6. Fixed typos in Test 1.6 Test Cases 2 and 3.

2016 July 14 (Version 6.0 r11)

David Woolf:

Adjusted procedure of subtest cases in Tests 2.1, 2.3, 2.4, 2.5, 2.6, 2.7 dealing with Invalid Namespace ID Errors.

2016 July 21 (Version 6.0 r12)

David Woolf:

Adjusted procedure of Case 1 in Tests 9.2 to check both the case where number of Namespaces is exceeded and the Namespace size is exceeded. Corrected typos in Test Procedure for Test 2.4 cases 8, 9, 10,11 and Test 2.3 case 11. Adjusted procedure in Test 1.3 Cases 7 and 8 so that enough bytes are read to increment the Data Units Read/Written value. Adjusted procedure in Test 2.5 Case 2, 3, 4, 5 and Test 2,7 case 5.

2016 July 21 (Version 6.0 r13)

David Woolf:

Adjusted procedures to Test 1.3 Case 1 and 4 to only check Mandatory values. Test 1.3 Case 5 was updated to check that error log entries properly increment. Clarified Test 1.3 Case 4.

2016 August 30 (Version 6.0)

David Woolf:

Final version published to UNH-IOL site ahead of October 2016 NVMe Plugfest #6.

2016 September 26 (Version 6.1 r01)

David Woolf:

- Added indication to several tests Group 7, that some cases are only applicable to Dual Port Devices.
- Updated Test 7.1 Case 2 Observable results steps 3 and 7.
- Updated Test 7.2 Case 3 Procedure step 1.b.vi.
- Separated Test 9.2 Case 1 into 2 separate cases, now Test 9.2 Case 1 addressing having too many Namespaces, and Case 3 addressing having insufficient capacity.
- Modified Test 1.3 Case 4 to accommodate for changes between specs 1.2a and 1.2.1.
- Modified Test 2.3 Case 3 and Test 2.4 Case 3.

January 17, 2017 (Version 7.0 r01)

David Woolf:

- Changed indication of FYI to Mandatory or In Progress for many tests based on results of Plugfest #6.
- Corrected Observable Results for Test 1.3 Case 2 and 3 to expect ‘Invalid Log Page’ rather than ‘Invalid Field in Command’.
- Fixed typo in Test 1.5.
- Updated Test 2.5 Case 5 and Test 2.7 Case 5 per ECN 003 of NVMe Specification v1.2.1. This ECN alters behavior expected in earlier versions. Now the expected behavior is each case is that the device report status of Successful.

January 24, 2017 (Version 7.0 r02)

David Woolf:

- Changed indication of FYI to Mandatory or In Progress for tests 3.4 Case 2, and 6.2 based on NVMe Interop and Compliance Committee direction.

February 9, 2017 (Version 7.0 r03)

David Woolf:

- Added FYI Test 1.4 Case 10.

© 2026 UNH–IOL

March 22, 2017 (Version 7.0)

David Woolf:

- Final version published to UNH-IOL site ahead of May 2017 NVMe Plugfest #7.

June 17, 2017 (Version 7.0a)

David Woolf:

- Test requirements relaxed in tests 1.1 Case 4, 2.7 Case 9, 2.8 Case 1, 7.1 Case 2, 7.2, and 9.2 Case 3, due to discoveries during May 2017 plugfest. Test Procedure clarified for Test 1.4 Case 7.

August 14, 2017 (Version 8.0)

David Woolf:

- Test requirements relaxed in tests 1.1 Case 4, 2.7 Case 9, 2.8 Case 1, 7.1 Case 2, 7.2, and 9.2 Case 3, due to discoveries during May 2017 plugfest. Test Procedure clarified for Test 1.4 Case 7.
- Clarification in procedure for test 1.2 Case 4.
- Tests 2.3 and 2.4 Case 4 (NLB>MDTS) modified to be 'Mandatory if Supported', as these tests are only applicable if MDTS is not equal to 0.
- Tests 2.3 and 2.4 Case 7 modified to be 'Mandatory if Supported', as these tests are only applicable if NN is not equal to 0xFFFFFFFF.
- Test 1.3 Case 4 modified to be Mandatory if Supported depending on whether MDTS=0 or not. Test modified to expect "Invalid Field in Command" if MDTS conflicts with NUMD/NUMDU/NUMDL.
- Added Test Case 1.3 Case 10.
- Modified Test 2.1 to perform compares using a known sequence rather than the Identify Log Page data to simplify test implementation.
- Clarified Test 2.9 Case 1 that both NABSN and NADSPF need to be set to zero for the test to applicable.
- Clarified Test 2.9 Case 2 that either NABSN and NADSPF need to be set to non-zero values for the test to applicable.
- Modified Test 5.5 to make each condition tested into a separate test case. Most test procedures were not modified and so remain Mandatory, with the exception of case 6 which had the procedure modified and is marked FYI.
- Modified Table in Test 5.5 to clarify which status codes were tested and which were not.
- Modified Appendix D to show abbreviations for test case requirements.
- Modified Test 5.5 Case 5, to indicate that if no features are indicated as 'Not Changeable' then the test case does not apply.
- Moved Abbreviations from Appendix D to a new section near the beginning of the document, named 'Abbreviations'.
- Tests 2.5 Case 5, 2.7 Case 5, 3.4 Case 1, 7.3 Case 1, 1.4 Case 9 had status changed from FYI to Mandatory or Mandatory if Supported.
- Tests 1.1 Case 4, 1.7 Case 1, 2.2 Case 3, 4, 5, 2.7 Case 9, 2.9 Case 2, 7.1 Case 2 and 7.2 had status changed from In Progress to FYI.

September 12, 2017 (Version 8.0a)

David Woolf:

- Updated Appendix C.

October 24, 2017 (Version 8.0b)

David Woolf:

- Updated Appendix D to clarify test requirements designations for NVMe-oF products.

November 14, 2017 (Version 9.0 draft)

David Woolf:

1. Updated Target Specification on cover page to be NVMe v1.3
2. Updated Default Test Setup diagram in Appendix A.
3. Updated "Test 1.3 Case 5" to check the M bit.
4. Fixed error in "Test 2.4 Case 3", where Step 1 was referred to in the Test Procedure rather than Step 2, also corrected the Status Code expected in Observable Result step 2 from 0x0A to 0x80.

© 2026 UNH–IOL

Also updated test to allow for DUT to report status code 0x81, since the status reported would be dependent on which check is performed first, accounting for NVMe v1.3 ECN 002.

5. Fixed error in “Test 2.4 Case 6” where a Read command was indicated instead of a Write command.
6. Fixed error in “Test 2.4 Case 6” where Step 1 was referred to in the Test Procedure rather than Step 2.
7. Fixed error in “Test 2.4 Case 7” where a Read command was indicated instead of a Write command.
8. Fixed error in “Test 2.4 Case 7” where Step 1 was referred to in the Test Procedure rather than Step 2.
9. Fixed error in “Test 2.3 Case 2, 3, 4, 5, 6, 7” where a data pattern was written, this step was not necessary.
10. Updated “Test 4.18” to allow devices to report support for NVMe v1.3. Added check to ensure that VER value from Identify Controller Data structure matches the value in VS.CAP.
11. Updated “Test 2.1 Case 4” to match new requirements in NVMe 1.3 ECN 002, which eliminated the requirement for a DUT to report the lowest numerical value status code first.
12. Updated “Test 2.3 Case 4” to match new requirements in NVMe 1.3 ECN 002, which eliminated the requirement for a DUT to report the lowest numerical value status code first.
13. Updated “Test 2.3 Case 7” to match new requirements in NVMe 1.3 ECN 002, which eliminated the requirement for a DUT to report the lowest numerical value status code first.
14. Updated “Test 2.4 Case 4” to match new requirements in NVMe 1.3 ECN 002, which eliminated the requirement for a DUT to report the lowest numerical value status code first.
15. Updated “Test 2.4 Case 7” to match new requirements in NVMe 1.3 ECN 002, which eliminated the requirement for a DUT to report the lowest numerical value status code first.
16. Updated “Test 2.7 Case 3” to match new requirements in NVMe 1.3 ECN 002, which eliminated the requirement for a DUT to report the lowest numerical value status code first.
17. Updated “Test 2.7 Case 5” to ensure that final READ command returns all 0’s.
18. Updated “Test 2.1 Case 3” to better match the wording in NVMe 1.3 Figure 32.

Removed steps in Test Procedure and Observable Results for Test 2.4 Cases 2-7 where a Read command was used to double check that a Write command had not caused data to be written, since those Read commands themselves could not be performed successfully, and those cases are also checked in Test 2.3

November 28, 2017 (Version 9.0 draft)

David Woolf:

1. Updated Test 1.2 Cases 1-4 Test Procedures and Observable Results to be more complete.
2. Updated Abbreviations section to stop using the ‘Mandatory if Supported’ nomenclature.
3. Updated Test 1.3 Case 4 to be ‘Mandatory’ instead of ‘Mandatory if Supported’. Added a check for MDTS value at the beginning of the test to determine if the test is applicable to the DUT or not.
4. Updated Test 1.6 to be ‘Mandatory’ instead of ‘Mandatory if Supported’. Added a check for OACS field at the beginning of the test to determine if the test is applicable to the DUT or not.
5. Updated Test 2.1 to be ‘Mandatory’ instead of ‘Mandatory if Supported’. Added a check for ONCS field at the beginning of the test to determine if the test is applicable to the DUT or not.
6. Updated Test 2.2 to be ‘Mandatory’ instead of ‘Mandatory if Supported’. Added a check for ONCS field at the beginning of the test to determine if the test is applicable to the DUT or not.
7. Updated Test 2.3 Case 4 to be ‘Mandatory’ instead of ‘Mandatory if Supported’. Added a check for MDTS value at the beginning of the test to determine if the test is applicable to the DUT or not.
8. Updated Test 2.3 Case 7 to be ‘Mandatory’ instead of ‘Mandatory if Supported’. Added a check for NN value at the beginning of the test to determine if the test is applicable to the DUT or not.
9. Updated Test 2.3 Case 6 to include a check for NN value at the beginning of the test to determine if the test is applicable to the DUT or not.
10. Updated Observable Results of Test 2.3 Case 6 to check for status code ’Invalid Namespace or Format (0BH).
11. Updated Test 2.4 Case 4 to be ‘Mandatory’ instead of ‘Mandatory if Supported’. Added a check for MDTS value at the beginning of the test to determine if the test is applicable to the DUT or not.
12. Updated Test 2.4 Case 7 to be ‘Mandatory’ instead of ‘Mandatory if Supported’. Added a check for NN value at the beginning of the test to determine if the test is applicable to the DUT or not.
13. Updated Test 2.5 to be ‘Mandatory’ instead of ‘Mandatory if Supported’. Added a check for ONCS field at the beginning of the test to determine if the test is applicable to the DUT or not. Added checks for MDTS and NN values where appropriate.

© 2026 UNH–IOL

14. Updated Test 2.7 to be ‘Mandatory’ instead of ‘Mandatory if Supported’. Added a check for ONCS field at the beginning of the test to determine if the test is applicable to the DUT or not. Added checks for MDTS and NN values where appropriate.
15. Updated Test 2.7 to be ‘Mandatory’ instead of ‘Mandatory if Supported’. Added a check for NABSN field at the beginning of the test to determine if the test is applicable to the DUT or not.
16. Updated Test 3.1 to be ‘Mandatory’ instead of ‘Mandatory if Supported’. Added a check for Metadata Size field at the beginning of the test to determine if the test is applicable to the DUT or not.
17. Updated Test 3.2 to be ‘Mandatory’ instead of ‘Mandatory if Supported’. Added a check for DPC field at the beginning of the test to determine if the test is applicable to the DUT or not.
18. Updated Test 3.4 to be ‘Mandatory’ instead of ‘Mandatory if Supported’. Added a check for HMPRE field at the beginning of the test to determine if the test is applicable to the DUT or not.
19. Updated Test 4.13 to be ‘Mandatory’ instead of ‘Mandatory if Supported’.
20. Updated Test 5.5 Case 3 to be ‘Mandatory’ instead of ‘Mandatory if Supported’.
21. Updated Test 6.4 to be ‘Mandatory’ instead of ‘Mandatory if Supported’.
22. Updated all tests in Group 7 to be ‘Mandatory’ instead of ‘Mandatory if Supported’. Added a check in all tests of the ONCS field to determine of the controller supports reservations.
23. Updated all tests in Group 9 to be ‘Mandatory’ instead of ‘Mandatory if Supported’. Added a check in all tests of the OACS field to determine of the controller supports Namespace Management.
24. Updated Appendix D to remove all references to ‘Mandatory if Supported’.
25. Updated Table in Test 1.2 to contain new Features: Timestamp, Keep Alive Timer, Host Controller Thermal Management, Non-Operational Power State Config.
26. Updated Test 2.4 Cases 8, 9, 10 to remove redundant READ operation in Test Procedure.

December 4, 2017 (Version 9.0 draft)

David Woolf:

1. Updated Test 2.5 Case 5 to include more than 1 READ command to be performed after the Write Uncorrectable command in order to ensure that all LBAs affected by the Write Uncorrectable command are affected.
2. Updated Test 2.7 Case 5 to include more than 1 READ command to be performed after the Write Zeroes command in order to ensure that all LBAs affected by the Write Zeroes command are affected.

December 6, 2017 (Version 9.0 draft)

David Woolf:

1. Updated Test 1.6 Case 3 to include a check of FNA Bit 2, to determine if the test case is applicable or not.
2. Updated Test 1.1 Case 5, to show that CNS reserved value testing should check only FFh.

December 12, 2017 (Version 9.0 draft)

David Woolf:

1. Updated Test 1.3 Case 3 to only check LID 7Fh.
2. Updated Test 1.2 Case 5 to only check SEL 111b.
3. Updated Test 2.2 Case 3, Case 4, and Case 5 to be Mandatory for NVMe Drives.
4. Updated Test 1.4 Case 10 to be Mandatory for NVMe Drives.
5. Updated Test 1.7 Case 1 to be Mandatory for NVMe Drives.
6. Updated Test 2.3 Case 11 to be Mandatory for NVMe Drives.
7. Updated Test 2.4 Case 11 to be Mandatory for NVMe Drives.
8. Updated Test 2.7 Case 9 to be Mandatory for NVMe Drives.
9. Updated Test 2.9 Case 2 to be Mandatory for NVMe Drives.
10. Updated Test 5.5 Case 6 to be Mandatory for NVMe Drives.

December 14, 2017 (Version 9.0 draft)

David Woolf:

1. Added Test 1.1 Case 6.
2. Added Test 1.10 Cases 1-8
3. Added Test 1.11 Cases 1-8

© 2026 UNH–IOL

4. Added Test 1.12 Cases 1-5
5. Added Test 1.13 Cases 1-4

December 21, 2017 (Version 9.0 draft)

David Woolf:

1. Updated Observable Results in Test 3.4 Case 2 to eliminate items that can notcannot be observed. Changed the test status to FYI.
2. Updated Observable Results in Test 3.4 Case 3 to eliminate items that cannot be observed. Changed the test status to FYI.

January 4, 2018 (Version 9.0 draft)

David Woolf:

1. Updated all references in Group 9 from 8.11 to 8.12 to match NVMe v1.3 specification.

January 22, 2018 (Version 9.0 draft)

Colin Dorsey:

1. Added tests 8.4, 8.5, 8.6, 8.7, 8.8. Updated test procedures for test 8.2

January 23, 2018 (Version 9.0 draft)

David Woolf:

1. Updated following tests from OF-FYI to OF to indicate that they are now mandatory for NVMe-oF products, per ICC decision: 1.1 Cases 1-2, 1.3 Cases 1-2, 1.7 Case 2, 2.5 all cases, 2.6 Case 2, 2.8, 3.3 Cases 1-4, 4.1, 4.2, 4.3, 4.4, 4.6, 4.7, 4.9, 4.10, 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17, 4.18, 5.1.

February 1, 2018 (Version 9.0 draft)

David Woolf:

1. Updated test 4.13 test procedure for clarity.
2. Updated test 3.4 Case 3 test procedure for clarity and completeness.

February 5, 2018 (Version 9.0 draft)

David Woolf:

1. Modified Case 6 to Test 1.1 to cover Namespace Identification Descriptors.
2. Added Test 1.3 Cases 11, 12, 13, 14, 15 to address Telemetry Log Pages.

March 15, 2018 (Version 9.0 draft)

David Woolf:

1. Modified Case 3 of Test 1.2 for clarity.
2. Modified Test 7.1 Case 2 and Test 7.2 Case 3 to include the step of performing a Reservation Acquire command.
3. Modified Test 1.2 Case 4, there was a typo indicated DWord 10 rather than DWord 0.

May 21, 2018 (Version 10.0 draft)

David Woolf:

1. Corrected typo in Test 2.2 Case 2

August 30, 2018 (Version 10.0 Release)

David Woolf:

1. 10.0 Release

November 20, 2018 (Version 11.0 draft)

David Woolf:

1. Modified Test 1.1 Case 4 as previous version of test case was invalid.
2. Renumbered Test 1.1. Case 5 as Test 1.1 Case 13.
3. Added new Test 1.1 Case 5 (FYI).
4. Added new Test 1.1 Case 6 (FYI).
5. Added new Test 1.1 Case 7 (FYI).

© 2026 UNH–IOL

6. Added new Test 1.1 Case 8 (FYI).
7. Added new Test 1.1 Case 9 (FYI).
8. Added new Test 1.1 Case 10 (FYI).
9. Added new Test 1.1 Case 11 (FYI).
10. Added new Test 1.1 Case 12 (FYI).
11. Added step to Test 1.2 Case 5 to check ONCS Bit 4 to determine if test is applicable before proceeding with test procedure.
12. Updated Test 1.2 Cases 1-4 to check if FIDs are changeable before proceeding with test procedure.
13. Updated Test 1.3 Case 2 to allow for the case where there are no unsupported LIDs because the DUT supports all Vendor Specific LIDs.
14. Expanded Observable Results for Test 1.3 Case 11 (FYI).
15. Added new Test 1.3 Case 16.
16. Updated Test 1.3 Case 3 to use a different value for the reserved value, now uses 0x6F instead of 0x7F.
17. Fixed typo in Test 1.3 Case 4 where the test procedure referenced steps 2 and 3 when it should reference steps 3 and 4.
18. In Test 1.3 Case 4, the reference to the list of mandatory LIDs in the specification was changed to be a list of the mandatory LIDs, instead of a reference.
19. Updated Test 1.3 Case 7 to perform 1000 Read/Compare operations of LBADS, 2x LBADS, and 4x LBADS to ensure that the DUT is following the requirements on recording Data Units Read in 1000 512 Byte units, rather than just recording the number of Read operations, as some devices have been observed to do. Added a step to record the value for LBADS.
20. Updated Test 1.3 Case 8 to perform 1000 Write operations of LBADS, 2x LBADS, and 4x LBADS to ensure that the DUT is following the requirements on recording Data Units Written in 1000 512 Byte units, rather than just recording the number of Write operations, as some devices have been observed to do. Remove the Write Uncorrectable operation from this test. Added a step to record the value for LBADS.
21. Added new Test 1.3 Case 17.
22. Updated Test 1.7 Case to expect a status of 07h Command Abort Requested if a completion queue entry is posted. If no completion queue entry is posted the test is not applicable.
23. Updated Test 1.9 to clarify that the Set Feature command in Step 2b must be performed with the Save bit set to 1 (SV=1).
24. Updated first 2 steps in Test 1.10 Case 1 to clarify how to ensure that no Device Self Test command is currently in progress.
25. Updated first 2 steps in Test 1.10 Case 2 to clarify how to ensure that no Device Self Test command is currently in progress.
26. Updated first 2 steps in Test 1.10 Case 3 to clarify how to ensure that no Device Self Test command is currently in progress.
27. Updated first 2 steps in Test 1.10 Case 4 to clarify how to ensure that no Device Self Test command is currently in progress.
28. Updated first 2 steps in Test 1.10 Case 5 to clarify how to ensure that no Device Self Test command is currently in progress.
29. Updated first 2 steps in Test 1.10 Case 6 to clarify how to ensure that no Device Self Test command is currently in progress.
30. Updated first 2 steps in Test 1.10 Case 7 to clarify how to ensure that no Device Self Test command is currently in progress.
31. Updated first 2 steps in Test 1.10 Case 8 to clarify how to ensure that no Device Self Test command is currently in progress.
32. Updated first 2 steps in Test 1.11 Case 1 to clarify how to ensure that no Device Self Test command is currently in progress.
33. Updated first 2 steps in Test 1.11 Case 2 to clarify how to ensure that no Device Self Test command is currently in progress.
34. Updated first 2 steps in Test 1.11 Case 3 to clarify how to ensure that no Device Self Test command is currently in progress.
35. Updated first 2 steps in Test 1.11 Case 4 to clarify how to ensure that no Device Self Test command is currently in progress.

© 2026 UNH–IOL

36. Updated first 2 steps in Test 1.11 Case 5 to clarify how to ensure that no Device Self Test command is currently in progress.
37. Updated first 2 steps in Test 1.11 Case 6 to clarify how to ensure that no Device Self Test command is currently in progress.
38. Updated first 2 steps in Test 1.11 Case 7 to clarify how to ensure that no Device Self Test command is currently in progress.
39. Updated first 2 steps in Test 1.11 Case 8 to clarify how to ensure that no Device Self Test command is currently in progress.
40. Updated first 2 steps in Test 1.12 Case 1 to clarify how to ensure that no Device Self Test command is currently in progress.
41. Updated first 2 steps in Test 1.12 Case 2 to clarify how to ensure that no Device Self Test command is currently in progress.
42. Updated first 2 steps in Test 1.12 Case 3 to clarify how to ensure that no Device Self Test command is currently in progress.
43. Updated first 2 steps in Test 1.12 Case 4 to clarify how to ensure that no Device Self Test command is currently in progress.
44. Updated first 2 steps in Test 1.12 Case 5 to clarify how to ensure that no Device Self Test command is currently in progress.
45. Updated first 2 steps in Test 1.13 Case 1 to clarify how to ensure that no Device Self Test command is currently in progress.
46. Updated first 2 steps in Test 1.13 Case 2 to clarify how to ensure that no Device Self Test command is currently in progress.
47. Updated first 2 steps in Test 1.13 Case 3 to clarify how to ensure that no Device Self Test command is currently in progress.
48. Updated first 2 steps in Test 1.13 Case 4 to clarify how to ensure that no Device Self Test command is currently in progress.
49. Updated Test 2.7 Case 9 to allow for an error code of 81h Invalid Protection Information, as well as 02h Invalid Field in Command.
50. Updated Test 1.10 Case 4 Observable Results step 1 to be the following: Verify that the Device Self-test command returns status Invalid Namespace or Format in Command.
51. Updated Test 1.11 Case 4 Observable Results step 1 to be the following: Verify that the Device Self-test command returns status Invalid Namespace or Format in Command.
52. Updated Test 1.10 Case 6 test procedure to account for the command completion for the DST command to be sent before the DST operation completes.
53. Updated Test 1.10 Case 7 test procedure to account for the command completion for the DST command to be sent before the DST operation completes.
54. Updated Test 1.10 Case 8 test procedure to account for the command completion for the DST command to be sent before the DST operation completes.
55. Updated Test 1.11 Case 6 test procedure to account for the command completion for the DST command to be sent before the DST operation completes.
56. Updated Test 1.11 Case 7 test procedure to account for the command completion for the DST command to be sent before the DST operation completes.
57. Updated Test 1.11 Case 8 test procedure to account for the command completion for the DST command to be sent before the DST operation completes.
58. Updated test procedure for test 4.12 to delete all existing I/O Submission and Completion queues prior to performing test.
59. Updated test procedure for test 5.5 Case 5 to ensure that when attempting to change a feature value, the value offered is different than the current value.
60. Added step to Test 8.8 Case 1, to check the HCTMA bit before proceeding with the test.
61. Added step to Test 8.8 Case 2, to check the HCTMA bit before proceeding with the test.
62. Added step to Test 7.1 Case 1 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
63. Added step to Test 7.1 Case 2 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
64. Added step to Test 7.2 Case 1 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

© 2026 UNH–IOL

65. Added step to Test 7.2 Case 2 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
66. Added step to Test 7.2 Case 3 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
67. Added step to Test 7.2 Case 4 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
68. Added step to Test 7.3 Case 1 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
69. Added step to Test 7.3 Case 2 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
70. Added step to Test 7.4 Case 1 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
71. Added step to Test 7.4 Case 2 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
72. Added step to Test 7.4 Case 3 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
73. Added step to Test 7.5 Case 1 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
74. Added step to Test 7.5 Case 2 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
75. Added step to Test 7.5 Case 3 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
76. Added step to Test 7.5 Case 4 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
77. Added step to Test 7.6 Case 1 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
78. Added step to Test 7.6 Case 2 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
79. Added step to Test 7.6 Case 3 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
80. Added step to Test 7.6 Case 4 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
81. Added step to Test 7.7 Case 1 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
82. Added step to Test 7.7 Case 2 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
83. Added step to Test 7.8 Case 1 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
84. Added step to Test 7.8 Case 2 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
85. Added step to Test 7.8 Case 3 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
86. Added step to Test 7.8 Case 4 to perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.
87. Added step to Test 7.6 Case 3 to ensure that the RTYPE field is set to either WRITE EXCLUSIVE ALL REGISTRANTS or EXCLUSIVE ACCESS ALL REGISTRANTS.
88. Updated Test 7.4 Case 2 to align with the updated use of the IEKEY parameter in NVMe v1.3 specification. Expectation is that when the IEKEY parameter is set to 1 in a Reservation Acquire command, that the DUT will return status of “Invalid Field in Command”, and that no reservation is acquired.
89. Added new Test 1.2 Case 6.
90. Added new Test 2.2 Case 6.
91. Updated Test 9.2.1 to check the FNA bit, to see whether the DUT supports Format NVM on a per namespace basis, or if all namespaces within the controller must use the same LBAF.
92. Updated Test 9.2.2 to ensure that a namespace exists before attempting to delete a namespace.
93. New test case for Directives Receive, Test 1.15 Case 1.
94. New test case for Directives Receive, Test 1.15 Case 2.

© 2026 UNH–IOL

95. New test case for Directives Send, Test 1.16 Case 1.
96. New test case for Directives Send, Test 1.16 Case 2.
97. New test case for Directives Send, Test 1.16 Case 3.
98. Test 1.1 Case 3 Identify Namespace List change from M, OF-FYI to M, OF
99. Test 1.1 Case 5 Identify to Reserved CNS change from M, OF-IP to M, OF-FYI
100. Test 1.3 Case 3 Get Log Page to Reserved LID change from M, OF-FYI to M, OF
101. Test 1.3 Case 4 Get Log Page with NUMD/MDTS Conflict change from M, OF-FYI to M, OF
102. Test 1.7 Case 4 Asynchronous Masking Event change from M, OF-FYI to M, OF
103. Test 2.2 Case 1 Dataset Management change from M, OF-FYI to M, OF
104. Test 2.2 Case 2 Dataset Management Deallocate change from M, OF-FYI to M, OF
105. Test 2.2 Case 5 Dataset Management Deallocate Correct Range change from M, OF-FYI to M, OF
106. Test 2.3 Case 1 Read change from M, OF-FYI to M, OF
107. Test 2.3 Case 3 Read with NLB Out of Range change from M, OF-FYI to M, OF
108. Test 2.3 Case 6 Read with Invalid NSID change from M, OF-FYI to M, OF
109. Test 2.3 Case 7 Read with Invalid NSID and SLBA Out of Range change from M, OF-FYI to M, OF
110. Test 2.3 Case 8 Read with LR=0, FUA=1 change from M, OF-FYI to M, OF
111. Test 2.3 Case 9 Read with LR=1, FUA=0 change from M, OF-FYI to M, OF
112. Test 2.3 Case 10 Read with LR=1, FUA=1 change from M, OF-FYI to M, OF
113. Test 2.4 Case 1 Write change from M, OF-FYI to M, OF
114. Test 2.4 Case 8 Write with LR=0, FUA=1 change from M, OF-FYI to M, OF
115. Test 2.4 Case 10 Write with LR=1, FUA=1 change from M, OF-FYI to M, OF
116. Test 2.6 Case 1 Flush change from M, OF-FYI to M, OF
117. Test 3.3 Case 5 Power Management Feature change from M, OF-FYI to M, OF
118. Test 4.5 CAP.TO change from M, OF-FYI to M, OF
119. Test 4.8 CAP.MQES change from M, OF-FYI to M, OF
120. Test 5.3 Status Field change from M, OF-FYI to M, OF
121. Test 6.1 Conventional Reset change from IP to FYI
122. Test 6.3 Controller Reset change from M, OF-IP to M, OF-FYI
123. Test 9.2 Case 3 Create Namespace when Insufficient Capacity change from IP to FYI

April 2, 2019 (Version 12.0 draft)

David Woolf:

1. Update Test 1.1 Case 4, to include a step where *[active]* namespaces are determined before sending the Identify to each active namespace.
2. Update Test 1.2 Case 4, Observable Result Step 2, to clarify that “bit 0 of Dword 0” is that of the Get Feature command performed in Procedure Step 3a.
3. Update Test 1.2 Case 4, Observable Results Step 4, to accommodate the special case of the Reservation Notification Mask feature, FID 82h. The response to the Get Feature command in step d of the procedure may be “Invalid Namespace” or “Invalid Field in Command” for the Feature ID 82h only, per 5.21.1.20. For all other FIDs the expected response is “Invalid Namespace”
4. Update Test 1.3 Case 2 to check for proper handling of a vendor-specific log page by checking a single known unsupported vendor-specific LID, rather than trying to check the entire range (C0h-FFh) of vendor-specific LIDs.
5. Update Test 1.3 Case 7 to check for support of Compare command before performing compare command. If compare not supported, the test should be not applicable.
6. Update Test 1.3 Case 17, to have a write operation at the end of the test so that the drive remains useable, since the previous operation was a Write Uncorrectable, which will render the drive unreadable.
7. Updated Observable Results in Test 1.5 for clarity.
8. Update Test 1.9 Step 2b, to ensure that rather than checking each FID, check only the FIDs that are both ‘Savable’ and ‘Changeable’
9. Update Test 1.16 Case 2, as the sub case is missing Observable Results. The expected Observable Results should be identical to Case 1, as although the procedure is different, using a different

© 2026 UNH–IOL

DTYPE value, the end results is that all the commands should complete successfully if Directives are supported.

10. Update Test 2.6.2 to fix typo in Test Procedure numbering.
11. Update Test 3.1 to include support for Fabrics implementations. This requires flagging this test as 'OF-FYI', and tracking the implementations of the test tools.
12. Test. 4.10 – Modify test to check required and maximum values for IOCQES from Identify Controller Data Structure. Verify that maximum value is greater than or equal to minimum value. Verify that IOCQES value in CC register falls within the specified range. Remove checking of CQES as this is not visible from test application.
13. Test 4.11 - Modify test to check required and maximum values for IOSQES from Identify Controller Data Structure. Verify that maximum value is greater than or equal to minimum value. Verify that IOSQES value in CC register falls within the specified range. Remove checking of SQES as this is not visible from test application.
14. Test 5.2 – Modify test to have only one SQ open at a time, instead of having 2 open SQs. Open a SQ and CQ, run a command through those queues and verify that SQID was used correctly. Close the SQ. Open a new SQ with a new SQID. Associate the new SQ with the previous CQ. Run a command through the SQ and CQ, verify that SQID was used properly.
15. Update Tests 5.4 to include support for Fabrics implementations. This requires flagging this test as 'OF-FYI', and tracking the implementations of the test tools.
16. Update Tests 5.5 to include support for Fabrics implementations. This requires flagging this test as 'OF-FYI', and tracking the implementations of the test tools.
17. Update Tests 5.6 to include support for Fabrics implementations. This requires flagging this test as 'OF-FYI', and tracking the implementations of the test tools.
18. Update typo in reference in Test 5.6 to be 4.6.1.2.3 rather than 4.5.1.2.3.
19. Test 6.4 – Modify test to check that drive is removed from host system after NSSR occurs. Then check to see that drive reappears in the host system after a rescan is performed. Remove the checking of the CSTS.NSSRO register as this is in the domain of the host system and not observable from test application.
20. Update test 9.2 Case 1, to have the procedure ensure that any created namespaces are created with the same format as all existing namespaces on the device, if that is required by the device, as indicated by setting FNA to 1. Remove the step where the entire device is formatted.
21. Updated Test 1.4 Case 9 to reflect changes due to NVMe v1.3 ECN 001.
22. Updated Test 1.4 Case 10 to reflect changes due to NVMe v1.3 ECN 001.
23. Updated Test 2.3 Case 11 to reflect changes due to NVMe v1.3 ECN 001.
24. Updated Test 2.7 Case 9, to check for support of End to End Data Protection before proceeding with the test.
25. Updated tests in Group 7 to be FYI for Fabrics products. All dual-port tests in Group 7 made FYI.
26. Updated Observable Results in Test 1.1. Case 1.
27. Updated Observable Results in Test 1.1. Case 4.
28. Added new FYI Test 1.12 Case 6
29. Added new FYI Test 1.13 Case 5
30. Added new FYI Test 1.17 Cases 1 and 2
31. Added new FYI Test 1.2 Case 7.
32. Added new FYI Test 1.2 Case 8.
33. Added new FYI Test 1.3 Case 19.
34. Added new FYI Test 1.6 Case 8.
35. Added new FYI Test 1.6 Case 9.
36. Added new FYI Test 1.6 Case 10.
37. Added new FYI Test 1.6 Case 11.
38. Added new FYI Test 1.6 Case 12.
39. Added new FYI Test 2.1 Case 7.
40. Added new FYI Test 2.6 Case 3.
41. Added new FYI Test 2.11 Case 1.
42. Added new FYI Test 2.11 Case 2.
43. Added new FYI Test 3.4 Case 4.
44. Added new FYI Test 3.4 Case 5.

© 2026 UNH–IOL

45. Updated Test Procedure for IP Test 3.5.
46. Added new FYI Tests 3.6 Cases 1 and 2.
47. Added new FYI Tests 3.7 Case 1.
48. Added new FYI Test 7.1 Case 3.
49. Added new FYI Test 7.1 Case 4.
50. Updated Observable Results in FYI Test 7.5 Case 2.
51. Added new FYI Test 7.9 Case 1.
52. Added new FYI Test 7.9 Case 2.
53. Added new FYI Test 7.9 Case 3.
54. Added new FYI Test 9.1 Case 5.
55. Added new FYI Test 9.2 Case 4.
56. Added check for NS Management Support in Test 1.10 Case 5 and 1.11 Case 5.

March 9, 2020 (Version 13.0)

David Woolf:

1. References to Figures and Tables in the NVMe specification which appeared outside of the references section of each description were removed.
2. Test 1.1 Case 4 status updated from (FYI, OF-FYI) to (M, OF)
3. Test 1.1 Case 5, 6, 8, 9, 10 status updated from (FYI, OF-FYI) to (M, OF-FYI).
4. Test 1.2 Case 1 status updated from (M, OF-FYI) to (M, OF).
5. Test 1.2 Case 6 status updated from (FYI, OF-FYI) to (M, OF-FYI).
6. Test 1.3 Case 5 status updated from (FYI, OF-FYI) to (M, OF-FYI).
7. Test 1.10 Case 1, 2, 3 status updated from (FYI, OF-FYI) to (M, OF-FYI).
8. Test 1.11 Case 1, 2, 3 status updated from (FYI, OF-FYI) to (M, OF-FYI).
9. Test 1.12 Case 2, 3, 4, 5 status updated from (FYI, OF-FYI) to (M, OF-FYI).
10. Test 1.13 Case 2, 3, 4 status updated from (FYI, OF-FYI) to (M, OF-FYI).
11. Test 2.2 Case 6 status updated from (FYI, OF-FYI) to (M, OF).
12. Test 2.3 Case 2 3, 5, 6, 7 status updated from (M, OF-FYI) to (M, OF).
13. Test 2.4 Case 2, 3, 5, 6, 7 status updated from (M, OF-FYI) to (M, OF).
14. Test 6.1 status updated from (FYI) to (M).
15. Tests 8.4, 8.5 status updated from (FYI) to (M).
16. 'Target Specification' for Conformance Test Plan has been updated to be NVMe v1.4.
17. Test 1.2.4, updated Step 2a of Observable Results to check for a status of 'Feature Not Saveable' to be returned instead of 'Invalid field in Command' if the feature is not saveable as indicated by Dword 0 bit 0 of the Completion Entry for the Get Feature request for that feature ID. Removed duplicate steps. Returned to FYI status due to significant test changes.
18. Test 1.3.3, updated test case to allow for different Error Codes being reported depending on the specification version supported.
19. Test 1.3.4, added a statement to end of step 4, which says, "If NUMD cannot be set to a value greater than MDTS, or if MDTS =0, then this test is not applicable".
20. Test 1.3.19, updated test description to indicate Data Units Read should increase by 7x LBADS/512 rather than 14x LBADS/512. Currently the UNH-IOL test implementation correctly checks for 7x, just the description needed to be fixed.
21. Test 1.2 Case 8, updated the 2nd step of the Observable Results to have the second command say "Get Features" rather than "Set Features".
22. Updated Test 1.3 Case 6 to make sure that if a device claims support for NVMe v1.4 or higher, that the Get Log Page - SMART / Health Status – Critical Warning Bit 1 is set to 1 only when a temperature is greater than or equal to an over temperature threshold; or less than or equal to an under temperature threshold. See NVMe revision 1.4 section 5.2 & 5.14
23. Modified Test Case 1.5 (now case 1.5.1) to address modifications to the Abort Command, specifically that if the command to abort was successfully aborted, then a completion queue entry for the aborted command shall be posted to the appropriate Admin or I/O Completion Queue with a status of Command Abort Requested before the completion queue entry for the Abort command is posted to the Admin Completion Queue. See NVMe v1.4 section 5.1.

© 2026 UNH–IOL

24. For all cases, 1-12, in Test 1.6, steps were added to record DPS settings at the beginning of the test, then when the test is complete, perform another Format NVM operation to restore the DPS settings recorded at the beginning of the test.
25. Test 1.6 Case 5 and 6, clarified that if the DUT supports all 16 LBAF, then the test is not applicable, as all valid LBAF values are supported, and the host cannot request an unsupported value. Renamed both cases with "Unsupported LBAF" instead of "Invalid LBAF". Added a step to the test procedure to check if all 16 LBAF are supported, and if yes, skip the test.
26. Test 1.6 Case 7, check DPC field for support for End to End Data Protection before starting the test. Observable results depend on the DPC values.
27. Test 1.6 Case 7, in Step 6, procedure clarified to say '…LBAF with *non-zero metadata size* supported by the DUT…'
28. Test 1.6 Case 8, in Step 4, text should say '…LBAF with *non-zero metadata size* supported by the DUT…'
29. Test 1.6 Case 8, updated to allow either PIL=1 or PIL=0, depending on what is supported by the DUT, as indicated in the DPC field. Removed the PIL from the title of the test case.
30. Removed Test 1.6 Case 9 NSID=FFFFFFFFh, no namespaces, SES=000, since it is an invalid test case.
31. Test 1.6 Case 10 NSID=FFFFFFFFh, no attached namespaces, SES=000, renamed to Test 1.6 Case 9 NSID=FFFFFFFFh, no attached namespaces, SES=000. Added an initial step where a check of bit 0 of FNA = 1, is performed. If bit 0 of FNA is not 1, the test is not applicable. Also added steps where the namespace is detached after the READ operation in Step 5. In Step 6, removed sentence saying bit 0 of FNA bit should be 1. In Step 6, the namespace should be reattached.
32. Removed Test 1.6 Case 11 NSID=FFFFFFFFh, no namespaces, SES≠000, since it is an invalid test case.
33. Test 1.6 Case 12 NSID=FFFFFFFFh, no attached namespaces, SES≠000, renamed to Test 1.6 Case 10 NSID=FFFFFFFFh, no attached namespaces, SES≠000. Added an initial step where a check of bit 1 of FNA = 1, is performed. If bit 1 of FNA is not 1, the test is not applicable. Also, added steps where the namespace is detached after the READ operation in Step 4. In Step 6, removed sentence saying bit 1 of FNA bit should be 1. In Step 6, the namespace should be reattached.
34. Test Case 1.17.1 added that a Get Log Page for the Sanitize status Log Page should be performed after each Sanitize operation, and added an Observable Result, to check that the Sanitize Status Log page is updated on successful completion of the SANITIZE command.
35. Test Case 1.17.2 updated so that the test is still performed if the SANICAP field is non-zero. Instead of skipping the test if the SANICAP field is non-zero, the test is performed for each SANITIZE operations, to check that the DUT responds with 'Invalid Field in Command' for operations indicated as not supported in the SANICAP field.
36. Test 2.1 Case 7, step 3 updated to have PRACT set to 1 instead of set to 0. This is a change to the documentation only, as the UNH-IOL test implementation already does this correctly. Also, in Step 4, the reference to Step 1 should instead be a reference to Step 3.
37. Test 3.6 Case 1 and 2, added a step to the test procedure to ensure that Predictable Latency Mode is enabled with a Set Feature command prior to checking the associated Log Pages.
38. Obsoleted and removed Tests 1.4 Case 9, Test 1.4 Case 10, Test 2.3 Case 11, and Test 2.4 Case 11, since there is no requirement to report PRP Offset Invalid errors. This was discovered when checking for compliance issues with NVMe v1.4 ECN 002.
39. In test 7.1.4, added a check of the CTRATT field to determine if the controller supports 128 bit Host identifiers, if not the test is not applicable.
40. Updated Test 7.5.2 which previously stated that step RES_REL4 should issue a Reservation Release command with an invalid CRKEY value, and IEKEY bit set to 0, and RES_REL3 should also use an invalid CRKEY value but with IEKEY bit set to 1. In doing so, RES_REL4 was written to do the same thing as RES_REL1 (invalid CRKEY, and IEKEY=0). RES_REL3 and RES_REL4 should actually use the correct CRKEY value. Therefore the documented procedure has been changed in steps RES_REL3 and RES_REL4 from "supplying any reservation key other than the current reservation key associated with the host in the CRKEY field" to "supplying the correct reservation key that is associated with the host in the CRKEY field". This correction has already been made in the UNH-IOL test implementation.
41. In test 7.9.3, updated test case to cycle 1000 times, rather than wrapping the counter. The reason for this is that wrapping the counter would take a prohibitively long time.
42. Modified Test 9.1.5 to just check for NSID uniqueness, by checking all controllers in the subsystems for their NSIDs. If all NSIDs are unique, test passes. If not all NSIDs are unique, check that

© 2026 UNH–IOL

the not-unique NSIDs are in fact shared namespaces. This can be done by writing a pattern to the namespace through one controller, and then checking that that pattern can be read through all other controller that are sharing that namespace.

43. Modifications to test case 1.3.18 for the Persistent Event log. First, support for PEL will be checked. Next, the PEL will be requested via Get Log command. Next an event will be caused that will cause an update to the PEL. The PEL will be requested again to see that the event was logged. To check persistence, a Controller Level Reset is performed. At this time persistence across power cycles will not be checked. After the reset completes, the PEL will be checked again and see that the recorded events were still present in the log.
44. New Test Case, 1.3.20, was added to specifically check Data Units Read count for the READ operation only. This was necessary because Test 1.3.7 includes both Read and Compare operations, but is skipped if the DUT doesn't support Compare, therefore the Data Units Read functionality was not being checked for READ operations. Test 1.3.7 properly checks Data Units Read for the Compare Command but does not check the READ command.
45. New Test Case, 1.5.2, to check proper operation of the Abort command when an abort is requested for an Admin Command (previously only IO commands were checked).
46. In Test 1.4 Case 7, some test items were broken out into separate test cases creating a completely new Test 1.4 Case 9 and Test 1.4 Case 10.
47. In Test 1.4 Case 7, some test items were broken out into separate test cases also creating a completely new Test 1.4 Case 10.
48. New Test 2.12 for Fused Operations with several sub tests.
49. New Test Case 7.1.5 to check proper use of the Controller ID field for a dynamic controller.
50. New test 9.1.6 to check properly handling of NSID FFFFFFFFh when Namespace Management is not supported.
51. New Test 3.8.1 for PMR.
52. New Test 3.8.2 for PMR.
53. New Test 3.9.1 for Get LBA Status.
54. New Test 3.10,1, 2, 3, 4, for Improved Performance Parameters.
55. Modified Test 1.2.7.
56. New Test 1.2.9
57. New Test 1.2.10
58. New Test 3.11.1
59. New Test 3.12.1 and 2
60. New Test 3.13.1 and 2
61. New Test 3.14.1 and 2
62. New Test 1.1.14.
63. New Test 3.15.1 and 2.
64. New Test 1.2.11.
65. New Test 1.4.11.
66. New Test 1.3.22.
67. New Test 1.3.23.
68. Updated Test 1.10 to ensure that that all DST tests properly allow the DST operation to run and complete in the background even after a completion is issued.
69. Updated Test 1.11 to ensure that that all DST tests properly allow the DST operation to run and complete in the background even after a completion is issued.
70. Updated Test 1.12to ensure that that all DST tests properly allow the DST operation to run and complete in the background even after a completion is issued.
71. Updated Test 1.13 to ensure that that all DST tests properly allow the DST operation to run and complete in the background even after a completion is issued.
72. New Observable Results for the CNTRLTYPE field in Test 1.1 Case 2.
73. New Test 3.16.1.
74. New Test 3.17.1.
75. New Test Case 2.1.8
76. New Test Case 2.2.17
77. New Test Case 2.3.11

© 2026 UNH–IOL

78. New Test Case 2.4.11
79. New Test Case 2.5.6
80. New Test Case 2.7.10
81. New Test Case 2.11.3
82. New Test Case 1.2.12
83. Updated tests 1.3.11-16 to include status OF-FYI.
84. New Test Cases 1.18.1
85. New Test Cases 1.18.2
86. New Test Cases 1.18.3
87. New Test Case 1.16.4
88. New Test Case 1.16.5
89. New Test Case 1.16.6
90. New Test Case 7.2.5
91. Modifications to Observable Results in test 1.3 Case 11.
92. Modified procedure in Test 1.1. Case 1 to match the current UNH-IOL test implementation.
93. Corrected Test 1.1 Case 4 Observable Results
94. Corrected Test 1.1 Case 6 Observable Results
95. Corrected Test 1.1 Case 7 Observable Results
96. Corrected Test 1.1 Case 12 Observable Results
97. Fixed Typo in Test 6.4 test procedure.
98. Clarification to test procedure in Test 9.2.1, which does not require update to current test implementation.
99. Updated Test 7.1 Case 3 and 4 to check that the Host Identifier Feature is supported as required.
100. Updated Test 7.1 Case 1 and 2 to check that the Host Identifier Feature is supported as required.
101. Fixed typo with wrong CNS value in Test 1.1.1, was CNS=03h, should be CNS-02h for Active Namespace ID list.

July 21, 2020 (Version 14.0)

David Woolf:

1. Updated reference to NVMe-MI specification.
2. Identify Command Tests 1.1.4, 1.1.5 status changed from FYI to Mandatory
3. Get/Set Features Tests 1.2.4, 1.2.5 status changed from FYI to Mandatory
4. Fixed issue with observables in Test 1.2.7
5. Test 1.3.5 status changed from FYI to Mandatory
6. Updated Observable results in Test 1.6 Case 6.
7. Device Self Test Tests 1.10.1-7 status changed from FYI to Mandatory
8. Device Self Test Extended Tests 1.11.1-4, 6-8 status changed from FYI to Mandatory
9. Abort Device Self Test Operation Tests 1.12.2-5 status changed from FYI to Mandatory
10. Abort Extended DST 1.13.2-4 Test status changed from FYI to Mandatory
11. Dataset Management Tests 2.2.1-6 status changed from FYI to Mandatory
12. Write Uncorrectable Test 2.5.5 status changed from FYI to Mandatory
13. Write Zeroes Test 2.7.5 status changed from FYI to Mandatory
14. Reset Test 6.1 status changed from FYI to Mandatory
15. Reservations Tests 7.1.2, 7.2.1, 7.2.2, 7.2.3, 7.3.1, 7.5.1, 7.5.2, 7.5.4 status changed from FYI to Mandatory
16. Test 1.1.13 status changed from OF-FYI to OF
17. Tests 1.2.2,3,4,5,6,8 status changed from OF-FYI to OF
18. Tests 1.3.5, 6, 7, 17 status changed from OF-FYI to OF
19. Test 1.6.1 status changed from OF-FYI to OF
20. Test 2.3.4 status changed from OF-FYI to OF
21. Test 2.4.4 status changed from OF-FYI to OF
22. Test 2.6.3 status changed from OF-FYI to OF
23. Test 2.7.1-8 status changed from OF-FYI to OF

© 2026 UNH–IOL

24. Test 3.6.2 status changed from OF-FYI to OF
25. Test 3.7.1 status changed from OF-FYI to OF
26. Fixed Typo in Test 3.17 which made it appear that this test applied to devices using PCIe transport. Test does not apply to PCIe transport devices. The test only applies to Fabric transport devices.
27. Test 6.3 status changed from OF-FYI to OF
28. Updated Test 1.1.1 to address NVMe v1.4 ECN 001 requirement that when the THINP bit in the NSFEAT field is set to '1', the controller shall track the number of allocated blocks in the Namespace Utilization field, and when the THINP bit in the NSFEAT field is set to '0', the controller shall report a value in the Namespace Capacity field that is equal to the Namespace Size.
29. Updated Test 1.1.2 to address NVMe v1.4 ECN 001 requirement that the reverse domain name in an NQN that uses the non-UUID format shall not be "org.nvmexpress".
30. New test case 1.1.15 added to address TP 4027 requirements for UUID List and CNS=17h.
31. Updated Test 1.3.11 to address TP 4063 requirements for the Telemetry Host-Initialed Data Generation Number increment.
32. Updated Test 1.3.12 to address TP 4063 requirements for the Telemetry Host-Initialed Data Generation Number increment.
33. Updated Test Case 1.3.21 to relating to Data Units Written to address NVMe v1.4 ECN 001 requirement that both Write Uncorrectable commands and Write Zeroes commands shall not impact the Data Units Written value.
34. Updated Test 1.4 Case 10 to replace 'CAP.MQES' with 'NCQA'.
35. Fixed typo in 1.11.4 which incorrectly said STC=2h corresponded to Short DST.
36. Clarified correct Abort status type in 1.12.1.
37. Clarified correct Abort status type in 1.12.2
38. Clarified correct Abort status type in 1.12.3.
39. Updated Test 1.12.4 to address NVME v1.3 ECN 006 clarification around terminating DST operations with Format NVM when NSIDs are the same between commands.
40. Clarified correct Abort status type in 1.12.5.
41. New Test Case 1.12.7 to address NVME v1.3 ECN 006 clarification around terminating DST operations with Format NVM when Format NVM has NSID=FFFFFFFFh.
42. New Test Case 1.12.8 to address NVME v1.3 ECN 006 clarification around terminating DST operations with Format NVM when DST and Format NVM both have NSID=FFFFFFFFh.
43. Clarified correct Abort status type in 1.13.1.
44. Clarified correct Abort status type in 1.13.2
45. Clarified correct Abort status type in 1.13.3.
46. Updated Test 1.13.4 to address NVME v1.3 ECN 006 clarification around terminating DST operations with Format NVM when NSIDs are the same between commands.
47. Clarified correct Abort status type in 1.13.5.
48. New Test Case 1.13.6 to address NVME v1.3 ECN 006 clarification around terminating DST operations with Format NVM when Format NVM has NSID=FFFFFFFFh.
49. New Test Case 1.13.7 to address NVME v1.3 ECN 006 clarification around terminating DST operations with Format NVM when DST and Format NVM both have NSID=FFFFFFFFh.
50. New Test case 1.16.7 added to address NVMe v1.3 ECN 006 clarification that Releasing of Stream Resources when a namespace is deleted or formatted was not described completely.
51. New test case 1.17.3 added to address TP4014 proper use of the 'Sanitize Config' Feature when the FID is not savable.
52. New test case 1.17.4 added to address TP4014 proper use of the 'Sanitize Config' Feature when NDI=1 and NODRM=1.
53. New test case 1.17.5 added to address TP4014 proper use of the 'Sanitize Config' Feature when NDI=1 and NODRM=0.
54. New test case 1.17.6 added to address TP4014 proper use of the SPROG field with respect to Sanitize Status Log when Sanitize is in progress.
55. New test case 1.17.7 added to address TP4014 proper use of the SPROG field with respect to Sanitize Status Log when Sanitize is not in progress.
56. New Test case 1.19.1 added to address used of the Security Receive command with SECP=00h.

© 2026 UNH–IOL

57. New Test case 1.20.2 added to address NVMe v1.3 ECN 006 clarification that The Security Receive command specified a failure status code that does not exist (Invalid Parameter). The correct error code is Invalid Field in Command.
58. New Test case 1.20.1 added to address NVMe v1.3 ECN 006 clarification that The Security Send command specified a failure status code that does not exist (Invalid Parameter). The correct error code is Invalid Field in Command.
59. New Test case 2.12.5 added to address NVMe v1.3 ECN 006 clarification that commands that request an unsupported fused operation fail. However, the specification did not specify the error. The error is now specified.
60. New Test Case 2.13.1 to check proper handling of Read commands when DULBE set to 0.
61. New Test Case 2.13.2 to check proper handling of Read, Verify, and Compare commands when DULBE set to 1.
62. New test case 2.13.3 added to address NVMe v1.4 ECN 001 requirement that a logical block shall be marked as allocated when it is written with a Write command.
63. New test case 2.13.3 added to address NVMe v1.4 ECN 001 requirement that a logical block shall be marked as allocated when it is written with a Write Uncorrectable command.
64. New test case 2.13.3 added to address NVMe v1.4 ECN 001 requirement that a logical block shall be marked as allocated when it is written with a Write Zeroes command that does not deallocate the logical block
65. Updated Test Procedure in 3.8.1, and made this test case Mandatory based on Technical WG feedback.
66. Updated Test Procedure in 3.8.2, and made this test case Mandatory based on Technical WG feedback.
67. Clarified Test Procedure in Test 3.9.1.
68. Updated all test cases in 3.10 to indicate that the NPWG, NPWD, NPDG, and NPDA parameters are contained in the Identify Namespace Data Structure rather than the Identify Controller Data Structure. The UNH-IOL implementation of this test already handles this correctly.
69. Updated Test Procedure in 3.11.1, and made this test case Mandatory based on Technical WG feedback.
70. New test case 3.12.3 added to address requirements for TP 4050 for Endurance Group Info Enhancements.
71. New test case 3.12.4 added to address requirements for proper values of the Endurance Group Identifier.
72. New test case 3.12.5 added to address requirements for proper values of the Endurance Group Identifier
73. New test case 3.12.6 added to address requirements for proper values of the Endurance Group Identifier
74. New test case 3.12.7 added to address TP 4050 requirements for Endurance Group Critical warnings configuration.
75. New test case 3.12.8 added to address requirements for Endurance Groups not supported.
76. Updated test case 3.13.2 to issue a Get Feature for Read Recovery Levels supported FID, and compare this to the RRLS value in the Identify Controller Data Structure, rather than using the Identify Namespace Data Structure. The UNH-IOL test implementation already handles this correctly.
77. Test cases 5.5.7 status is returned to FYI only to address NVMe v1.3 ECN 006 clarification that Requirements for validation in the LBA Range Type feature were made too strict in a previous ECN; those requirements have been relaxed (“shall” changed to “may”) to improve compatibility with earlier versions of the specification. The Possible Problems section of the test was updated to provide more detail on this condition.
78. New Test case 7.6.5 added to address NVMe v1.3 ECN 006 clarification that Reservation Acquire with preempt action and incorrect CRKEY should produce an error of Reservation Conflict.
79. Updated Test 9.1.5 to use CNS=02h rather than CNS=00h.
80. Minor edits to Test 1.1 Case 1.
81. Minor edits to Test 1.2 Case 4.
82. Minor edits to Test 1.4 Case 9.
83. Minor edits to Test 1.4. Case 10.

May 3, 2021 (Version 15.0)

© 2026 UNH–IOL

David Woolf:

Tests with Status Changes:

1. Test 2.6.3 status updated from (OF-FYI) to (OF).
2. Test 3.6.2 status updated from (OF-FYI) to (OF).
3. Test 9.1.6 status updated from (OF-FYI) to (OF).
4. Test 1.2.4, 7, 8, 9 status updated from (FYI) to (M).
5. 1.3.11, 12, 13, 14, 15, 16, 17, 23 (Get Log) status updated from (FYI) to (M).
6. 1.10.4, 5, 6 (DST) status updated from (FYI) to (M).
7. 1.11.4, 5, 6, 7, 8 (DST) status updated from (FYI) to (M).
8. 1.12.1 (Abort DST) status updated from (FYI) to (M).
9. 1.13.1 (Abort DST) status updated from (FYI) to (M).
10. 1.16.4 (Directives) status updated from (FYI) to (M).
11. 1.17.1, 2 (Sanitize) status updated from (FYI) to (M).
12. 2.1.8 (Compare) status updated from (FYI) to (M).
13. 2.2.7 (DSM) status updated from (FYI) to (M).
14. 2.5.6 (Write Uncorrectable) status updated from (FYI) to (M).
15. 2.6.3 (Flush) status updated from (FYI) to (M).
16. 2.8.1 (Atomicity) status updated from (FYI) to (M).
17. 2.9.1, 2 (AWUN) status updated from (FYI) to (M).
18. 2.11.3 (Verify) status updated from (FYI) to (M).
19. 3.6.2 (IO Determinism) status updated from (FYI) to (M).
20. 3.8.1, 2 (PMR) status updated from (FYI) to (M).
21. 7.8.1, 2, 3, 4(Reservation Types) status updated from (FYI) to (M).
22. 8.6 (APST) status updated from (FYI) to (M).
23. 8.7 (APST) status updated from (FYI) to (M).
24. 8.8.1, 2 (APST) status updated from (FYI) to (M).
25. 9.1.5 (NS Mgmt) status updated from (FYI) to (M).
26. 9.2.3, 4 (NS Mgmt) status updated from (FYI) to (M).

Tests with modifications or New Tests:

1. Added Test 1.2 Case 13 for checking new requirements around which FIDs are namespace specific.
2. Removed unnecessary Observable Results in Test 1.3.11.
3. Fixed Typo in Test 1.3.21 test procedure. UNH-IOL script implementation is already correct.
4. Updated test 1.12.6 to check for support for NVMe v1.4 or higher. Clarified Observable Result that status is checked in the Self-test Result Data Structure, not in the Command Completion.
5. Updated test 1.13.5 to check for support for NVMe v1.4 or higher. Clarified Observable Result that status is checked in the Self-test Result Data Structure, not in the Command Completion.
6. Test 2.11.3 test procedure modified to clarify that all Verify commands in this test case are performed with NSID=FFFFFFFFh, namely step 3 of the procedure should be modified to say “Perform a Verify command of LBADS bytes and NSID=FFFFFFFFh. If the name-space is formatted with protection information, ensure the Verify Command of LBADS bytes with NSID=FFFFFFFFh uses the same protection information as the namespace was formatted with, and PRACT=0.”
7. Updated test procedure in Test 2.13.1 to include the Write operation necessary to complete the test. This is already included in the UNH-IOL script implementation of this test.
8. Changed the order of commands in Test 3.6 Case 1.
9. Changed the order of commands in Test 3.6 Case 2.
10. Added Test 3.6 Case 3, Predictable Latency Mode Not Enabled.
11. Clarified Test Procedure in Test 3.7 Case 1.
12. Test 3.12.6 test procedure modified to query ENDGIDMAX, then send a Get Feature with FID=18h with ENDGID = ENDGIDMAX+1. The observable result is unchanged.

© 2026 UNH–IOL

13. Clarified procedure in Test 5.2. No changes to Observable Results.
14. Test 5.5.3 added a check for OACS Bit 2=1 to make sure the test is applicable.
15. Clarified Test Procedure in Test 7.9 Case 1, 2, 3.
16. Clarified Test Procedure in Test 7.2 Case 1, 2, 3, 4, 5
17. Added Test 1.2 Case 13 for requirements in TP 4009 Domains and Partitioning around the Get Features Commands for feature values that apply to a namespace and the NSID field is set to FFFFFFFFh.
18. Added Test 1.2 Case 14 for requirements in TP 4009 Domains and Partitioning around the Get Features Commands for feature values that apply to a namespace and the NSID field is set to FFFFFFFFh: If the NSID field is set to FFFFFFFFh for the Get Features command, the controller shall, unless otherwise specified fail abort the command with a status code of Invalid Namespace or Format unless otherwise specified.
19. Added Test 1.2 Case 15 for requirements in TP 4009 Domains and Partitioning around the Set Features Command for feature values that apply to a namespace and the NSID field is set to FFFFFFFFh. If the NSID field is set to FFFFFFFFh for the Set Features command, and the MDS bit is set to '1' in the Identify Controller data structure, the controller shall abort the command with Invalid Field in Command; or If the NSID field is set to FFFFFFFFh for the Set Features command, and the MDS bit is cleared to '0' in the Identify Controller data structure, unless otherwise specified, the controller shall set the specified feature value for all namespaces attached to the controller processing the command.
20. Added Test 1.3.24 to check requirements around the Domain Identifier field in the Endurance Group Log page as specified in TP 4009.
21. Added new Observable Result to Test 1.1.2 to check proper use of the Command Size Limit fields (DMRL, DMRSL, and DMSL) according to TP 4040.
22. Updated Test Procedure in Test 2.5.5 to check for Command Size Limit support.
23. Added Test 2.5.7 to check for proper uses of Non-MDTS Command Size Limits in the Write Uncorrectable Command per TP 4040.
24. Updated Test Procedure in Test 2.7.5 to check for Command Size Limit support.
25. Added Test 2.7.11 to check for proper uses of Non-MDTS Command Size Limits in the Write Zeroes Command per TP 4040.
26. Added Test 2.2.8 to check that a DUT does not return status Command Size Limit Exceeded when ONCS Bit 2 = 1 and DMRL is exceeded per TP 4040.
27. Added Test 2.2.9 to check that a DUT does not return status Command Size Limit Exceeded when ONCS Bit 2 = 1and DMRSL is exceeded per TP 4040.
28. Added Test 2.2.10 to check that a DUT does not return status Command Size Limit Exceeded ONCS Bit 2 = 1 and DMSL is exceeded per TP 4040.
29. Added Test 2.2.11 to check that a DUT returns status Command Size Limit Exceeded when ONCS Bit 2 = 0 and DMRL is exceeded per TP 4040.
30. Added Test 2.2.12 to check that a DUT returns status Command Size Limit Exceeded when ONCS Bit 2 = 0 and DMRSL is exceeded per TP 4040.
31. Added Test 2.2.13 to check that a DUT returns status Command Size Limit Exceeded when ONCS Bit 2 = 0 and DMSL is exceeded per TP 4040.
32. Added Test 2.11.4 to check that the VSL field is checked when a Verify command is performed per TP 4040.
33. Updated procedure for test 2.2.1, 2, 3, 4, 5, 6, 7 to check for Command support non-MDTS Command Size Limit support per TP 4040.
34. Updated procedure for test 2.5.1, 2, 3, 4, 6 to check for Command support non-MDTS Command Size Limit support per TP 4040.
35. Updated procedure for test 2.7.1, 2, 3, 4, 6, 7, 8, 9, 10 to check for Command support non-MDTS Command Size Limit support per TP 4040.
36. Updated procedure for test 2.11.1, 2, 3 to check for Command support non-MDTS Command Size Limit support per TP 4040.
37. Updated Observable Results for Test 4.3 per TP 4056 requirements.
38. Added Test 1.2.16 to check that the I/O Command Set Profile Feature (19h) is implemented if bit 43 is set to '1' in CAP.CSS per TP 4056.
39. Updated Test Procedure for Test 2.8.1 per TP 4056.

© 2026 UNH–IOL

40. Added Test 2.8.2 to check requirements around AWUN, AWUPF, and AWWU per TP 4056.
41. Added Test 1.2.17 to check that the controller shall abort a command that specifies an index (IOCSCI) that corresponds to an I/O Command Set Combination that has a value of 0h with a status of I/O Command Set Combination Rejected per TP 4056.
42. Added Test 1.2.18 to check that the controller shall abort a command that specifies an index (IOCSCI) that corresponds to an I/O Command Set Combination that is not supported with a status of I/O Command Set Combination Rejected per TP 4056.
43. Updated Test Procedure and Observable Results for test 1.1.4 around the NID and NIDT fields when CNS=03h per TP 4056.
44. Added Test 1.1.16 to check for proper use of the CSI field across a variety of CNS values per TP 4056.
45. Added Test 1.1.17 to check that the Identify I/O Command Set data structure (CNS 1Ch) is implemented if bit 43 is set to '1' in CAP.CSS per TP 4056.
46. Added Test 1.1.18 to check that Identify Commands with CNS values of 00h, 11h, and 16h complete with the correct error code when the Namespace is not associated with the NVM command set.
47. Added Test 1.1.19 to check requirements around CNS 05h when I/O Command Set associated with the namespace identified by the NSID field does not support the Identify Namespace data structure specified by the CSI field, that the controller shall abort the command with a status of Invalid Field in Command per TP 4056.
48. Added Test 1.1.20 to check requirements around CNS 05h with NSID=FFFFFFFFh per TP 4056.
49. Added Test 1.1.21 to check requirements around CNS 06h when the command set indicated in the CSI field is not supported per TP 4056.
50. Added Test 1.1.22 to check requirements around CNS 1Bh when the command set indicated in the CSI field is not supported per TP 4056.
51. Added Test 1.1.23 to check requirements around CNS 1Bh when the Namespace ID is invalid, per TP 4056.
52. Added Test 1.1.24 to check requirements around CNS 1Ch and which I/O Command Set Combination fields should have a value of 0h, per TP 4056.
53. Added Test 9.3.3 to check requirements around Namespace Attach commands when the namespace does not support the same I/O Command Set as the controller per TP 4056.
54. Added Test 9.3.4 to check requirements around Namespace Attach commands when the namespace supports the same I/O Command Set as the controller, but that command set is not enabled by the current I/O Command Set Profile feature, per TP 4056.
55. Added Test 1.6.11 to check new requirements around the Format NVM command. If bit 3 in the FNA field is set to '0' and a Format NVM command has the NSID field set to FFFFFFFFh, then the command shall complete successfully.
56. Added Test 1.6.12 to check new requirements around the Format NVM command. If bit 3 in the FNA field is set to '1' and a Format NVM command has the NSID field set to FFFFFFFFh, then the controller shall abort the command with a status code of Invalid Field In Command.
57. Added new Observable Result and slightly modified procedure to test 1.1.2 to check requirements of the MAXDNA field per TP 4078.
58. Added new Observable Result and slightly modified procedure to test 1.1.2 to check requirements of the MAXCNA field per TP 4078.
59. Added new Test 9.3.5 to check that an attempt to attach a new Namespace when the MAXDNA Namespace attachment limit has been met, is aborted with status of Namespace Attachment Limit Exceeded per TP 4078.
60. Added new Test 9.3.6 to check that an attempt to attach a new Namespace when the MAXCNA Namespace attachment limit has been met, is aborted with status of Namespace Attachment Limit Exceeded per TP 4078.

September 23, 2021(Version 16.0)

David Woolf:

Tests with Status Changes:

© 2026 UNH–IOL

1. Removed Test 2.2.3 “Deallocate Out of Range”, since it is only recommended behavior, not a specification requirement. Sub cases after this test were renumbered.
2. Test 1.1.7 updated from FYI to M
3. Test 1.2.7 updated from FYI to M
4. Test 1.2.10 updated from FYI to M
5. Test 1.2.11 updated from FYI to M
6. Test 1.3.11 updated from FYI to M
7. Test 1.3.12 updated from FYI to M
8. Test 1.3.13 updated from FYI to M
9. Test 1.3.14 updated from FYI to M
10. Test 1.3.15 updated from FYI to M
11. Test 1.3.16 updated from FYI to M
12. Test 1.3.17 updated from FYI to M
13. Test 1.3.20 updated from FYI to M
14. Test 1.10.7 updated from FYI to M
15. Test 1.10.8 updated from FYI to M
16. Test 1.11.5 updated from FYI to M
17. Test 1.13.1 updated from FYI to M
18. Test 1.13.2 updated from FYI to M
19. Test 1.13.3 updated from FYI to M
20. Test 1.13.4 updated from FYI to M
21. Test 1.17.1 updated from FYI to M
22. Test 1.17.2 updated from FYI to M
23. Test 2.4.11 updated from FYI to M
24. Test 2.6.3 updated from FYI to M
25. Test 2.7.10 updated from FYI to M
26. Test 3.6.2 updated from FYI to M
27. Test 8.6.1 updated from FYI to M
28. Test 8.7.1 updated from FYI to M
29. Test 8.8.1 updated from FYI to M
30. Test 8.8.2 updated from FYI to M
31. Test 9.1.6 updated from FYI to M
32. Test 9.2.3 updated from FYI to M
33. Test 9.2.4 updated from FYI to M
34. Test 1.3.8 updated from OF-FYI to OF
35. Test 1.3.20 updated from OF-FYI to OF
36. Test 2.4.11 updated from OF-FYI to OF
37. Test 2.6.3 updated from OF-FYI to OF
38. Test 2.7.1 updated from OF-FYI to OF
39. Test 2.7.2 updated from OF-FYI to OF
40. Test 2.7.3 updated from OF-FYI to OF
41. Test 2.7.4 updated from OF-FYI to OF
42. Test 2.7.5 updated from OF-FYI to OF
43. Test 2.7.6 updated from OF-FYI to OF
44. Test 2.7.7 updated from OF-FYI to OF
45. Test 2.7.8 updated from OF-FYI to OF
46. Test 3.6.2 updated from OF-FYI to OF
47. Test 6.3.1 updated from OF-FYI to OF
48. Test 9.1.1 updated from OF-FYI to OF
49. Test 9.1.2 updated from OF-FYI to OF
50. Test 9.1.4 updated from OF-FYI to OF
51. Test 9.1.5 updated from OF-FYI to OF

New and Modified test cases:

© 2026 UNH–IOL

1. Test 1.1.2 updated per requirements in NVMe v1.4 ECN 004 that the value of 00b in the NODMMAS field in the Identify Controller Data Structure is only allowed for controllers compliant with versions 1.3 and earlier of the specification or that have bits 2:0 of the SANICAP field cleared to 0h.
2. Test 1.2.11 updated to be only applicable if FID 03h is changeable.
3. Added Test 1.3.25 to check requirements from NVMe v1.4 TP 4071b that if Log Page 00h is supported, that the FID Scope (FSP) field for any given feature, shall have no more than 1 bit set to '1', or else be set to 0.
4. Added Test 1.3.26 to check requirements from NVMe v1.4 TP 4071b that if Log Page 12h is supported, that the FID Scope (FSP) field for any given feature, shall have no more than 1 bit set to '1', or else be set to 0, that if the FID Supported bit (FSUPP) is set to 0 for any given feature, then all fields in the FID Supported and Effects Data Structure for that feature shall be set to 0, that the Command Scope (CSP) field for any given command, shall have no more than 1 bit set to '1', or else be set to 0, and that if the Command Supported bit (CSUPP) is set to 0 for any given feature, then all fields in the NVME-MI Commands Supported and Effects Data Structure for that command shall be set to 0.
5. Added Test 1.17.8 to check requirements from NVMe v1.4 ECN 003 that if a sanitize operation is not in progress and the most recent sanitize operation did not fail, then a Sanitize command with a Sanitize Action set to 001b (i.e., Exit Failure Mode) shall complete with a status of Successful Completion and perform no other action.
6. Added Test 1.17.9 to check requirements in NVMe v1.4 ECN 007 that all I/O Commands other than a Flush command shall be aborted with a status of Sanitize In Progress.
7. Test 1.18.3 updated to fix typo in procedure, expected result should be 'command completes with status Invalid Controller Identifier'.
8. Added Test 1.2.19 to check requirements in NVMe v1.4 ECN 004 that if a Feature is not persistent across power cycles and resets, then the current value of that Feature shall be set to the default value of that Feature as part of a Controller Level Reset.
9. Added Test 2.6.4 to check requirements in NVMe v1.4 ECN 007 that if a volatile write cache is not present or not enabled, then Flush commands shall complete successfully and have no effect if a sanitize operation is not in progress.
10. Updated Names of Tests 2.13.1.
11. Test 2.13.6 added to check requirements from TP 4065b that if the read portion of a copy operation attempts to access a deallocated or unwritten logical block, the controller shall operate as described in section 6.7.1.1.
12. Test 2.14.1 added to check requirements from TP 4065b that the MSSRL and MCL fields in the Identify Namespace data structure (CNS 00h) shall be set to a non-zero value If the controller supports the Copy command .
13. Test 2.14.2 added to check requirements from TP 4065b that if the Source Range Entry specifies a Number of Logical Blocks field that is greater than the value in the MSSRL field, then the Copy command shall be aborted with a status code of Command Size Limit Exceeded.
14. Test 2.14.3 added to check requirements from TP 4065b that if the number of Source Range entries (i.e., the value in the NR field) is greater than the value in the MSRC field, then the Copy command shall be aborted with a status code of Command Size Limit Exceeded.
15. Test 2.14.4 added to check requirements from TP 4065b that if the sum of all Number of Logical Blocks fields in all Source Range entries is greater than the value in the MCL field, then the Copy command shall be aborted with a status code of Command Size Limit Exceeded.
16. Test 2.14.5 added to check requirements from TP 4065b that if the specified Descriptor Format is not supported by the controller, then the command shall be aborted with a status code of Invalid Field in Command.
17. Test 2.14.6 added to check requirements from TP 4065b that if the PRACT bit is cleared to '0' in the PRINFOR field and the PRACT bit is set to '1' in the PRINFOW field, then the Copy command shall be aborted with a status code of Invalid Field in Command.
18. Test 2.14.7 added to check requirements from TP 4065b that the controller shall return the Attempted Write to Read Only Range (82h) Status if the read-only condition on the media is a result of a change in the write protection state of a namespace

© 2026 UNH–IOL

19. Added Test 3.8.3 per NVMe v1.4 ECN 006 to check that the PMRMSCL and PMRMSCU registers are not changed on Controller Reset.
20. Updated Test 3.8.1 per NVMe v1.4 ECN 006 to properly reference the PMRMSCL and PMRMSCU registers.
21. Added Test 3.12.9 per NVMe v1.4 ECN 004 to check requirements that if a namespace is associated with an NVM set, NOWS defined for this namespace shall be set to the Optimal Write Size field setting defined in NVM Set Attributes Entry for the NVM Set with which this namespace is associated.
22. Added Test 3.14.3, 4, 5 to check requirements per NVMe v1.4 ECN 008 that a controller shall not send an Asymmetric Namespace Access Change event notice if a Namespace Attribute Changed notice is sent for the same event, such as a the change is due to the attachment of a namespace, or the deletion of a namespace, or the detachment of a namespace.
23. Added Tests 3.17.3 to check requirements per NVMe v1.4 ECN 004 that if SGL Descriptor Sub Type is set to 1h and SGL Descriptor Type is set to 1h, The controller shall abort the command with the status of SGL Descriptor Type Invalid.
24. Added Tests 3.17.4 to check requirements per NVMe v1.4 ECN 004 that if SGL Descriptor Sub Type is set to 4h and SGL Descriptor Type is set to 1h, The controller shall abort the command with the status of SGL Descriptor Type Invalid.
25. Test 9.3.5 updated to be only applicable to devices with multiple controllers in a single domain.
26. Added Test 2.3.12 per requirements in NVMe v1.4 ECN 003 that if bit 0 of the PRCHK field is set to '1', and the namespace is formatted for Type 1 protection, the controller shall complete the command with a status of Invalid Protection Information if the ILBRT field does not match the least significant four bytes of the SLBA field.
27. Added Test 2.3.13 per requirements in NVMe v1.4 ECN 003 that if bit 0 of the PRCHK field is set to '1', and the namespace is formatted for Type 1 protection, the controller shall complete the command with a status of Invalid Protection Information if the EILBRT field does not match the least significant four bytes of the SLBA field.
28. Eliminated some Observable Results in Test 1.3.11 that were not required.
29. Updated Observable Results in Test 2.13.2.
30. Clarified Test Procedure in Test 1.4.11, to indicate that test is not applicable if the DUT does not support NVMe v1.3 or higher.
31. Updated Test 1.6.3 and 1.6.4 to account for the DPS value being reported slightly differently in NVMe v1.4 compared to earlier specification versions.

January 21, 2022 (Version 17.0)

Tim Sheehan:

Tests with Status Changes:

1. Tests 1.1.19, 20, 21, 22, 23, 24 updated from (FYI) to (M)
2. Tests 1.2.13, 14, 15 updated from (FYI) to (M)
3. Test 1.3.22 updated from (FYI) to (M)
4. Test 1.4.9 updated from (FYI) to (M)
5. Tests 1.19.1 & 2 updated from (FYI) to (M)
6. Test 1.20.1 updated from (FYI) to (M)
7. Tests 2.2.8, 9, 10, 11, 12 updated from (FYI) to (M)
8. Test 2.3.11 updated from (FYI) to (M)

New and Modified test cases:

1. The root of this test plan is the UNH-IOL NVMe Conformance test suite, version 16.0.
2. This version takes into account the NVMexpress.org's 2.0 effort in refactoring NVMe Technical Specifications.
3. Test cases 6.1, 6.2, 6.4, 8.1-8.8 and 10.1 were moved to the UNH-IOL_PCIe_Transport_Conformance_Test_Suite_v17
4. Added tests 1.1.25-1.1.34 per requirements for TP4105a IO Command Set Independent data structure
5. Added tests 1.3.27 & 1.3.28 per requirements for TP4083 PEL Context Counter

© 2026 UNH–IOL

6. Added tests 1.21.1-1.21.10 per requirements for TP4046a Command Group Control feature
7. Corrected Test 1.1.20 Test Procedure 1 to remove the “not” so it actually tests Namespace Management not supported
8. Numerous Tables throughout the test suite were updated to 2.0 refactored status
9. Test 1.2.14 Observable Results step 1 updated for FID 03h
10. Test 2.12.3 Test Procedure 4 added to provide a non-fused command to correct test

July 14, 2022 (Version 18.0)

Tim Sheehan:

Tests with Status Changes:

1. Test 1.1.16 updated from (FYI) to (M)
2. Test 1.2.12 updated from (FYI) to (M)
3. Tests 1.3.18 & 21 updated from (FYI) to (M)
4. Test 1.6.11 updated from (FYI) to (M)
5. Tests 1.12.7 & 8 updated from (FYI) to (M)
6. Tests 1.13.6 & 7 updated from (FYI) to (M)
7. Test 2.6.4 updated from (FYI) to (M)

New and Modified test cases:

1. Added New Tests 1.3.29-35 per requirements in TP4075a, get Log Page Index Offsets
2. Added New Tests 1.3.36-44 per requirements in TP4079, Telemetry Log Size Changes
3. Modified Test 4.18 to add in version check for 1.4 and 2.0 version of NVM Express Base Specification
4. Added New Tests 1.3.46-66, 70 & 71, 1.1.35&36, 1.22.1-15, 1.3.67&68, 3.12.10&11and 8.2.5-9 per requirements in TP4052c, Endurance Group Management
5. Added new Tests 1.3.71-73, per requirements in TP4109, Allow Host to specify Telemetry Host-Initiated data area
6. Added new Tests 1.1.37-43, per requirements in TP4095, Namespace Capability Reporting
7. Modified test 1.6.8 for version check to utilize PIL field, per requirements in ECN 101 2021.07.26
8. Modified tests 1.6.1-6 and 1.6.7-10, per requirements in ECN 101 2021.07.26
9. Modified test 1.3.26 for version check, per requirements in ECN 102 2022.01.04
10. Added New Tests 3.2.3-6, per requirements in ECN 102 2022.01.04
11. Modified tests 1.1.20&25, 1.3.26, 1.6.5, for version checking
12. Modified tests 1.6.1-4 and 1.6.7-10 for checking of FNA bit 3
13. Tests 1.10.8 and 1.11.8 test case names were slightly modified to include ‘second DST’

January 04, 2023 (Version 19.0)

Tim Sheehan:

Tests with Status Changes:

1. Test 1.1.28 updated from (FYI) to (M)
2. Test 1.1.29 updated from (FYI) to (M)
3. Test 1.1.33 updated from (FYI) to (M)
4. Test 1.1.34 updated from (FYI) to (M)
5. Test 1.13.5 updated from (FYI) to (M)
6. Test 1.21.2 updated from (FYI) to (M)
7. Test 2.13.1, 3, 5 updated from (FYI) to (M)
8. Test 3.12.8 updated from (FYI) to (M)
9. Test 5.5.7 updated from (FYI) to (M)
10. Test 1.4.9 updated from (M) to (FYI)

New and Modified test cases:

1. Added New Test 1.3.76, per requirements of ECN105
2. Modified test cases 1.3.18, 27, 28, & 73, per requirements in ECN109

© 2026 UNH–IOL

3. Added New Observable Result to test case 4.3.1, per requirements of ECN 110
4. Added New Tests 1.2.24-31, per requirements of TP4077 Performance Characteristics Reporting
5. Added New Tests 1.3.74, and 1.2.20-23, per requirements in TP4094 Management Network Addresses
6. Added New test 2.9.3, per requirements in TP4098 Atomicity Alignment Enhancements
7. Added New Test 1.3.75, per requirements in TP4113, Opcode Field to Error Information
8. Added New Test 1.1.44, per requirements in TP4135 NVMe Spec Version Reporting
9. Added New Tests 3.10.5-8, per requirements in TP4116 Optimal Read Size and Granularity
10. Added New Tests 3.2.7-13, per requirements in TP4141Storage Tag Mask Enhancements
11. Added 'Possible Problem' statement to many test cases per requirement in TP4110
12. Added New Tests 1.6.13-16, 8.2.10, per requirements in TP4140 Format Index Mismatch Protection
13. Added New Tests 1.2.32-34, per requirements in TP4110 Align PCIe and Fabrics HOSTID Management
14. Added New Tests 8.4.1-4, per requirements in TP4034 Dispersed namespaces
15. Modified Test 3.2.6 to reference MC instead of MS
16. Modified Test 1.6.8 to handle Invalid Field in Command' return status
17. Modified Test 2.11.2 to add a procedure step to repeat for all PRINFO values
18. Modified Test 2.3.13 to Perform a write on the SLBA for validity
19. Modified Tests 1.22.11, 1.22.15, 8.2.5, 8.2.8, and 8.2.9 for correctness.
20. Modified Test 1.12.6 to check Sanitize status

July 19, 2023 (Version 20.0)

Tim Sheehan:

Tests with Status Changes:

1. Test 1.3 case 29 updated from (FYI) to (M)
2. Test 1.3 case 32 updated from (FYI) to (M)
3. Test 1.3 case 33 updated from (FYI) to (M)
4. Test 1.3 case 36 updated from (FYI) to (M)
5. Test 1.3 case 40 updated from (FYI) to (M)
6. Test 1.4 case 11 updated from (FYI) to (M)
7. Test 1.17 case 6 updated from (FYI) to (M)
8. Test 1.17 case 7 updated from (FYI) to (M)
9. Test 1.17 case 8 updated from (FYI) to (M)
10. Test 1.17 case 9 updated from (FYI) to (M)
11. Test 1.7 case 3 updated to FYI
12. Test 3.8 case 1&2 updated to IP
13. Test 3.11.1 case 1 updated to IP

New and Modified test cases

1. Modified Test 1.14.1 clarified procedure steps 2&3
2. Modified Tests 7.9.1&2 procedure step 3.b.ii
3. Modified Test 1.1.44 procedure step wording
4. Modified Tests 3.2.7-11 Clarified PIC Bit 2 in procedure steps
5. Modified Test 5.5.1 Possible Problem added for status code returned
6. Modified Test 3.10.6 Observation Result 1 for clarity
7. Modified Tests 1.2.2,3,& 4 Grammar
8. Modified Test 3.2.6 to clarify Procedure steps
9. Modified Test 1.6.8 add new status code in results
10. Modified Test 1.12.6 added procedure step to check sanitize status
11. Modified Test 1.22.2 to remove skip procedure step
12. Modified Test 1.1.24 to remove CSI from procedure step and modified CAP.CSS bit

© 2026 UNH–IOL

13. Modified Test 1.1.2 to remove Observation Result 8 to follow NVMe Specification
14. Modified Test 2.3.12&13 to use correct variable name
15. Modified Test 2.14.7 to clarify return status code
16. Modified Test 2.11.2 observable result
17. Modified Test 1.7.5 name to reflect test procedures
18. Added New Tests 1.6.17, 1.7.5, 3.14.6, 1.3.77, & 1.17.10 per requirements in ECN 105
19. Added New Tests 1.2.35 & 36 per requirements in TP4142 Temperature Threshold Hysteresis
20. Added New Tests 1.2.44&45, 8.1.7, 1.1.45, 1.6.19, 1.3.79, 8.1.8 per requirements in TP4136 Should to Shall
21. Modified Tests 1.6.8, 5.5.7, 2.11.2, & 3.2.1 per requirements in TP4136 Should to Shall
22. Added New Tests 1.2.37-43 per requirements in TP4145 Namespace Admin Label
23. Added New Tests in Group 9, Flexible Data Placement per requirements in TP4146a Flexible Data Placement
24. Added New Tests 1.5.3&4, 1.3.80&81, 1.23.1-8 per requirements in TP4097a Abort Enhancements
25. Added New Test 1.16.8 per requirements in TP4114 Vendor Specific Directives

January 18, 2024 (Version 21.0)

Carter Snay:

Tests with Status Changes:

1. Test 1.3.28 updated from (FYI) to (M)
2. Test 1.3.31 updated from (FYI) to (M)
3. Test1.12.6 updated from (FYI) to (M)
4. Test 3.2.3 updated from (FYI) to (M)
5. Test 3.2.4 updated from (FYI) to (M)
6. Test 3.2.5 updated from (FYI) to (M)
7. Test 3.10.6 updated from (FYI) to (M)

New and Modified test cases

1. Added new group 10 tests 10.1.1-7 per requirements in TP4156a
2. Added new test 1.3.82 as per requirements in TP4113a
3. Added new tests 1.1.46 and 1.1.47 per requirements in TP4162a
4. Added new tests 1.1.48-51 per requirements in TP4155
5. Added new tests 1.24.1-3 to address updates for KATO in ECN117
6. Added new tests 1.25.1-6 per requirements in TP4088
7. Modified test 3.9.1 to allow for action type 00h and 01h to be unsupported per new requirements in TP4165
8. Added new tests 3.9.2 and 3.9.3 per new requirements in TP4165 using the new Action type 2h
9. Added new tests 1.3.83, 4.19 and 4.20 per new requirements in ECN111.
10. Modified test 1.17.3 by adding a second Get Feature Command with SEL=001b which is required by the observable result but was not explicitly stated.
11. Modified case 1.7.3 to specify setting the RAE set to 0 in the Get Log page command, and added observable result 3 to verify that each AE is cleared to zero due to the Get Log Page command with RAE = 0 for each event type. This caused what was previously observable result 3 to become observable result 4.
12. Added new test case 1.2.46 per requirements of ECN113
13. Added new test cases 2.14.9-24 to address updates to the Copy Command and Cross Namespace Copy TP4130.
14. Modified test 3.6 case 3 to check if CTRATT bit 5 is 0, not 1.
15. Modified test 1.7 case 3 step 1b to set RAE = 0.

August 1, 2024 (Version 22.0)

Carter Snay:

Tests with Status Changes:

© 2026 UNH–IOL

1. Test 1.1.14 updated from (FYI) to (M)
2. Test 1.1.15 updated from (FYI) to (M)
3. Test 1.1.35 updated from (FYI) to (M)
4. Test 1.1.37 updated from (FYI) to (M)
5. Test 1.1.38 updated from (FYI) to (M)
6. Test 1.1.41 updated from (FYI) to (M)
7. Test 1.1.45 updated from (FYI) to (M)
8. Test 1.2.43 updated from (FYI) to (M)
9. Test 1.2.45 updated from (FYI) to (M)
10. Test 1.3.19 updated from (FYI) to (M)
11. Test 1.3.24 updated from (FYI) to (M)
12. Test 1.3.25 updated from (FYI) to (M)
13. Test 1.3.26 updated from (FYI) to (M)
14. Test 1.3.27 updated from (FYI) to (M)
15. Test 1.3.34 updated from (FYI) to (M)
16. Test 1.3.37 updated from (FYI) to (M)
17. Test 1.3.41 updated from (FYI) to (M)
18. Test 1.3.42 updated from (FYI) to (M)
19. Test 1.3.43 updated from (FYI) to (M)
20. Test 1.3.44 updated from (FYI) to (M)
21. Test 1.3.71 updated from (FYI) to (M)
22. Test 1.3.78 updated from (FYI) to (M)
23. Test 1.3.79 updated from (FYI) to (M)
24. Test 1.3.80 updated from (FYI) to (M)
25. Test 1.7.3 updated from (FYI) to (M)
26. Test 1.7.5 updated from (FYI) to (M)
27. Test 1.17.4 updated from (FYI) to (M)
28. Test 1.17.5 updated from (FYI) to (M)
29. Test 1.21.1 updated from (FYI) to (M)
30. Test 1.21.7 updated from (FYI) to (M)
31. Test updated 2.6.5 (previously 9.5.2) from (FYI) to (M)
32. Test 2.7.11 updated from (FYI) to (M)
33. Test 2.12.2 updated from (FYI) to (M)
34. Test 3.4.2 updated from (FYI) to (M)
35. Test 3.4.3 updated from (FYI) to (M)
36. Test 3.4.4 updated from (FYI) to (M)
37. Test 3.4.5 updated from (FYI) to (M)
38. Test 3.12.3 updated from (FYI) to (M)
39. Test 3.12.4 updated from (FYI) to (M)
40. Test 3.12.5 updated from (FYI) to (M)
41. Test 3.12.6 updated from (FYI) to (M)
42. Test 3.12.7 updated from (FYI) to (M)
43. Test 3.12.10 updated from (FYI) to (M)
44. Test 3.12.11 updated from (FYI) to (M)
45. Test 3.15.1 updated from (FYI) to (M)
46. Test updated 8.1.7 from (FYI) to (M)
47. Test updated 8.1.8 from (FYI) to (M)
48. Test updated 8.2.5 from (FYI) to (M)
49. Test updated 8.2.6 from (FYI) to (M)

New and Modified test cases

1. Added test cases 84 and 85 to Test 1.3 Get Log Page Command for requirements defined in TP4167
2. Added test case 25 to Test 2.14 Copy Command to address requirements in TP4175

© 2026 UNH–IOL

3. Added new test case to Test 2.6 Flush Command, Case 6, addressing need to test Flush performance over a power cycle.
4. Modified test 1.7.3 Asynchronous Events Clearing Events by removing expectation of a command status of aborted in observable result 2.
5. Modified test 3.6.3 IO Determinism Test Procedure step 1 to check if CTRATT bit 5 is cleared to 0 instead of set to 1
6. Modified test 2.14.15 Copy Command NVM Copy Single Atomicity Verification by adjusting destination LBA range from 2-4 to 2-3 in step 3 and 4 of the Test Procedure, Fixing Typo in observable result 3 to refer to Test Procedure step 4 instead of step 6, and adding a new step to the Test Procedure (step 5) to read the targeted LBA range to verify the copied data is present.
7. Modified test titles for 1.6.18 and 1.6.19, they have been swapped. The original test titles were reversed.
8. Modified test case 1.16.6 by removing requirement to skip test if Streams Directive is not supported, and by adding a step to recreate the original namespace at the end of the test to prevent errors in subsequent tests.
9. Modified Copy Command test 2.14.8 to add a step to check for support of Copy Format 2h and 3h.
10. Modified test case 8.3.4 by adding more clarification steps and select alternative command sets if available
11. Moved test case 9.5.2 from the Flexible Data Placement Group 10 tests into Group 2.6 Flush command tests. This test does not require Flexible Data Placement Support, and is only testing the Flush command success over a Controller Level Reset.
12. Modified test 1.2.7 by adding a check for the value of MDS in CTRATT. If it is set to 1, the test no longer expects a successful status response on Observable Result 3.
13. Modified test cases 3.2.7-11 by changing the bit to check for Protection Information Capabilities field Qualified Protection Information Format Support (QPIFS). The test also now checks performs an Identify with CNS Ah and CNS 5h to distinguish between unique LBAs.
14. Modified test 1.6.8 to allow for successful status in observable result 1 and 2 in the case that the command does not complete with an error. Updates to the NVM I/O Command Set specification 1.0d allows for all commands in the test procedure to complete successfully, or if they are aborted to return a specific status. In the case that the commands are not aborted, the test has to be updated to expect success.
15. Updated Possible Problems section in test 6.1 Controller Level Reset – Controller Reset
16. Added new test case 3.14.7 Asymmetric Namespace Access - IOS=1, OT=1 to test the offset field updates from ECN119
17. Added test case 2.12.6 to test that a controller returns the status code of “Command Aborted due to Failed Fuse Command” when one part of the fused operation is not successful, addressing requirements in ECN119.
18. Modified test 1.6.10 to remove FNA bit check, and modified test step #5 to reference SES≠0 to address requirements in ECN115
19. Updated test 1.6.3 to check bit 3 of FNA instead of bit 2
20. Added test case 1.3.86 to address ECN119 requirement to list the port number in the PEL.
21. Added test case 52 to test 1.1 to verify the capability of CNS 02h ensuring they meet the requirements of ECN115
22. Added new group 11 for Boot Partitions to test the requirements made in TP4170.
23. Added tests 1.17.11-29 and 1.7.6-9 to address requirements in TP4152.
24. Added 3 cases to test 3.4 to address requirements for the new HMB With Low Power Support feature from TP4104a.
25. Added case 4 to test case 9.4 FDP Events Log Page, Log Identifier 23h to address new requirement in TP4146b
26. Added cases 12-17 to test 2.7 Write Zeroes and added a new Identify test to check for expected supported bits in the I/O Command Set Specific Identify Controller Data Structure to meet requirements in TP4160.
27. Fixed issue in 1.2.1 with mislabeling of the Observable Results

© 2026 UNH–IOL

28. Modified test 1.14 to allow for a status code of Invalid Opcode, because TP6028 changed VPD Write support from mandatory to optional.
29. Subcases added to several tests that require multiple FIDs, LIDs, or metadata transfer methods. The subcases are added to increase the granularity of the tests. These new subcases are FYI, and are re-implementations of the parent test.

2025 January 23 (Version 23.0)

Carter Snay:

Tests with Status Changes:

1. Test 1.2.10 updated from (FYI) to (M)
2. Test 1.3.81 updated from (FYI) to (M)
3. Test 1.14.1 updated from (FYI) to (M)
4. Test 3.2.11 updated from (FYI) to (M)
5. Test 3.2.12 updated from (FYI) to (M)
6. Test 3.2.13 updated from (FYI) to (M)
7. Test 3.10.5 updated from (FYI) to (M)
8. Test 3.10.8 updated from (FYI) to (M)
9. Test 4.19.1 updated from (FYI) to (M)
10. Test 4.20.1 updated from (FYI) to (M)
11. Test 10.1.7 updated from (FYI) to (M)

New and Modified test cases

1. Updated test 1.22.4 to make the test less likely to skip if the selected configuration is not cleared to 0.
2. Updated test 1.2.15 subcase 3 to check for a status code of Invalid Field in Command rather than success.
3. Updated test 1.2.3 and 1.9.1 subcases to only include FIDs that have values that are persistent across resets.
4. Updated test 3.4.8 Observable Result 2 to check for bit 1 in the completion queue entry DWORDs to be set to ‘1’ instead of cleared to ‘0’.
5. Added a subcase to tests 2.11.1-3 and 1.6.7-8 to account for when MS = 0.
6. Modified test 1.23 by removing over Fabrics testing, as this test is not applicable for over Fabrics testing.
7. Updated subcases that expect mandatory support of LID 13h, which is optional if NVMe-MI Send and Receive commands are not supported. This affected test 1.3.1, 1.3.29, and 1.3.33
8. Removed subcases 1.7.4.1 and 1.7.4.4, which were found to be not applicable to the test procedure of 1.7.4.
9. New test cases 1.2.47,48 and FID subcases added to address requirements in ECN120.
10. Added subcase 29 for LID 1Ch to tests 1.3.1, 1.3.10, 1.3.22, 1.3.29, 1.3.30, 1.3.31, 1.3.32, 1.3.33, 1.3.34, 1.3.35.
11. Added subcase 1.3.78.1 for LID 1Ch.
12. Updated test case 1.3.1 with a possible problem referring to the possibility that the device is NVM Express Base Specification, Revision 2.0 or greater and supports LID 00h, or is a 1.4 device, and supports LID 00h.
13. Added new tests cases 1.7.10 to verify the Attached and Allocated Namespace Attribute Notice Asynchronous Event work as expected by TP4150.
14. New test cases 1.6.20 added To test for error condition when a namespace is formatted with a non-zero STS when in a different format & LBAFEE = 1 to test requirements in TP4186.
15. Added group 12 to test new conformance requirements and optional commands described in TP4159
16. Added new test to 1.17 Sanitize Operations to test the scenario when a Sanitize is sent to a controller that is Suspended as described in TP4159
17. Added FID 21h subcase to 1.2.1, 1.2.2, 1.2.6 to test the new Controller Data Queue feature from TP4159

© 2026 UNH–IOL

18. New Test 1.7.10 implemented to test for the Controller Data Queue Tail Pointer Event Trigger capability defined in TP4159
19. Added new test 3.9.4 to test the RALBAS bit in use with the Track Send command as defined in TP4159.
20. Added new test group 4.21 to test the NVM Subsystem Shutdown Enhancements as defined in TP4169.
21. Updated test case 3.2.7 to attempt creating namespaces with LBSTM not set to all 1s.
22. Added new subcase to 1.2.3 and 1.9.1 for FID 1Eh Flexible Data Placement Events
23. Updated test case 3.4.6 to set EHM to 0 to disable the host memory buffer feature prior to re-enabling to prevent an issue where “Command Sequence Error” was being returned.
24. Updated subcases 17 and 18 in test 1.2.7 to expect “Invalid Field in Command” statuses for FIDs 82h and 83h

2025 July 31 (Version 24.0)

Carter Snay:

Tests with Status Changes:

1. Test 12.4.2 updated from (IP) to (FYI)
2. Test Cases 1.3.83-86 updated from (FYI) to (M)
3. Test Case 1.4.9 updated from (FYI) to (M)
4. Test Cases 1.6.18 and 1.6.19 updated from (FYI) to (M)
5. Test Case 1.17.24 updated from (FYI) to (M)
6. Test Case 1.25.6 updated from (FYI) to (M)

Sub-tests with Status Changes:

1. Sub-test Cases 1.1.16.1-4,6-9 updated from (FYI) to (M)
2. Sub-test Cases 1.2.1.1-6,8-13,16,17,22,23,30 updated from (FYI) to (M)
3. Sub-test Case 1.2.3.14 updated from (FYI) to (M)
4. Sub-test Cases 1.2.5.1-14,16,17,22,23,30 updated from (FYI) to (M)
5. Sub-test Cases 1.2.10.1,3-8,10 updated from (FYI) to (M)
6. Sub-test Cases 1.2.14.1,2 updated from (FYI) to (M)
7. Sub-test Cases 1.2.15.1,2 updated from (FYI) to (M)
8. Sub-test Cases 1.2.19.2,11,12,13 updated from (FYI) to (M)
9. Sub-test Cases 1.3.1.1,2,4-9,14,19-22,26 updated from (FYI) to (M)
10. Sub-test Cases 1.3.31.1-9,14,19-22,26 updated from (FYI) to (M)
11. Sub-test Cases 1.3.33.1-9,14,19-22,26 updated from (FYI) to (M)
12. Sub-test Cases 1.3.35.1-8,14,19,20,22,26 updated from (FYI) to (M)
13. Sub-test Cases 1.7.1.1-3 updated from (FYI) to (M)
14. Sub-test Cases 1.7.7,8 updated from (FYI) to (M)
15. Sub-test Case 1.7.9.2 updated from (FYI) to (M)
16. Sub-test Cases 1.8.1.3,10-12,16,30 updated from (FYI) to (M)
17. Sub-test Cases 1.9.1.1,2,4,14 updated from (FYI) to (M)

New and Modified test cases

1. Added subcases to test 1.2 for the Embedded Management Controller Address and Host Management Agent Address and LID 18h Management Address List subcases in test 1.3 for new requirements in TP4094a Management Network Addresses.
2. New Test Case 1.24.4 to for KATO Cleanup Within Quiesce Time and Test Case 1.24.5 for Command Based Keep Alive Timer Restart for TP 4129 KATO Corrections and Clarifications
3. New Test Case 2.6.4 added with a status of IP test new requirements based on TP4194 SGL Support Clarifications. IP due to the lack of test implementation available in IOL tools at time of writing.
4. Modified the expected status of 12.4.3 to return “Not Enough Resources” instead of “Invalid Field”.
5. Added test 1.26 for the Device Self-test Host-Initiated Refresh Operation and address requirements in TP4058.

© 2026 UNH–IOL

6. Fixed issue with test 4.21.1 which was erroneously checking for CSTS.RDY to change from 1 to 0 instead of 0 to 1.
7. Removed subcase 19 from test 1.2 Case 7, the Controller Data Queue feature's scope does not apply to this test case.
8. Modified test case 9.9.4 by adding a step to ensure that the Reclaim Unit Handle Status Data Structure itself is valid by checking the reserved bits are set to 0, that the number of descriptors is actually equal to NRUHSD, and verifying that it is 0's filled after the Reclaim Unit Handle Status Data Structure, which is a result of clarifications from ECN123.
9. New test case in test 7.2 Reservation Registration command, which verifies that the Contoller ID field is FFFFh when the namespace is detached from all controllers, as per ECN122.
10. Changed 3 instances of REGCTL to REGSTRNT due to a naming convection change in ECN122
11. Modified test 1.1.41 by removing requirement to report non-zero values in the KPIO field.
12. Modified test 1.1.44 by adding the 1h minor version for NVM I/O command set support
13. Removed observable result #2 from test 12.3.2 because the Write commands completing successfully are encapsulated by the first observable, and the Write command not completing in step 5 is already mentioned in the first observable result.
14. Updated test 3.4.6 by to change the observable result to expect different values of the 0Dh feature identifier before and after enabling HMNARE.
15. Modified test 1.2.48 to change the requirement from verifying that a command is rejected when the NSID is non-zero, rather than only when it is 0.
16. Removed subcase of test 1.2 case 7, subcase 14.
17. Added new test cases 5 and 6 to test 3.17 to test the length requirements in TP 4194a SGL Support Clarifications.

2026 February 5 (Version 25.0)

Carter Snay:

Tests with Status Changes:

1. Test 1.2.19 updated from (FYI) to (M)
2. Test 1.3.9 updated from (FYI) to (M)
3. Test Cases 1.17.14-18 updated from (FYI) to (M)
4. Test Cases 1.21.8 and 1.21.10 updated from (FYI) to (M)
5. Test Case 1.26.2 updated from (FYI) to (M)
6. Test Case 2.6.6 updated from (FYI) to (M)
7. Test Case 4.21.3 updated from (FYI) to (M)
8. Test Case 12.1.2 updated from (FYI) to (M)
9. Sub-test Cases 1.1.16.10-13, 16, 17, 20, 21 updated from (FYI) to (M)
10. Sub-test Case 1.2.1.17 updated from (FYI) to (M)
11. Sub-test Cases 1.2.2.1-6, 8-13, 16, 17, 22, 23, 30 updated from (FYI) to (M)
12. Sub-test Cases 1.2.3.1, 2, 4 updated from (FYI) to (M)
13. Sub-test Cases 1.2.4.1-6, 8-14, 16, 23, 30 updated from (FYI) to (M)
14. Sub-test Cases 1.2.5.22, 23, 27, 28, 29 updated from (FYI) to (M)
15. Sub-test Cases 1.2.10.15, 16, 20 updated from (FYI) to (M)
16. Sub-test Cases 1.2.13.1, 2 updated from (FYI) to (M)
17. Sub-test Cases 1.2.19.1, 4-6, 8-10, 17 updated from (FYI) to (M)
18. Sub-test Case 1.3.1.10 updated from (FYI) to (M)
19. Sub-test Cases 1.3.10.1-3, 5-8, 19-22, 26 updated from (FYI) to (M)
20. Sub-test Cases 1.3.22.1-7, 10, 19-22, 26 updated from (FYI) to (M)
21. Sub-test Cases 1.3.23.1-7, 10, 19-22, 26 updated from (FYI) to (M)
22. Sub-test Cases 1.3.29.1-10, 14, 19-22, 26 updated from (FYI) to (M)
23. Sub-test Case 1.3.31.10 updated from (FYI) to (M)
24. Sub-test Case 1.3.33.10 updated from (FYI) to (M)
25. Sub-test Case 1.3.35.10, 21 updated from (FYI) to (M)

© 2026 UNH–IOL

26. Sub-test Cases 1.7.1.1-3 updated from (FYI) to (M)
27. Sub-test Cases 1.7.4.1, 2 updated from (FYI) to (M)
28. Sub-test Case 1.7.9.1 updated from (FYI) to (M)
29. Sub-test Cases 1.8.1.2, 5, 23 updated from (FYI) to (M)
30. Sub-test Case 2.11.3.3 updated from (FYI) to (M)
31. Sub-test Cases 5.6.1.2, 6 updated from (FYI) to (M)

New and Modified test cases

1. Updated test 3.11 to account for CMBSZ.LISTS and CMBSZ.SQS clarifications in ECN 124.
2. Updated test 9.4.6 to fix error with FDPEF, changing expected value from 111b to 11b.
3. Updated test 11.1.1 by replacing observable result 3 with a new check to verify an expected returned status to a Set Features command.
4. Modified test 1.2.4 by removing extraneous step in procedure which was already being achieved in the Observable Results.
5. Modified test 1.3.87 to fix the bit check for Sanitize Media Verification Event, which was checking bit 13 instead of bit 14.
6. Modified Subcase 2.11.2.3 to expect Success instead of error.
7. Modified case 1.1.52 by swapping step 3 and 4, such that the initial identify command contains the correct information about the namespace rather than all zeroes when it is compared with the second identify command.
8. Updated test 2.5.7 to make it clearer that if WUSL = 0h, the test is not applicable.
9. Removed 1.2.47.1 because values are persistent across resets for FID 03h.
10. Updated Cancel Command tests (1.23) to adjust for being unable to guarantee the timing of the order of commands being processed off the submission queue.
11. Updated 2.1.7 to adjust use of PI Type 3 edge case.
12. Swapped order of checks in test 8.3.5 due to dependency on NS Management Support and the list-controller opcode.
13. Updated test 1.23.1,6, and 8 to account for the fact that the test station cannot guarantee the order of commands being processed off the submission queue.
14. Modified observable result 1 of case 1.1.41 to verify that NMIC and Key Per I/O fields return non-zero values.
15. Updated test 3.11 Controller Memory Buffer Step 4 to skip if CMBSZ.LIST and CMBSZ.SQS are cleared to 0 per clarification in ECN124.
16. New test 1.1.55-60 for TP4190 PLS/MUP requirements
17. New tests 6.2.1-10 for TP4199 Self-Reported Drive Power
18. New subcase 1.9.16 for FID 24h (Power Threshold) which is expected to be saveable across resets.
19. Added subcases to tests 1.3.1, 1.3.10, 1.3.22, 1.3.23, 1.3.29, 1.3.30, 1.3.31, 1.3.32, 1.3.33, 1.3.34,1.3.35, and 1.3.78 to for LID 25h.
20. Added subcases to tests 1.2.1-7, 1.2.10, 1.2.19, 1.2.47, and 1.8.1 to include validation for Feature Identifier 24h (Power Threshold) as per TP4199.
21. Added subcases to tests 1.2.1-7, 1.2.10, 1.2.19, 1.2.47, 1.8.1, and 1.9.1 to incorporate procedures and validation for Feature Identifier 25h (Power Measurement) as per TP4199.
22. New Group 13 Configurable Device Personality Tests, which test freeze/unfreeze, security personality, and authentication as per TP4163. Test 1.2.49 and 1.2.50 are also added to test Get/Set Feature functionality of specific personalities.

2026 July 21 (Version 26.0)

Carter Snay:

Tests with Status Changes:

1. Test Case 1.2.19 updated from (FYI) to (M)
2. Test Case 1.21.9 updated from (FYI) to (M)
3. Test Cases 2.12.1, 4-6 updated from (FYI) to (M)
4. Test Case 12.1.2 updated from (FYI) to (M)
5. Sub-test Case 1.2.1.21 updated from (FYI) to (M)

© 2026 UNH–IOL

6. Sub-test Case 1.2.2.21 updated from (FYI) to (M)
7. Sub-test Cases 1.2.4.14, 17, 21, 22 updated from (FYI) to (M)
8. Sub-test Cases 1.2.5.21, 27-29 updated from (FYI) to (M)
9. Sub-test Cases 1.2.10.14, 20 updated from (FYI) to (M)
10. Sub-test Cases 1.3.1.15, 26 updated from (FYI) to (M)
11. Sub-test Cases 1.3.10.10, 15 updated from (FYI) to (M)
12. Sub-test Case 1.3.22.10 updated from (FYI) to (M)
13. Sub-test Case 1.3.29.15 updated from (FYI) to (M)
14. Sub-test Case 1.3.31.15 updated from (FYI) to (M)
15. Sub-test Case 1.3.33.15 updated from (FYI) to (M)
16. Sub-test Cases 1.3.35.9, 15 updated from (FYI) to (M)
17. Sub-test Case 1.8.1.4 updated from (FYI) to (M)

New and Modified test cases:

1. Renamed test case 13.1.6 from “Unfreezing Personality with Unsupported Authentication Method Remains Frozen” to “Security Personality – Attribute Change Logged in Persistent Event Log”, since it shared a name with 13.1.5 despite testing that a Security Personality Attribute change is recorded in the Persistent Event Log, not personality freeze behavior.
2. Renamed test case 13.1.8 from “CDP Programmable Key – Pre-Existing Key” to “CDP Programmable Key – Key Not Programmed After Personality Switch”, since it shared a name with 13.1.7 despite testing that a previously programmed Host Key is not retained after switching away from and back to a personality.
3. Fixed subcases in tests 1.2.1-7, 1.2.10, 1.2.19, 1.2.47, and 1.8.1 which were incorrectly named “FID 25h Power Management” instead of “FID 25h Power Measurement” per TP4199.
4. Modified test case 6.2.8 to remove the recording of SMTR from the Get Log Page step and the irrelevant Get Features FID=24h (Power Threshold) step, moving the SMTR recording to the Get Features FID=25h step where PME and SMT are already recorded, and updated the Observable Results to reference the correct step.
5. Modified test case 1.1.58 by adding a new step 1 to set the current Power State to NPSS - 1 before Power State Descriptors are removed, renumbering the remaining procedure steps accordingly, and updating the repeat step and Observable Results 1-4 to reference the correct step numbers and to verify that the current Power State is decremented by however many Power States were removed by the most recent Set Features command, rather than always moving down by 1.
6. Modified test case 1.1.59 to update Observable Results 1–3:
    - Observable 1 updated to verify that for every repeated step 5 the previous Power State 0 Descriptor is updated to be the highest missing Power State Descriptor most recently added to the table and that the previous Power State 0 is offset by however many missing Power State Descriptors were added. Repeat for the rest of the preexisting Power State Descriptors.
    - Observable 2 updated to verify that for every repeated step 5 the Power Descriptor table is correctly ordered, such that each contiguous entry has a higher Maximum Power (MP * MXPS) value than the previous entry.
    - Observable 3 updated to verify for every repeated step 6 that the current Power State is incremented by however many Power States were added by the most recent Set Features command in step 4.
7. Modified test case 6.2.1 by adding a skip condition if OLEC is cleared to 0h in step 3 of the test procedure.
8. Modified test case 1.12.4 by changing changing step 4 to set the Interface field to 00b or 01b instead of 01b or 10b, and changed the scope field from from 2h to 4h.
9. Modified test case 13.1.17 by fixing issue with observable results copy/paste. New observable results added.
10. Modified test case 13.1.5 by fixing switching PERID with PERFS in step 1.
11. Modified test case 13.1.12 by changing observable result 2 to echeck for PMDSS Field to be set to 00h.
12. Modified test case 1.7.4 by changing an NVMe v2.1 check for observble result 2.

© 2026 UNH–IOL

13. Updated test 11.1.1 by removing Firmware Commit logic
14. Modified test case 1.1.26 to only check UIDREUSE when NGUID and EUI64 are both cleared to 0h, per Base Spec 2.3 §4.7.1, and fixed a data structure name error in step 2.
15. Modified test case 6.2.7 to add a missing Set Features command to stop power measurement before the final Get Log Page read, and simplified the Observable Results to match.
16. Modified test case 2.8.2 to also mark it not applicable when the Zoned Namespace Command Set is supported, since AWUN, AWUPF, and ACWU apply to ZNS namespaces as well as NVM.
17. Modified test case 1.2.4 to allow "Feature Not Changeable" as an acceptable response to the Set Features command in step 2b, and to use the current Namespace Identifier instead of NSID=0h in step 2c, checking for success or "Feature Not Namespace Specific" instead of comparing returned values.
18. Modified test cases 1.2.32 and 1.2.33 to account for the Extended Host Identifier (EXHID) bit, verifying the correct 64-bit or 128-bit Host Identifier length is returned, per ECN 132.
19. Added test case 1.2.51 to verify a Set Features command exceeding the 255 Host Metadata Element Descriptor limit is aborted with Invalid Field in Command, per ECN 132.
20. Added test 9.14 to verify that controllers supporting Flexible Data Placement correctly report the Reclaim Unit Handle Scope (RUHS) bit in the Feature Identifiers Supported and Effects log page (LID 12h) for FID 1Eh (FDP Events), aligning with the new scope reporting requirements introduced in TP4215.
21. Modified test 9.12 case 2 to verify that a Get Features command for Flexible Data Placement Events (FID 1Eh) is aborted with a status code of Invalid Field in Command when the NSID field is set to FFFFFFFFh, aligning with the Reclaim Unit Handle scope requirements introduced in TP4215.
22. New Group 14 Sanitize Per Namespace tests (14.1.1-8), which test namespace-scoped sanitize operations initiated via the Sanitize Namespace command (opcode 8Ch), including prohibited command handling while a namespace sanitize is in progress, NDE/OPC/SOS field reporting, the MNSOIP concurrency limit, NSID target validation, and Sanitize Namespace command option/rejection constraints, per TP4153.
23. Added test cases 1.3.88-92 for TP4153 Sanitize Per Namespace: Persistent Event Log Sanitize Start Event (Event Type 09h) and Sanitize Completion Event (Event Type 0Ah), Sanitize Status Log Page (LID 81h) behavior for an unallocated NSID, target-specific field validations (ETO/ETBE/ETODMM/ETBENMM/ETCENMM/MNSOIP and STNSID), and Sanitize Namespace Status List (LID 7Fh) log page contents.
24. Added subcase 1.7.7.1 to verify that EVNTSP is set to the NSID of the sanitization target when the Asynchronous Event Request command completes following a Sanitize Namespace command, per TP4153.
25. Modified test case 1.17.9 to add Sanitize Namespace Command (if supported) as a prohibited command aborted with Sanitize In Progress while a Sanitize command is executing, per TP4153.
26. Added subcase 1.17.10.1 to verify a Sanitize Namespace command targeting a write-protected namespace is aborted with Namespace is Write Protected, per TP4153.
27. Added test cases 1.17.31-35 for TP4153: Sanitize execution behavior with EMVS cleared, EMVS ignored when SANACT does not start a sanitize operation, Sanitize command option and rejection constraints, Sanitize Namespace interruption and Media Verification Canceled (MVCNCLD) tracking, and Sanitize command abort/restriction in the Media Verification state across migratable controllers.

© 2026 UNH–IOL

ACKNOWLEDGEMENTS

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

David Woolf	UNH Interoperability Labs
Raju Mishra	UNH Interoperability Labs
Neeraj Gill	UNH Interoperability Labs
Qian Liu	UNH Interoperability Labs
Kasra Dalvand	UNH Interoperability Labs
Benjamin Novak	UNH Interoperability Labs
Mike Bogochow	UNH Interoperability Labs
Jeff Hensel	UNH Interoperability Labs
Colin Dorsey	UNH Interoperability Labs
Tim Sheehan	UNH Interoperability Labs
Carter Snay	UNH Interoperability Labs
James Kahn	UNH Interoperability Labs
William Lee	UNH Interoperability Labs
Michael Crawley	UNH Interoperability Labs
Timofei Nikshych	UNH Interoperability Labs
Mitchell Tilton	UNH Interoperability Labs
Austin Snow	UNH Interoperability Labs
Liam Warren	UNH Interoperability Labs
Jae Butler	UNH Interoperability Labs
Parker Hertigan	UNH Interoperability Labs
Oliver Willis	UNH Interoperability Labs
Benjamin Svoboda	UNH Interoperability Labs

© 2026 UNH–IOL

INTRODUCTION

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

These tests are designed to determine if a product conforms to specifications defined in the NVMe NVM Command Set Specification Revision 1.0a specification, hereafter referred to as the "NVMe Specification". Successful completion of these tests provide a reasonable level of confidence that the Device Under Test (DUT) will function properly in many NVMe environments. The tests in this test suite are designed to validate the NVMe NVM Admin command set along with the NVM command set.

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.

© 2026 UNH–IOL

Observable Results

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

Possible Problems

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

© 2026 UNH–IOL

REFERENCES

The following documents are referenced in this text:

1. NVM Express NVM Command Set Specification 1.3 (July 31, 2026)
2. NVM Express Base Specification 2.4 (July 31, 2026)
3. NVM Express Management Interface Specification 2.2 (July 31, 2026)

© 2026 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 - For Your Information
IP - In Progress

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

OF – Test applies to NVMe-oF products

The following legacy numbering is applied to the test titles of each of the test cases to map the origin of the test case movement due to 2.0 refactoring efforts. If there is no change to the test case number then this field will be left off of the 2.0 refactored test case title.

NC – UNH-IOL NVMe Conformance Test Plan version 16.0, September 23, 2021
Legacy number : (PCIeC:x.x)

© 2026 UNH–IOL

Group 1: Admin Command Set

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

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

© 2026 UNH–IOL

Test 1.1 – Identify Command (OF, M)

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

References:

Old Ref: [1] NVMe Specification 5.15, NVMe v1.3 ECN 004a
NVM Express Base specification 2.0a : 5.17

Resource Requirements:

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

Last Modification: December 1, 2019

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

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

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

Test Setup: See Appendix A.

Case 1: CNS=00h Identify Namespace Data Structure (OF, M)

Test Procedure:

1. Retrieve a list of all active NSIDs using an Identify Command to CNS 02h.
2. For each namespace in the NVM subsystem, configure the NVMe Host to issue an Identify command specifying CNS value 00h to each active namespace in order to receive back an Identify Namespace data structure for the specified namespace.

Observable Results:

1. Verify that the requested data structure is posted to the memory buffer indicated in PRP Entry 1, PRP Entry 2, and Command Dword 10, and that a command completion queue entry is posted to the Admin Completion Queue.
2. If the specified namespace ID is inactive, verify that the data structure returned by the controller is zero filled.
3. Verify that all received responses have all Reserved fields set to 0h.
4. Verify '*n*' the Number of LBA Formats (NLBAF) set, and that any LBA Format Support descriptors beyond the NLBAF (i.e. LBAF*n*+*i*) are set to zero.
5. Verify that the NGUID or EUI64 field contains a globally unique namespace identifier, or a Namespace UUID is provided in the Namespace Identification Descriptor. If the controller is not able to provide a globally unique identifier in the NGUID or EUI64 field, then this field shall be cleared to 0h.
6. Verify that if the THINP bit in the NSFEAT field is cleared to '0h', that the value reported in the Namespace Capacity field that is equal to the Namespace Size;
7. Verify that if the THINP bit in the NSFEAT field is set to '1h', the DUT tracks the number of allocated blocks in the Namespace Utilization (NUSE) field.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

© 2026 UNH–IOL

Case 2: CNS=01h Identify Controller Data Structure (OF, M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command specifying CNS value 01h to the controller in order to receive back an Identify Controller data structure.
2. Repeat the previous step for all controllers in the domain.

Observable Results:

1. Verify that the requested data structure is posted to the memory buffer indicated in PRP Entry 1, PRP Entry 2, and Command Dword 10, and that a command completion queue entry is posted to the Admin Completion Queue.
2. Verify that all received responses have all Reserved fields set to 0h.
3. If the DUT claims support for NVMe Version 1.2 or higher, verify that the value reported for Version support in the CAP register (VS Offset 08h), matches the value reported in the Identify Controller Data Structure VER field (83:80) and is non zero.
4. Verify that any power state descriptors not supported (i.e. Power States beyond the number reported in the NPSS value (263) in the Identify Controller Data Structure) are set to 0h.
5. Verify that any values that are reported as ASCII strings (specifically Serial Number – SN, Model Number – MN, and Firmware Revision – FR) are left justified and padded with spaces (ASCII 20h character) to the right.
6. Verify that if the DUT claims to support NVMe Specification version 1.4 or later, that the CNTRLTYPE field in the Identify Controller Data Structure is not set to 0h.
7. Verify that if the NQN format does not include a UUID (i.e. it is the first type of NQN format) that the reverse domain name in the NQN field is not “org.nvmexpress”.
8. Verify that if FNA bit 3 is set to 1h, that FNA Bit 0 and 1 are cleared to 0h.
9. Verify that the same MAXDNA value is reported by all I/O Controllers in the Domain.
10. Verify that the same MAXCNA value is less than or equal to the MNAN field.
11. Verify that if the NODMMAS field is set to 00b, that the VER field indicates NVMe v1.3 or earlier, or bits 2:0 of the SANICAP field are set to 00b. Otherwise setting NODMMAS field to 00b is prohibited.

Case 3: CNS=02h Namespace List (OF, M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command specifying CNS value 02h and CDW1.NSID value 0 to the controller in order to receive back a Namespace List containing active namespace IDs.
2. For each namespace ID returned in the Namespace List, configure the host to issue an Identify command specifying CNS value 00h to the controller in order to receive back an Identify Namespace data structure for the specified namespace.

Observable Results:

1. Verify that the requested Namespace List is posted to the memory buffer indicated in PRP Entry 1, PRP Entry 2, and Command Dword 10, and that a command completion queue entry is posted to the Admin Completion Queue.
2. For each Identify command in step 2, verify that the returned data structure is not zero filled since each namespace ID in the Namespace List should be active.
3. Verify that all received responses have all Reserved fields set to 0h.

Case 4: CNS=03h Namespace Identification Descriptor (OF, M)

Test Procedure:

1. Check bit 43 of CAP.CSS.
2. Check the Identify Controller Data Structure to determine what version of the NVMe specification the product under test claims to support. If the product under test does not support NVMe v1.3 or higher, this test is not applicable.
3. Send an Identify Command with CNS=02h to determine all active namespaces in the NVM Subsystem.

© 2026 UNH–IOL

4. For each active namespace in the NVM subsystem, as determined in the previous step, configure the NVMe Host to issue an Identify command specifying CNS value 03h to the controller in order to receive back the Namespace Identification Descriptor data structure for the specified namespace.

Observable Results:

1. Verify that the command returns one or more Namespace Identification Descriptor structures that fit into the 4096 byte Identify payload.
2. Verify that the controller does not return multiple descriptors with the same Namespace Identification Descriptor Type (NIDT).
3. Verify that if NGUID and EUI64 are set to 0h in the Identify Namespace Data Structure, the Namespace Identification Descriptor reports a value of type 3.
4. If Bit 43 of CAP.CSS is set to 1h, verify that the DUT included the NID value in each Namespace Identification Descriptor returned and that only one descriptor was returned for each NIDT value.

Case 5: CNS=10h Namespace ID List (OF-FYI, M)

Test Procedure:

1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure OACS Bit 3 to determine if the product under test supports Namespace Management. If the product under test does not support Namespace Management, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 10h and a value of 0h for the Namespace Identifier in CDW1.NSID to the controller in order to receive back the Allocated Namespace ID List.

Observable Results:

1. Verify that the command returns the namespace list of allocated NSIDs in increasing order that are greater than the value specified in the Namespace Identifier in CDW1.NSID in the Identify Command with CNS=10h.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of “Command Sequence Error” on some later 2.0 Base drives.

Case 6: CNS=11h Identify Allocated Namespace Data Structure Allocated NSID (OF-FYI, M)

Test Procedure:

1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure OACS Bit 3 to determine if the product under test supports Namespace Management. If the product under test does not support Namespace Management, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 10h and a value of 0h for the Namespace Identifier in CDW1.NSID to the controller in order to receive back the Allocated Namespace ID List.
3. Configure the NVMe Host to issue an Identify command specifying CNS value 11h and a Namespace Identifier in CDW1.NSID that is a known allocated Namespace, if available, based on the response to the Identify Command tow CNS=10h in the previous step. If there is no allocated Namespace this test is not applicable.

Observable Results:

1. Verify that the command returns the Identify Namespace Data Structure for the specified allocated namespace.

Case 7: CNS=11h Identify Allocated Namespace Data Structure Unallocated NSID (OF-FYI, M)

Test Procedure:

1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure OACS Bit 3 to determine if the product under test supports Namespace Management. If the product under test does not support Namespace Management, this test is not applicable.

© 2026 UNH–IOL

2. Configure the NVMe Host to issue an Identify command specifying CNS value 10h and a value of 0h for the Namespace Identifier in CDW1.NSID to the controller in order to receive back the Allocated Namespace ID List.
3. Configure the NVMe Host to issue an Identify command specifying CNS value 11h and a Namespace Identifier in CDW1.NSID that is a known unallocated Namespace, if available, based on the response to the Identify Command tow CNS=10h in the previous step. If there is no unallocated Namespace this test is not applicable.

Observable Results:

1. Verify that the command returns a zero filled Identify Namespace Data Structure for the specified unallocated namespace.

Case 8: CNS=11h Identify Allocated Namespace Data Structure Invalid NSID (OF-FYI, M)

Test Procedure:

1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure OACS Bit 3 to determine if the product under test supports Namespace Management. If the product under test does not support Namespace Management, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 10h and a value of 0h for the Namespace Identifier in CDW1.NSID to the controller in order to receive back the Allocated Namespace ID List.
3. Configure the NVMe Host to issue an Identify command specifying CNS value 11h and an invalid Namespace Identifier in CDW1.NSID, if available. If there is no invalid NSID this test is not applicable.

Observable Results:

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

Case 9: CNS=12h Namespace Attached Controller List (OF-FYI, M)

Test Procedure:

1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure OACS Bit 3 to determine if the product under test supports Namespace Management. If the product under test does not support Namespace Management, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 10h and a value of 0h for the Namespace Identifier in CDW1.NSID to the controller in order to receive back the Allocated Namespace ID List.
3. Configure the NVMe Host to issue an Identify command specifying CNS value 12h and a valid value for Namespace Identifier in CDW1.NSID and Controller ID in CDW10.CNTID.

Observable Results:

1. Verify that the command completes with status success and returns a list of controller identifier that are attached to the namespace specified in the CDW1.NSID field.

Case 10: CNS=13h Controller List (OF-FYI, M)

Test Procedure:

1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure OACS Bit 3 to determine if the product under test supports Namespace Management. If the product under test does not support Namespace Management, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 13h and a valid value for Controller ID in CDW10.CNTID.

Observable Results:

1. Verify that the command completes with status success and returns a list of controller identifiers in the NVM subsystem that may or may not be attached to a namespace.

© 2026 UNH–IOL

Case 11: CNS=14h Primary Controller Data Structure (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure OACS Bit 7 to determine if the product under test supports Virtualization Enhancements. If the product under test does not support Virtualization Enhancements, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 14h and a valid value for Controller ID in CDW10.CNTID.

Observable Results:

1. Verify that the command completes with status success and returns the Primary Controller Capabilities Structure.

Case 12: CNS=15h Secondary Controller List (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure OACS Bit 7 to determine if the product under test supports Virtualization Enhancements. If the product under test does not support Virtualization Enhancements, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 15h and a valid value for Controller ID in CDW10.CNTID.

Observable Results:

1. Verify that the command completes with status success and returns the Secondary Controller List for Controller Identifiers greater than or equal to what was specified in the CDW10.CNTID field.

Case 13: Identify to reserved CNS Value (OF, M)

Test Procedure:

1. For each namespace in the NVM subsystem, configure the NVMe Host to issue an Identify command specifying a CNS of FFh.

Observable Results:

1. Verify that the command returns status ‘Invalid Field in Command’ (02h).

Case 14: CNS=16h Namespace Granularity List (OF-FYI, M)

Test Procedure:

1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure CTRATT Bit 7 to determine if the product under test supports Namespace Granularity. If the product under test does not support Namespace Granularity (Bit 7 = 1) this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 16h and a valid value for Controller ID in CDW10.CNTID.

Observable Results:

1. Verify that the Identify command for CNS=16h completes successfully and includes a properly formatted Namespace Granularity List.
2. Verify that the namespace granularity descriptors with an index greater than the value in the Number of Descriptors field are cleared to 0h.

Case 15: CNS=17h UUID List (OF-FYI, M)

Test Procedure:

© 2026 UNH–IOL

1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure CTRATT Bit 9 to determine if the product under test supports UUID List. If the product under test does not support UUID List (Bit 9 = 1) this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 17h.

Observable Results:

1. Verify that the Identify command for CNS=17h completes successfully and returns a properly formatted UUID List.
2. Verify that The UUID List:
    a. Contains at least one valid UUID.
    b. The UUID 1 field contains a non-zero value.
    c. The UUID 127 field is cleared to 0h
    d. The last UUID entry is cleared to 0h to indicate the end of the UUID List.

Case 16: CSI Field (OF-FYI, M)

Test Procedure:

1. Perform an Identify Command for each of the following supported CNS values: 00h, 01h, 02h, 03h, 04h, 10h, 11h, 12h, 13h, 14h, 15h, 16h, 17h, 1Ch.

Observable Results:

1. Verify that for the CNS values which are supported by the DUT that the CDW11.CSI field was set to 0h.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.1.16 and expect the same observable results.

1. **CNS 00h**
2. **CNS 01h**
3. **CNS 02h**
4. **CNS 03h**
5. **CNS 04h**
6. **CNS 05h**
7. **CNS 06h**
8. **CNS 07h**
9. **CNS 08h**
10. **CNS 10h**
11. **CNS 11h**
12. **CNS 12h**
13. **CNS 13h**
14. **CNS 14h**
15. **CNS 15h**
16. **CNS 16h**
17. **CNS 17h**
18. **CNS 18h**
19. **CNS 19h**
20. **CNS 1Ah**
21. **CNS 1Bh**
22. **CNS 1Ch**

Case 17: CNS=1Ch I/O Command Set Data Structure (OF-FYI, FYI)

Test Procedure:

1. Check Bit 43 of CAP.CSS. If this bit is set to 0h then this test is not applicable.

© 2026 UNH–IOL

2. Perform an Identify Command for CNS=1Ch.

Observable Results:

1. Verify that the Identify Command completes successfully, and the expected Data Structure is returned.

Case 18: NVM Command Set Not Supported (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Command for CNS=07h to obtain the list of Active Namespace ID's associated with a specific command set.
2. For all namespaces that are not associated with the NVM Command set perform the following commands:
    a. Identify Command with CNS=00h.
    b. Identify Command with CNS=11h.
    c. Identify Command with CNS=16h.

Observable Results:

1. Verify that the Identify Command with CNS=00h completes with status 'Invalid I/O Command Set' (2Ch).
2. Verify that the Identify Command with CNS=11h completes with status 'Invalid I/O Command Set' (2Ch).
3. Verify that the Identify Command with CNS=16h completes with status 'Invalid Field in Command' (02h).

Case 19: CNS=05h (OF-FYI, M)

Test Procedure:

1. Perform an Identify Command for CNS=07h to obtain the list of Active Namespace ID's associated with a specific command set.
2. Perform an Identify Command for CNS=05h to a given namespace using a CSI value that indicates an I/O Command Set which is not supported by the namespace.

Observable Results:

1. Verify that the Identify Command with CNS=05h completes with status 'Invalid Field in Command' (02h).

Case 20: CNS=05h Namespace Management Not Supported (OF-FYI, M)

Test Procedure:

1. Check the NVME Specification version supported by DUT using the value reported in the Identify Controller Data Structure VER field (83:80). If the DUT supports v1.4 or lower skip this test.
2. Check Bit 3 of the OACS field to determine if the DUT supports Namespace Management. If the DUT does support Namespace Management, then this test is not applicable.
3. Perform an Identify Command for CNS=05h with NSID=FFFFFFFFh.

Observable Results:

1. Verify that the Identify Command with CNS=05h completes with status 'Invalid Namespace or Format' (0Bh).

Possible Problems: The controller status response is suggested in the 2.0 specification.

Case 21: CNS=06h (OF-FYI, M)

Test Procedure:

1. Perform an Identify Command for CNS=07h to obtain the list of Active Namespace ID's associated with a specific command set.
2. Perform an Identify Command for CNS=06h to a given namespace using a CSI value that indicates an I/O Command Set which is not supported by that namespace.

Observable Results:

1. Verify that the Identify Command with CNS=06h completes with status 'Invalid Field in Command' (02h).

© 2026 UNH–IOL

Case 22: CNS=1Bh (OF-FYI, M)

Test Procedure:

1. Perform an Identify Command for CNS=07h to obtain the list of Active Namespace ID's associated with a specific command set.
2. Perform an Identify Command for CNS=1Bh to a given namespace using a CSI value that indicates an I/O Command Set which is not supported by the by that namespace.

Observable Results:

1. Verify that the Identify Command with CNS=1Bh completes with status 'Invalid Field in Command' (02h).

Case 23: CNS=1Bh, Invalid NSID (OF-FYI, M)

Test Procedure:

1. Perform an Identify Command for CNS=07h to obtain the list of Active Namespace ID's associated with a specific command set.
2. Perform an Identify Command for CNS=1Bh to an invalid namespace using a CSI value that indicates an I/O Command Set which is supported by the by the DUT.

Observable Results:

1. Verify that the Identify Command with CNS=1Bh completes with status 'Invalid Namespace or Format' (0Bh).

Case 24: CNS=1Ch (OF-FYI, M)

Test Procedure:

1. Check bit 6 of the CAP.CSS. If this bit is set to 0h this test is not applicable.
2. Perform an Identify Command for CNS=1Ch.

Observable Results:

1. Verify that the DUT returns the I/O Command Set Data Structure and that the first I/O Command Set Combination 'N' field that is set to 0h, (N from 1 to 511) is the last one with a non-zero value, and all subsequent I/O Command Set Combinations have a value of 0h.

Case 25: CNS=08h Namespace Management Not Supported (OF-FYI, FYI)

Test Procedure:

1. Check the NVME Specification version supported by DUT using the value reported in the Identify Controller Data Structure VER field (83:80). If the DUT supports v1.4 or lower skip this test.
2. Configure the Testing Station acting as a host to Check Bit 3 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports Namespace Management. If Bit 3 is not set to 0h, this test is not applicable.
3. Perform an I/O Command Set Independent Identify for CNS=08h using a NSID =FFFFFFFFh.

Observable Results:

1. Verify that the DUT returns a status code of 'Invalid Namespace or Format' (0Bh).

Possible Problems: The controller status response is suggested in the 2.0 specification.

Case 26: CNS=08h UIDREUSE (OF-FYI, FYI)

Test Procedure:

1. Retrieve a list of all active NSIDs using an Identify Command to CNS 02h.
2. For each namespace in the NVM subsystem, configure the NVMe Host to issue an Identify command specifying CNS value 00h to each active namespace in order to receive back the Identify Namespace data structure

© 2026 UNH–IOL

for the specified namespace. Record the NGUID and EUI64 fields. If the NGUID or EUI64 field is not cleared to 0h for a namespace, this test is not applicable to that namespace.

3. For each namespace identified in step 2 with both NGUID and EUI64 cleared to 0h, configure the NVMe Host to issue an I/O Command Set Independent Identify command specifying CNS value 08h to that namespace.

Observable Results:

1. Verify that the DUT returns bit 3 (UIDREUSE) of the NSFEAT field in the I/O Command Set Independent Identify command and it is cleared to zero.

Case 27: CNS=08h Reservation Capabilities (OF-FYI, FYI)

Test Procedure:

1. Retrieve a list of all active NSIDs using an Identify Command to CNS 02h.
2. For each namespace in the NVM subsystem, configure the NVMe Host to issue an Identify command specifying CNS value 00h to each active namespace in order to receive back an I/O Command Set Independent Identify data structure for the specified namespace.
3. For each namespace in the NVM Subsystem, configure the NVMe Host to issue an I/O Command Set Independent Identify command specifying CNS value 08h to each active namespace.

Observable Results:

1. Verify that the DUT returns bit 7 of the RESCAP field in the I/O Command Set Independent Identify Command and it is set to ‘1’ if the Controller support version 1.3 or later as indicated in the Version Register.

Case 28: CNS=08h Format Progress Indicator (OF-FYI, M)

Test Procedure:

1. Retrieve a list of all active NSIDs using an Identify Command to CNS 02h.
2. For each namespace in the NVM subsystem, configure the NVMe Host to issue an Identify command specifying CNS value 00h to each active namespace in order to receive back an I/O Command Set Independent Identify data structure for the specified namespace.
3. For each namespace in the NVM Subsystem, configure the NVMe Host to issue an I/O Command Set Independent Identify command specifying CNS value 08h to each active namespace.

Observable Results:

1. Verify that the DUT returns bit 7 of the FPI field in the I/O Command Set Independent Identify Command and it is set to ‘0’, then bits 6:0 in this field must be cleared to 0h.

Case 29: CNS=08h ADA Group Identifier Cleared (OF-FYI, M)

Test Procedure:

1. Retrieve a list of all active NSIDs using an Identify Command to CNS 02h.
2. For each namespace in the NVM subsystem, configure the NVMe Host to issue an Identify command specifying CNS value 00h to each active namespace in order to receive back an I/O Command Set Independent Identify data structure for the specified namespace.
3. If bit 3 in the CMIC field in the Identify Controller structure is set to ‘1’ then skip this test.
4. For each namespace in the NVM Subsystem, configure the NVMe Host to issue an I/O Command Set Independent Identify command specifying CNS value 08h to each active namespace.

Observable Results:

1. Verify that the DUT returns the ANAGRPID field in the I/O Command Set Independent Identify Command and it is set to 0h.

© 2026 UNH–IOL

Case 30: CNS=08h ADA Group Identifier Valid (OF-FYI, FYI)

Test Procedure:

1. Retrieve a list of all active NSIDs using an Identify Command to CNS 02h.
2. For each namespace in the NVM subsystem, configure the NVMe Host to issue an Identify command specifying CNS value 00h to each active namespace in order to receive back an I/O Command Set Independent Identify data structure for the specified namespace.
3. For each namespace in the NVM Subsystem, configure the NVMe Host to issue an I/O Command Set Independent Identify command specifying CNS value 08h to each active namespace.

Observable Results:

1. Verify that the DUT returns a valid ANAGRPID

Case 31: CNS=08h ADA Group Identifier Change Notice (OF-FYI, FYI)

Test Procedure:

1. Retrieve a list of all active NSIDs using an Identify Command to CNS 02h.
2. For each namespace in the NVM subsystem, configure the NVMe Host to issue an Identify command specifying CNS value 00h to each active namespace in order to receive back an I/O Command Set Independent Identify data structure for the specified namespace.
3. Modify the ANAGRPID value associated with the namespace.
4. For each namespace in the NVM Subsystem, configure the NVMe Host to issue an I/O Command Set Independent Identify command specifying CNS value 08h to each active namespace.

Observable Results:

1. Verify that the DUT's ANAGRPID value has changed and verify that the DUT sent an Asymmetric Namespace Access Change event notice

Case 32: CNS=08h Namespace Attributes (OF-FYI, FYI)

Test Procedure:

1. Retrieve a list of all active NSIDs using an Identify Command to CNS 02h.
2. For each namespace in the NVM subsystem, configure the NVMe Host to issue an Identify command specifying CNS value 00h to each active namespace in order to receive back an I/O Command Set Independent Identify data structure for the specified namespace.
3. For each namespace in the NVM Subsystem, configure the NVMe Host to issue an I/O Command Set Independent Identify command specifying CNS value 08h to each active namespace.
4. Perform a Set Feature operation to set Namespace Write Protection Config to "Write Protect" (001b).
5. Perform a Write operation to write a known data pattern (i.e. AAAAh) to the Namespace.

Observable Results:

1. Verify that the DUT returns bit 0 in the NSATTR field of the I/O Command Independent Identify command.
2. Verify that the Write operation attempted in Step 7 completed with a status 'Namespace is Write Protected' (20h).

Case 33: CNS=08h NVM Set Identifier (OF-FYI, M)

Test Procedure:

1. Retrieve a list of all active NSIDs using an Identify Command to CNS 02h.
2. For each namespace in the NVM subsystem, configure the NVMe Host to issue an Identify command specifying CNS value 00h to each active namespace in order to receive back an I/O Command Set Independent Identify data structure for the specified namespace.
3. For each namespace in the NVM Subsystem, configure the NVMe Host to issue an I/O Command Set Independent Identify command specifying CNS value 08h to each active namespace.

© 2026 UNH–IOL

4. Configure the NVMe Host to query the Identifier Controller Data Structure, Bit 2 (NVM Sets) of the CTRATT field of the Identify Controller structure.

Observable Results:

1. Verify that the DUT's NVMSETID value is cleared to 0h, if NVM sets are not supported by the controller

Case 34: CNS=08h Endurance Group Identifier (OF-FYI, M)

Test Procedure:

1. Retrieve a list of all active NSIDs using an Identify Command to CNS 02h.
2. For each namespace in the NVM subsystem, configure the NVMe Host to issue an Identify command specifying CNS value 00h to each active namespace in order to receive back an I/O Command Set Independent Identify data structure for the specified namespace.
3. For each namespace in the NVM Subsystem, configure the NVMe Host to issue an I/O Command Set Independent Identify command specifying CNS value 08h to each active namespace.
4. Configure the NVMe Host to query the Identifier Controller Data Structure, Bit 4 (Endurance Groups) of the CTRATT field.

Observable Results:

1. Verify that the DUT's ENDGID value is cleared to 0h, if Endurance Groups are not supported by the controller

Case 35: CNS=04h NVM Set List, No Zero Entry (OF-FYI, M)

Test Procedure:

1. Check Bit 2 (NVM Sets) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is not set to 1h, then this test is not applicable.
2. Check Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure. If Bit 4 is not set to 1h, then this test is not applicable.
3. Perform an Identify NVM Set List (CNS=04h).

Observable Results:

1. Verify that none of the NVM Set Attributes Entry data areas are cleared to '0h'.

Case 36: CNS=19h Endurance Group, ENDGID > ENDGIDMAX (OF-FYI, FYI)

Test Procedure:

1. Check Bit 12 (Variable Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is not set to 1h, then this test is not applicable.
2. Check Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure. If Bit 4 is not set to 1h, then this test is not applicable.
3. Perform an Identify Controller Data Structure Command (CNS=01h) and retrieve the Endurance Group Identifier Maximum (ENDGIDMAX).
4. Configure the NVMe Host to issue an Identify command specifying CNS value 19h and setting the CNS Specific Identifier (Dword 11) to ENDGIDMAX+1.

Observable Results:

1. Verify that Identify command completes successfully and that the Endurance Group List does not contain any Endurance Group Identifiers.

Case 37: CNS=00h Identify Namespace Data Structure, with Reported field (OF-FYI, M)

Test Procedure:

1. Check the NVME Specification version supported by DUT using the value reported in the Identify Controller Data Structure VER field (83:80). If the DUT supports v1.4 or lower skip this test.

© 2026 UNH–IOL

2. Retrieve a list of all active NSIDs using an Identify Command to CNS 02h.
3. Configure the Testing Station acting as a host to Check Bit 3 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports Namespace Management. If Bit 3 is set to 0h, this test is not applicable.
4. Configure the NVMe Host to issue an Identify command specifying CNS value 00h, a CSI value of 00h, with an NSID=FFFFFFFFh, receive back an Identify Namespace data structure for the specified namespace with the Reported field entries.

Observable Results:

1. Verify that the requested data structure returns successfully in each step and that the Identify command specifying CNS=00h with an NSID=FFFFFFFFh returns values in the following fields: NLBAF, MC, DPC, NMIC, NULBAF, and all LBA Format Support structures. That all other fields are cleared to 0h.
2. Verify that all active namespaces have the same settings in their Identify CNS 00h data structure as the fields mentioned in Observable 1.

Case 38: CNS=05h Identify Namespace Data Structure, with Reported field (OF-FYI, M)

Test Procedure:

1. Check the NVME Specification version supported by DUT using the value reported in the Identify Controller Data Structure VER field (83:80). If the DUT supports v1.4 or lower skip this test.
2. Configure the Testing Station acting as a host to Check Bit 3 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports Namespace Management. If Bit 3 is set to 0h, this test is not applicable.
3. Retrieve a list of all active NSIDs using an Identify Command to CNS 02h.
4. Configure the NVMe Host to issue an Identify command specifying CNS value 05h, with a CSI value 00h, with an NSID=FFFFFFFFh, to receive back an Identify Namespace data structure.

Observable Results:

1. Verify that the requested data structure returns successfully in each step and that the Identify command specifying CNS=05h with an NSID=FFFFFFFFh returns values in the following fields: PIC, and all Extended LBA Format Support structures. That all other fields are cleared to 0h.
2. Verify that all active namespaces have the same settings in their Identify CNS 05h data structure as the fields mentioned in Observable 1.

Case 39: CNS=09h Namespace Capabilities, NVM Command Set (FYI, FYI)

Test Procedure:

1. Check the NVME Specification version supported by DUT using the value reported in the Identify Controller Data Structure VER field (83:80). If the DUT supports v1.4 or lower skip this test.
2. Configure the Testing Station acting as a host to Check Bit 3 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports Namespace Management. If Bit 3 is set to 0h, this test is not applicable.
3. Retrieve a list of all active NSIDs using an Identify Command to CNS 02h.
4. Configure the NVMe Host to issue an Identify command specifying CNS value 00h, with a CSI value of 00h, with an NSID=FFFFFFFFh, to receive back an Identify Namespace data structure for the specified namespace with the Reported field entries. Select one of the valid LBA Formats in the returned structure.
5. Configure the NVMe Host to issue an Identify command specifying CNS value 09h, with a CSI value of 00h, with CNS Specific Identifier field set to a valid Format Index from step 3.

Observable Results:

1. Verify that the requested data structure returns successfully in each step and that the Identify command specifying CNS=00 returns values in the following fields: NLBAF, MC, DPC, NMIC, NULBAF, and LBA Format Support Structures.

© 2026 UNH–IOL

2. Verify that all active namespaces have the same settings in their Identify CNS 09h data structure as the fields mentioned in Observable 1.

Case 40: CNS=0Ah, Namespace Capabilities, IO Command Set (FYI, FYI)

Test Procedure:

1. Check the NVME Specification version supported by DUT using the value reported in the Identify Controller Data Structure VER field (83:80). If the DUT supports v1.4 or lower skip this test.
2. Configure the Testing Station acting as a host to Check Bit 3 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports Namespace Management. If Bit 3 is set to 0h, this test is not applicable.
3. Retrieve a list of all active NSIDs using an Identify Command to CNS 02h.
4. Configure the NVMe Host to issue an Identify command specifying CNS value 05h, with a CSI value of 00h, with an NSID=FFFFFFFFh, to receive back an Identify Namespace data structure for the specified namespace with the Reported field entries. Select one of the valid LBA Formats in the returned structure.
5. Configure the NVMe Host to issue an Identify command specifying CNS value 0Ah, with a CSI value of 00h, with CNS Specific Identifier set to the valid LBA Format Index from step 3.

Observable Results:

1. Verify that the requested data structure returns successfully in each step and that the Identify command specifying CNS=0A returns values in the following fields: PIC, and Extended LBA Format Support structures.
2. Verify that all active namespaces have the same settings in their Identify CNS 0Ah data structure as the fields mentioned in Observable 1.

Case 41: CNS=08h Identify Namespace Data Structure, with Reported field (OF-FYI, M)

Test Procedure:

1. Check the NVME Specification version supported by DUT using the value reported in the Identify Controller Data Structure VER field (83:80). If the DUT supports v1.4 or lower skip this test.
2. Configure the Testing Station acting as a host to Check Bit 3 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports Namespace Management. If Bit 3 is set to 0h, this test is not applicable.
3. Retrieve a list of all active NSIDs using an Identify Command to CNS 02h.
4. Configure the NVMe Host to issue an Identify command specifying CNS value 08h, with a CSI value 00h, with an NSID=FFFFFFFFh, receive the Identify Namespace data structure for the specified namespace with the Reported field entries.
5. Configure the NVMe Host to issue an Identify command specifying CNS value 08h, with a CSI value 00h, with an NSID=with each active NSID from step 3, receive the Identify Namespace data structure for the specified namespace.

Observable Results:

1. Verify that the requested data structure returns successfully in each step and that the Identify command specifying CNS=08h with an NSID=FFFFFFFFh returns values in the following fields: NMIC and Key Per I/O Status contain values. That all other fields are cleared to 0h.
2. Verify that all active namespaces have the same settings in their Identify CNS 08h data structure as the fields mentioned in Observable 1.

Case 42: CNS=18h Domain List (FYI, FYI)

Test Procedure:

1. Check Bit 11 (Variable Capacity Management) and Bit 12 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If both Bits 11 & 12 are cleared to ‘0’, skip this test.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 18h, using the DWord11 equal to ‘0’.

© 2026 UNH–IOL

Observable Results:

1. Verify that the requested data structure returns successfully and list out its contents.

Case 43: CNS=19h Endurance Group List (FYI, FYI)

Test Procedure:

1. Check Bit 12 (Variable Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 12 is cleared to '0', skip this test.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 19h, using the Endurance Group Identifier equal to '0'.

Observable Results:

1. Verify that the requested data structure returns successfully and list out its contents.

Possible Problems: None known.

Case 44: CNS=06h, CSI=00h VERSION (OF-FYI, M)

Test Procedure:

1. Check the specification version supported by the DUT. If the DUT supports v2.0 or higher this test is applicable, otherwise this test is not applicable.
2. Perform an Identify Command for CNS=06h to a given namespace using a CSI value of 00h.

Observable Results:

1. Verify that the Identify Command completes successfully and that the VERSION field (23:20) contains correct MJR, MNR and TER fields that correlate with a published NVM Command Set Specification version (e.g. 1:0:0, 1:1:0, 1:2:0).

Case 45: CNS=11h, NSID=FFFFFFFFh (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify Namespace command with the NSID being set to 0xFFFFFFFF and CNS set to 11h.

Observable Results:

1. Verify Identify Namespace command aborts with status Invalid Namespace or Format (0Bh).

Possible Problems: It is possible that if Namespace Management is not supported that CNS 11h will not be supported.

Case 46: CNS=08h Identify Namespace Data Structure, with Non-Storage Namespace-NVM-SETID (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify command for CNS=01h, then check the Identify Controller Data Structure OACS Bit 3 to determine if the product under test supports Namespace Management. If the product under test does not support Namespace Management, this test is not applicable.
2. Perform an I/O Command Set Independent Identify for CNS=08h specifying an NSID for a Namespace that is known to not contain formatted storage if available. If there is no namespace that does not contain unformatted storage then this test is not applicable.

Observable Results:

1. Verify that the NVMSETID field of the returned Identify Namespace Data Structure is cleared to 0h.

© 2026 UNH–IOL

Case 47: CNS=08h Identify Namespace Data Structure, with Non-Storage Namespace-ENDGID (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify command for CNS=01h, then check the Identify Controller Data Structure OACS Bit 3 to determine if the product under test supports Namespace Management. If the product under test does not support Namespace Management, this test is not applicable.
2. Perform an I/O Command Set Independent Identify for CNS=08h specifying an NSID for a Namespace that is known to not contain formatted storage if available. If there is no namespace that does not contain unformatted storage then this test is not applicable.

Observable Results:

1. Verify that the ENDGID field of the returned Identify Namespace Data Structure is cleared to 0h.

Case 48: CNS=1Fh Allocated Namespace Data Structure Allocated NSID (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify command for CNS=01h, then check the Identify Controller Data Structure OACS Bit 3 to determine if the product under test supports Namespace Management. If the product under test does not support Namespace Management, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 10h and a value of 0h for the Namespace Identifier in CDW1.NSID to the controller in order to receive back the Allocated Namespace ID List.
3. Configure the NVMe host to Issue an Identify command specifying CNS value 1Fh and a Namespace Identifier in CDW1.NSID that is a known allocated Namespace, if available, based on the response to the Identify Command.

Observable Results:

1. Verify that the command returns a valid Identify Namespace Data Structure for the specified allocated namespace.
2. If the Identify command in step 3 completes with a status of ‘Invalid Field in Command’ (02h) and the DUT reports NVMe Base version 1.4 or less, then this test is not applicable.

Case 49: CNS=1Fh Allocated Namespace Data Structure Unallocated NSID (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Command for CNS=01h, then check the Identify Controller Data Structure OACS bit 3 to determine if the device under test supports Namespace Management. If the device under test does not support Namespace Management, this test is not applicable.
2. Configure the NVMe Host to issue an Identify Command specifying CNS value 10h and a value of 0h for the Namespace Identifier in CDW1.NSID to the controller in order to receive back the Allocated Namespace ID List.
3. Configure the NVMe Host to issue an Identify Command Specifying CNS value 1Fh and a Namespace Identifier in CDW1.NSID that is a known unallocated Namespace, if available, based on the response to the Identify Command to CNS=10h in the previous step. If there is no unallocated Namespace this test is not applicable.

Observable Results:

1. Verify that the command returns a zero filled Identify Namespace Data Structure for the specified unallocated namespace.
2. If the Identify command in step 3 completes with a status of ‘Invalid Field in Command’ (02h) and the DUT reports NVMe Base version 1.4 or less, then this test is not applicable.

© 2026 UNH–IOL

Case 50: CNS=1Fh Invalid NSID (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Command for CNS=07h to obtain the list of Active Namespace ID's associated with a specific command set.
2. Perform an Identify Command for CNS=1Fh to an invalid namespace using a CSI value that indicates an I/O command set which is supported by the DUT.

Observable Results:

1. Verify that the Identify Command with CNS=1Fh aborts with status 'Invalid Namespace or Format' (0Bh).

Case 51: CNS=1Fh NSID=FFFFFFFFh (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify Namespace command with the NSID being set to FFFFFFFFh and CNS=1Fh.

Observable Results:

1. Verify that the Identify Namespace Command aborts with status 'Invalid Namespace or Format' (0Bh).
2. If the Identify command in step 1 completes with a status of 'Invalid Field in Command' (02h) and the DUT reports NVMe Base version 1.4 or less, then this test is not applicable.

Case 52: CNS=1Bh Identify Namespace Data Structure Allocated NSID Detached (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If the DUT claims support for a version less than 2.0 or OACS bit 3 is cleared to 0h, then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command with CNS 03h to obtain the CSI value of the target namespace.
3. Configure the NVMe Host to issue an Identify command to obtain the I/O Command Set Specific Identify Namespace data structure, with CNS 05h and CSI set to the value discovered in step 2.
4. Configure the NVMe Host to issue a Namespace Attachment command to detach a namespace from the controller and record its NSID.
5. Configure the NVMe Host to issue an Identify command to obtain the I/O Command Set specific Identify Namespace data structure for the specified allocated NSID detached in step 3, with CNS 1Bh and CSI set to the value discovered in step 2.
6. Configure the NVMe Host to issue a Namespace Attachment command to attach the namespace detached in step 3.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the data structures returned in step 3 and 5 are identical.

Case 53: Media Verification Support: CNS=01h BES=0b and CES= 0b then VERS=0b (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure.

Observable Results:

1. Verify that the Identify command completed successfully.

© 2026 UNH–IOL

2. Verify that if Block Erase Support (BES) and Crypto Erase Support (CES) are both cleared to 0h, then Verification Support (VERS) is also cleared to 0h.

Case 54: Identify I/O Command Set Specific WZSL Not Supported (OF-FYI, FYI)

Test Procedure:

1. Check the WZSL field in the I/O Command Set Specific Identify Controller Data Structure (CNS 06h, CSI 00h). If the WZSL field is 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 supported, then this test is not applicable

Observable Results:

1. Verify that bits 11 and 12 of ONCS are cleared to 0h.

Case 55: Maximum Unlimited Power (MUP) Field is set to 0 when the Power Limit Support (PLS) Field is set to 0 (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to send an Identify Command Specifying CNS value 01h to the Controller in order to receive back an Identify Controller Data Structure. If the Power Limit Support (PLS) bit of the Controller Attributes field in the returned Identify Controller Data Structure is set to 0h, this test is not applicable.

Observable Results:

1. Verify that the Maximum Unlimited Power (MUP) field is set to 0h.

Case 56: Maximum Unlimited Power (MUP) and Maximum Unlimited Power Scale (MUPS) Field Indicate Maximum Power (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to send an Identify Command Specifying CNS value 01h to the Controller in order to receive back an Identify Controller Data Structure. If the Power Limit Support (PLS) bit of the Controller Attributes field in the Identify Controller Data Structure is not set to 1h, this test is not applicable.
2. Configure the NVMe host to send a Set Features command to the DUT for FID 23h with a Power Limit Value (PLV) of 0.
3. Configure the NVMe Host to send an Identify Command Specifying CNS value 01h to the Controller in order to receive back an Identify Controller Data Structure.

Observable Results:

1. Verify that the product of Maximum Unlimited Power Scale (MUPS) field and the Maximum Unlimited Power (MUP) field from the Identify Controller Data Structure returned in step 3 is equal to the product of the Max Power Scale (MXPS) and Maximum Power (MP) Field of the PSD0 field in the Identify Controller Data Structure.

Case 57: Power Limit Scale Reserved Values (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to send an Identify Command Specifying CNS value 01h to the Controller in order to receive back an Identify Controller Data Structure. If the Power Limit Support (PLS) bit of the Controller Attributes field in the Identify Controller Data Structure is not set to 1h, this test is not applicable.
2. Configure the NVMe host to send a Set Features command to the DUT for FID 23h with a Power Limit Scale (PLS) value of 00b.
3. Configure the NVMe Host to send a Set Features command to the DUT for FID 23h with a Power Limit Scale (PLS) value of 11b.
4. Configure the NVMe Host to send a Get Features command to the DUT for FID 23h.

© 2026 UNH–IOL

Observable Results:

1. Verify that Set Features commands issued in steps 2 and 3 are aborted with the status code of 'Invalid Field in Command' (02h).
2. Verify that the Get Feature command returned in step 4 does not report a reserved Power Limit Scale (PLS) value.

Case 58: Invalid Power Limit when Power Limit Value Lower than all Maximum Power Values (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to send a Set Features command for FID 02h with a PS set to NPSS - 1.
2. Configure the NVMe Host to send an Identify Command Specifying CNS value 01h to the Controller in order to receive back an Identify Controller Data Structure. If the Power Limit Support (PLS) bit of the Controller Attributes field in the Identify Controller Data Structure is not set to 1h, this test is not applicable.
3. Configure the NVMe Host to send a Set Feature command for FID 23h with a PLV set to 0h.
4. Configure the NVMe Host to send an Identify Command Specifying CNS value 01h to the Controller in order to receive back an Identify Controller Data Structure. Record all the Power State Descriptors.
5. Configure the NVMe Host to send a Set Features command such that the product of the Power Level Value (PLV) and Power Level Scale (PLS) is lower than the product of the Maximum Power (MP) value and the Max Power Scale (MXPS) of Power State Descriptor 0.
6. Configure the NVMe Host to send an Identify Command Specifying CNS value 01h to the Controller in order to receive back an Identify Controller Data Structure. Record all the Power State Descriptors.
7. Configure the NVMe Host to send a Get Feature for FID 02h.
8. Configure the NVMe Host to send a Get Feature for FID 0Ch.
9. Repeat steps 5, 6, 7, and 8 for the new Power State Descriptor 0 until all remaining Power State Descriptors have matching Maximum Power (MP) values when multiplied by their Max Power Scale (MXPS) values.

Observable Results:

1. Verify that for every repeated step 6 that the Previous Power State 0 Descriptor is removed from the Power State Descriptor table.
2. Verify that for every repeated step 6 that the Power Descriptor table is correctly ordered, such that each contiguous entry in the Power State Descriptor table has a higher Maximum Power value when multiplied by its Max Power Scale (MXPS) value than the previous entry.
3. Verify for every repeated step 7 that the current Power State is decremented by however many Power States were removed by the most recent Set Features command in step 5.
4. Verify for every repeated step 8 that the Autonomous Power State Transition Data Structure matches the Power State Descriptor table returned in observable 1.

Case 59: Power States Added when Power Level Value is higher than MP (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to send an Identify Command Specifying CNS value 01h to the Controller in order to receive back an Identify Controller Data Structure. If the Power Limit Support (PLS) bit of the Controller Attributes field in the Identify Controller Data Structure is not set to 1h, this test is not applicable.
2. Configure the NVMe Host to send a Set Feature for FID 23h command with a Power Level Value (PLV) that, when multiplied by the Power Level Scale (PLS) value, is equal to the lowest Maximum Power (MP) value in the Power State Descriptor Table when the MP is multiplied by its Maximum Power Scale value (MXPS).
3. Configure the NVMe Host to send an Identify Command Specifying CNS value 01h to the Controller in order to receive back an Identify Controller Data Structure. Record all the Power State Descriptors.
4. Configure the NVMe Host to send a Set Features command such that the product of the Power Level Value (PLV) and Power Level Scale (PLS) is equal to the product of the Maximum Power (MP) value and the Max Power Scale (MXPS) of the next supported but not enabled Power Descriptor.
5. Configure the NVMe Host to send an Identify Command Specifying CNS value 01h to the Controller in order to receive back an Identify Controller Data Structure. Record all the Power State Descriptors.

© 2026 UNH–IOL

6. Configure the NVMe Host to send a Get Feature for FID 02h.
7. Configure the NVMe Host to send a Get Feature for FID 0Ch.
8. Repeat steps 4, 5, 6, and 7 for the new Power State Descriptor 0 until all Power State Descriptors are reenabled.

Observable Results:

1. Verify that for every repeated step 5 that the previous Power State 0 Descriptor is updated to be the highest missing Power State Descriptor that was most recently added to the table and that the previous Power State 0 is now offset by however many missing Power State Descriptors were added. Repeat this check for the rest of the preexisting Power State Descriptors.
2. Verify that for every repeated step 5 that the Power Descriptor table is correctly ordered, such that each contiguous entry in the Power State Descriptor table has a higher Maximum Power value when multiplied by its Max Power Scale (MXPS) value than the previous entry.
3. Verify for every repeated step 6 that the current Power State is incremented by however many Power States were added by the most recent Set Features command in step 4.
4. Verify for every repeated step 7 that the Autonomous Power State Transition Data Structure matches the Power State Descriptor table returned in observable 1.

Case 60: Power Limit Value does not disable or enable any Power States (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to send an Identify Command Specifying CNS value 01h to the Controller in order to receive back an Identify Controller Data Structure. If the Power Limit Support (PLS) bit of the Controller Attributes field in the Identify Controller Data Structure is not set to 1h, this test is not applicable.
2. Configure the NVMe host to send a Set Features command to the DUT for FID 23h with a Power Limit Value (PLV) of 0.
3. Configure the NVMe Host to send an Identify Command Specifying CNS value 01h to the Controller in order to receive back an Identify Controller Data Structure.
4. Configure the NVMe Host to send a Set Features command to the DUT for FID 23h with a Power Limit Value (PLV) and Power Limit Scale (PLS) set such that when multiplied together is higher than the Maximum Power (MP) and Max Power Scale (MXPS) of Power State Descriptor 0 multiplied together.
5. Configure the NVMe Host to send an Identify Command Specifying CNS value 01h to the Controller in order to receive back an Identify Controller Data Structure.

Observable Results:

1. Verify that there are no changes between the returned Power State Descriptor Table from steps 3 and 5.

© 2026 UNH–IOL

Test 1.2 – Set/Get Features Commands (OF-FYI, M)

Purpose: To verify that an NVMe Controller can properly execute a Get Features command. Also to verify that the NVMe Controller properly sets feature values after the NVMe Host issues a Set Features command.

References:

Old Ref: [1] NVMe Specification 5.13, 5.21, NVMe v1.3 ECN 002, NVMe v1.3 ECN 004a
NVM Express Base specification 2.0a : 5.15, 5.27
NVM Express Base Specification 5.2.26.1.30

Resource Requirements:

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

Last Modification: February 5th, 2026

Discussion: The Get Features command retrieves the attributes of the Feature indicated. The Get Features command uses the PRP Entry 1, PRP Entry 2, and Command Dword 10 fields. All other command specific fields are reserved.

The Set Features command specifies the attributes of the Feature indicated. The Set Features command uses the PRP Entry 1, PRP Entry 2, Command Dword 10, Command Dword 11, Command Dword 12, Command Dword 13, Command Dword 14, and Command Dword 15 fields. All other command specific fields are reserved.

The desired feature is specified in the Feature Identifier (FID) field of CDW10. Valid Feature Identifiers are described Table 1 and Table 2.

© 2026 UNH–IOL

Table 1 – Feature Identifiers

Feature Identifier	Optional/ Mandatory	Feature
00h	N/A	Reserved
01h	Mandatory	Arbitration
02h	Mandatory	Power Management
03h	Optional	LBA Range Type
04h	Mandatory	Temperature Threshold
05h	Mandatory	Error Recovery
06h	Optional	Volatile Write Cache
07h	Mandatory	Number of Queues
08h	Mandatory	Interrupt Coalescing
09h	Mandatory	Interrupt Vector Configuration
0Ah	Mandatory	Write Atomicity Normal
0Bh	Mandatory	Asynchronous Event Configuration
0Ch	Optional	Autonomous Power State Transition
0Dh	Optional	Host Memory Buffer
0Eh	Optional	Timestamp
0Fh	Optional	Keep Alive Timer
10h	Optional	Host Controller Thermal Management
11h	Optional	Non-Operational Power State Config
12h	Optional	Read Recovery Level Config
13h	Optional	Predictable Latency Mode Config
14h	Optional	Predictable Latency Mode Window
15h	Optional	Refer to NVM Command Set
16h	Optional	Host Behavior Support
17h	Optional	Sanitize Config
18h	Optional	Endurance Group Event Configuration
19h	Optional	I/O Command Set Profile
1Ah		Spinup Control
1Bh to 1Fh		Reserved
20h		Refer to the Key Value Command Set
21h to 77h		Reserved
78h – 7Ch	N/A	Reserved for Management Features
7Dh	Optional	Enhanced Controller Metadata
7Eh	Optional	Controller Metadata
7Fh	Optional	Namespace Metadata
80h – BFh	N/A	Command Set Specific (Reserved)
C0h – FFh	N/A	Vendor Specific

© 2026 UNH–IOL

Table 2 – NVM Command Set Specific Feature Identifiers

Feature Identifier	Optional/ Mandatory	Feature
80h	Optional	Software Progress Marker
81h	Optional	Host Identifier
82h	Optional	Reservation Notification Mask
83h	Optional	Reservation Persistence
84h	Optional	Namespace Write Protection Config
85h – BFh	N/A	Reserved

A completion queue entry is posted to the Admin Completion Queue when the controller has completed returning any attributes associated with the Feature. Depending on the Feature Identifier, Dword 0 of the completion queue entry may contain feature information

Test Setup: See Appendix A.

Case 1: SEL = 000b (OF, M)

Test Procedure:

1. For each of the features described in Table 1 and Table 2, perform the following:
    a. Configure the NVMe Host to issue a Get Features command with SEL field set to 000b indicating the specified FID value in CDW10 to the controller.
    b. Configure the NVMe Host to issue a Set Features command indicating the specified FID value in CDW10 setting valid values for each feature. If any feature is indicated as not changeable by the controller, testing that FID is not applicable. If all FIDs are indicated as not changeable the test is not applicable.
    c. Configure the NVMe Host to issue a second Get Features command with SEL field set to 000b indicating the specified FID value in CDW10 to the controller.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
2. Verify that the controller returns the Feature Information for each FID as specified in of the NVMe Specification for each Get Features command issued by the NVMe Host.
3. Verify that the Feature Information returned in the second Get Features command matches the values which were set by the NVMe Host using the Set Features command, if that feature is changeable.
4. Verify that all received responses have all Reserved fields set to 0h.

Possible Problems: If the current Host Identifier value is set to a non-zero value, then it is acceptable for the controller to abort the command with a status code of Command Sequence Error.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.2.1 and expect the same observable results.

1. **FID 01h Arbitration**
2. **FID 02h Power Management**
3. **FID 03h LBA Range Type**
4. **FID 04h Temperature Threshold**
5. **FID 05h Error Recovery**
6. **FID 06h Volatile Write Cache**
7. **FID 07h Number of Queues**
8. **FID 08h Interrupt Coalescing**
9. **FID 09h Interrupt Vector Configuration**

© 2026 UNH–IOL

10. **FID 0Ah Write Atomicity Normal**
11. **FID 0Bh Asynchronous Event Configuration**
12. **FID 0Ch Autonomous Power State Transition**
13. **FID 0Dh Host Memory Buffer**
14. **FID 0Eh Timestamp**
15. **FID 0Fh Keep Alive Timer**
16. **FID 10h Host Controlled Thermal Management**
17. **FID 11h Non-Operational Power State Config**
18. **FID 12h Read Recovery Level Config**
19. **FID 13h Predictable Latency Mode Config**
20. **FID 14h Predictable Latency Mode Window**
21. **FID 15h LBA Status Information Report Interval**
22. **FID 16h Host Behavior Support**
23. **FID 17h Sanitize Config**
24. **FID 18h Endurance Group Event Configuration**
25. **FID 19h I/O Command Set Profile**
26. **FID 1Ah Spinup Control**
27. **FID 7Dh Enhanced Controller Metadata**
28. **FID 7Eh Controller Metadata**
29. **FID 7Fh Namespace Metadata**
30. **FID 80h Software Progress Marker**
31. **FID 81h Host Identifier**
32. **FID 82h Reservation Notification Mask**
33. **FID 83h Reservation Persistence**
34. **FID 84h Namespace Write Protection Config**
35. **FID 21h Controller Data Queue**
36. **FID 78h Embedded Management Controller Address**
37. **FID 79h Host Management Agent Address**
38. **FID 24h Power Threshold**
39. **FID 25h Power Measurement**

Case 2: SEL = 001b (OF, M)

Test Procedure:

1. Check Bit 4 of the ONCS field to determine if the DUT supports setting the Save field in a Set Features command to a non-zero value and the Select field of the Get Features command to a non-zero value. If Bit 4 is set to 0h this test is not applicable.
2. Configure the NVMe Host to issue a Set Features command indicating the specified FID value in CDW10 setting valid values for each feature. If any feature is indicated as not changeable by the controller, testing that FID is not applicable. If all FIDs are indicated as not changeable the test is not applicable.
3. For each of the features described in Table 1 and Table 2, perform the following:
    a. Configure the NVMe Host to issue a Get Features command with SEL field set to 000b indicating the specified FID value in CDW10 to the controller.
    b. Configure the NVMe Host to issue a Get Features command with SEL field set to 001b indicating the specified FID value in CDW10 to the controller.
    c. Configure the NVMe Host to issue a Set Features command indicating the specified FID value in CDW10 changing the feature values for the feature to the controller, if that feature is changeable.
    d. Configure the NVMe Host to issue a Get Features command with SEL field set to 000b indicating the specified FID value in CDW10 to the controller.
    e. Configure the NVMe Host to issue a Get Features command with SEL field set to 001b indicating the specified FID value in CDW10 to the controller.

© 2026 UNH–IOL

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
2. Verify that the controller returns the Feature Information for each FID as specified in section 5.15 of the NVMe Specification for each Get Features command issued by the NVMe Host.
3. Verify that features values returned in step b and e have not changed.
4. Verify that feature values returned in step a and d have changed.
5. Verify that all received responses have all Reserved fields set to 0h.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.2.2 and expect the same observable results.

1. **FID 01h Arbitration**
2. **FID 02h Power Management**
3. **FID 03h LBA Range Type**

   Sub-case Procedure:

   1. Verify that the controller returns a status code of 'Invalid Field in Command' (02h) when testing FID 03h.

4. **FID 04h Temperature Threshold**
5. **FID 05h Error Recovery**
6. **FID 06h Volatile Write Cache**
7. **FID 07h Number of Queues**
8. **FID 08h Interrupt Coalescing**
9. **FID 09h Interrupt Vector Configuration**
10. **FID 0Ah Write Atomicity Normal**
11. **FID 0Bh Asynchronous Event Configuration**
12. **FID 0Ch Autonomous Power State Transition**
13. **FID 0Dh Host Memory Buffer**
14. **FID 0Eh Timestamp**
15. **FID 0Fh Keep Alive Timer**
16. **FID 10h Host Controlled Thermal Management**
17. **FID 11h Non-Operational Power State Config**
18. **FID 12h Read Recovery Level Config**
19. **FID 13h Predictable Latency Mode Config**
20. **FID 14h Predictable Latency Mode Window**
21. **FID 15h LBA Status Information Report Interval**
22. **FID 16h Host Behavior Support**
23. **FID 17h Sanitize Config**
24. **FID 18h Endurance Group Event Configuration**
25. **FID 19h I/O Command Set Profile**
26. **FID 1Ah Spinup Control**
27. **FID 7Dh Enhanced Controller Metadata**
28. **FID 7Eh Controller Metadata**
29. **FID 7Fh Namespace Metadata**
30. **FID 80h Software Progress Marker**
31. **FID 81h Host Identifier**
32. **FID 82h Reservation Notification Mask**
33. **FID 83h Reservation Persistence**
34. **FID 84h Namespace Write Protection Config**
35. **FID 21h Controller Data Queue**
36. **FID 78h Embedded Management Controller Address**
37. **FID 79h Host Management Agent Address**

© 2026 UNH–IOL

38. **FID 24h Power Threshold**
39. **FID 25h Power Measurement**

Case 3: SEL = 010b (OF, M)

Test Procedure:

1. Check Bit 4 of the ONCS field to determine of the DUT supports setting the Save field in a Set Features command to a non-zero value and the Select field of the Get Features command to a non-zero value. If Bit 4 is set to 0h this test is not applicable.
2. Configure the NVMe Host to issue a Set Features command indicating the specified FID value in CDW10 setting valid values for each feature. If any feature is indicated as not changeable by the controller, testing that FID is not applicable. If all FIDs are indicated as not changeable the test is not applicable.
3. For each of the features described in Table 1 and Table 2, perform ensure that the Feature is supported as saveable by the DUT. If the DUT does not support any features as saveable, this Test Case is not applicable. If the DUT does support certain features as saveable, perform the following:
    a. Configure the NVMe Host to issue a Get Features command with SEL set to 000b indicating the specified FID value in CDW10 to the controller.
    b. Configure the NVMe Host to issue a Set Features command with the SV bit set to 1h indicating the specified FID value in CDW10 setting changed values for the feature to the controller, for all changeable values.
    c. Configure the NVMe Host to issue a second Get Features command with SEL=010b indicating the specified FID value in CDW10 to the controller.
    d. Perform a Controller reset.
    e. Configure the NVMe Host to issue a second Get Features command with SEL=010b indicating the specified FID value in CDW10 to the controller.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
2. Verify that the controller returns the Feature Information for each FID as specified in section 5.12 of the NVMe Specification for each Get Features command issued by the NVMe Host.
3. Verify that the Feature Information returned in the second Get Features command matches the values which were set by the NVMe Host using the Set Features command, *and were saved across the reset event.*
4. Verify that all received responses have all Reserved fields set to 0h.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.2.3 and expect the same observable results.

1. **FID 03h LBA Range Type**
2. **FID 10h Host Controlled Thermal Management**
3. **FID 12h Read Recovery Level Config**
4. **FID 17h Sanitize Config**
5. **FID 19h I/O Command Set Profile**
6. **FID 1Ah Spinup Control**
7. **FID 1Bh Power Loss Signaling Config**
8. **FID 1Ch Performance Characteristics**
9. **FID 1Dh Flexible Data Placement**
10. **FID 1Eh Flexible Data Placement Events**
11. **FID 1Fh Namespace Admin Label**
12. **FID 78h Embedded Management Controller Address**
13. **FID 79h Host Management Agent Address**
14. **FID 80h Software Progress Marker**
15. **FID 83h Reservation Persistence**
16. **FID 24h Power Threshold**

© 2026 UNH–IOL

17. **FID 25h Power Measurement**

Case 4: SEL = 011b (OF, M)

Test Procedure:

1. Check Bit 4 of the ONCS field to determine of the DUT supports setting the Save field in a Set Features command to a non-zero value and the Select field of the Get Features command to a non-zero value. If Bit 4 is set to 0h this test is not applicable.
2. For each of the features described in Table 1 and Table 2, perform the following:
    a. Configure the NVMe Host to issue a Get Features command with SEL=011b indicating the specified FID value in CDW10 to the controller. Record the value of Dword 0 bits 0, 1, and 2 returned in the Completion Entry.
    b. Perform a Set Feature command with the SV field set to 1 for the specified FID.
    c. Perform a Get Feature command with the SEL field set to 000b, with NSID set to the current Namespace Identifier.
    d. Perform a valid Get Feature command with SEL set to 000b, followed by a valid Set Feature command with a new value for the specified FID.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
2. Verify that in step 2b,
    a. If bit 2 of Dword 0 of the completion entry from step 2a is set to 0h (i.e. the Feature Identifier is not changeable), the controller may return 'Feature Not Changeable' for the Set Features command.
    b. If bit 0 of Dword 0 of the completion entry from step 2a is set to 0h (i.e. the feature is not saveable), verify that the controller returns 'Feature Identifier Not Saveable' (0Dh) to the Set Feature Command in step 2b.
    c. If bit 0 of Dword 0 of the completion entry from step 2a is set to 1h (i.e. the feature is saveable), verify that the Set Feature Command in step 2b completes successfully.
3. Verify that in step 2c,
    a. If bit 1 of Dword 0 of the completion entry from step 2a is set to 0h (i.e. the Feature Identifier is not namespace specific), the DUT returns 'Feature Not Namespace Specific' (0Fh) for the Get Features command.
    b. If bit 1 of Dword 0 of the completion entry from step 2a is set to 1h (i.e. the Feature Identifier is namespace specific), the Get Features command in step 2c completes successfully.
4. Verify that in step 2d,
    a. If bit 2 of Dword 0 of the completion entry is set to 0h, the controller returns 'Feature not Changeable' (0Eh) for the Set Features command.
    b. If bit 2 of Dword 0 of the completion entry is set to 1h, the command completes successfully.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.2.4 and expect the same observable results.

1. **FID 01h Arbitration**
2. **FID 02h Power Management**
3. **FID 03h LBA Range Type**
4. **FID 04h Temperature Threshold**
5. **FID 05h Error Recovery**
6. **FID 06h Volatile Write Cache**
7. **FID 07h Number of Queues**
8. **FID 08h Interrupt Coalescing**
9. **FID 09h Interrupt Vector Configuration**
10. **FID 0Ah Write Atomicity Normal**
11. **FID 0Bh Asynchronous Event Configuration**

© 2026 UNH–IOL

12. **FID 0Ch Autonomous Power State Transition**
13. **FID 0Dh Host Memory Buffer**
14. **FID 0Eh Timestamp**
15. **FID 0Fh Keep Alive Timer**
16. **FID 10h Host Controlled Thermal Management**
17. **FID 11h Non-Operational Power State Config**
18. **FID 12h Read Recovery Level Config**
19. **FID 13h Predictable Latency Mode Config**
20. **FID 14h Predictable Latency Mode Window**
21. **FID 15h LBA Status Information Report Interval**
22. **FID 16h Host Behavior Support**
23. **FID 17h Sanitize Config**
24. **FID 18h Endurance Group Event Configuration**
25. **FID 19h I/O Command Set Profile**
26. **FID 1Ah Spinup Control**
27. **FID 7Dh Enhanced Controller Metadata**
28. **FID 7Eh Controller Metadata**
29. **FID 7Fh Namespace Metadata**
30. **FID 80h Software Progress Marker**
31. **FID 81h Host Identifier**
32. **FID 82h Reservation Notification Mask**
33. **FID 83h Reservation Persistence**
34. **FID 84h Namespace Write Protection Config**
35. **FID 21h Controller Data Queue**
36. **FID 78h Embedded Management Controller Address**
37. **FID 79h Host Management Agent Address**
38. **FID 24h Power Threshold**
39. **FID 25h Power Measurement**

Case 5: SEL = Reserved Value (OF, M)

Test Procedure:

1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure ONCS Bit 4 to determine if the product under test supports the Save field. If the product under test does not support the Save field, this test is not applicable.
2. For each of the features described in Table 1 and Table 2, perform a Get Features command with the SEL field set to 111b.

Observable Results:

1. Verify that after the completion of each command, the controller returns Error Status ‘Invalid Field in Command’ (02h).

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.2.5 and expect the same observable results.

1. **FID 01h Arbitration**
2. **FID 02h Power Management**
3. **FID 03h LBA Range Type**
4. **FID 04h Temperature Threshold**
5. **FID 05h Error Recovery**
6. **FID 06h Volatile Write Cache**
7. **FID 07h Number of Queues**

© 2026 UNH–IOL

8. **FID 08h Interrupt Coalescing**
9. **FID 09h Interrupt Vector Configuration**
10. **FID 0Ah Write Atomicity Normal**
11. **FID 0Bh Asynchronous Event Configuration**
12. **FID 0Ch Autonomous Power State Transition**
13. **FID 0Dh Host Memory Buffer**
14. **FID 0Eh Timestamp**
15. **FID 0Fh Keep Alive Timer**
16. **FID 10h Host Controlled Thermal Management**
17. **FID 11h Non-Operational Power State Config**
18. **FID 12h Read Recovery Level Config**
19. **FID 13h Predictable Latency Mode Config**
20. **FID 14h Predictable Latency Mode Window**
21. **FID 15h LBA Status Information Report Interval**
22. **FID 16h Host Behavior Support**
23. **FID 17h Sanitize Config**
24. **FID 18h Endurance Group Event Configuration**
25. **FID 19h I/O Command Set Profile**
26. **FID 1Ah Spinup Control**
27. **FID 7Dh Enhanced Controller Metadata**
28. **FID 7Eh Controller Metadata**
29. **FID 7Fh Namespace Metadata**
30. **FID 80h Software Progress Marker**
31. **FID 81h Host Identifier**
32. **FID 82h Reservation Notification Mask**
33. **FID 83h Reservation Persistence**
34. **FID 84h Namespace Write Protection Config**
35. **FID 21h Controller Data Queue**
36. **FID 78h Embedded Management Controller Address**
37. **FID 79h Host Management Agent Address**
38. **FID 24h Power Threshold**
39. **FID 25h Power Measurement**

Case 6: SEL = 011b Attempt to Change value indicated as Not Changeable (OF, M)

Test Procedure:

1. Configure the NVMe Host to issue a Get Feature command to the NVMe Controller for each supported Feature ID with SEL set to 011b to determine which features are changeable. If Dword 0 bit 2 of the completion entry of the Get Features command is set to ‘1’, then the Feature Identifier is changeable. If Dword 0 bit 2 of the completion entry of the Get Features command is cleared to ‘0’, then the Feature Identifier is not changeable.
2. For each Feature indicated as Not Changeable, issue a Set Feature Command for that feature offering a value identical to the current value.

Observable Results:

1. Verify that after the completion of each command, the controller returns status of either “Success” or “Feature not Changeable”.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.2.6 and expect the same observable results.

1. **FID 01h Arbitration**
2. **FID 02h Power Management**

© 2026 UNH–IOL

3. **FID 03h LBA Range Type**
4. **FID 04h Temperature Threshold**
5. **FID 05h Error Recovery**
6. **FID 06h Volatile Write Cache**
7. **FID 07h Number of Queues**
8. **FID 08h Interrupt Coalescing**
9. **FID 09h Interrupt Vector Configuration**
10. **FID 0Ah Write Atomicity Normal**
11. **FID 0Bh Asynchronous Event Configuration**
12. **FID 0Ch Autonomous Power State Transition**
13. **FID 0Dh Host Memory Buffer**
14. **FID 0Eh Timestamp**
15. **FID 0Fh Keep Alive Timer**
16. **FID 10h Host Controlled Thermal Management**
17. **FID 11h Non-Operational Power State Config**
18. **FID 12h Read Recovery Level Config**
19. **FID 13h Predictable Latency Mode Config**
20. **FID 14h Predictable Latency Mode Window**
21. **FID 15h LBA Status Information Report Interval**
22. **FID 16h Host Behavior Support**
23. **FID 17h Sanitize Config**
24. **FID 18h Endurance Group Event Configuration**
25. **FID 19h I/O Command Set Profile**
26. **FID 1Ah Spinup Control**
27. **FID 7Dh Enhanced Controller Metadata**
28. **FID 7Eh Controller Metadata**
29. **FID 7Fh Namespace Metadata**
30. **FID 80h Software Progress Marker**
31. **FID 81h Host Identifier**
32. **FID 82h Reservation Notification Mask**
33. **FID 83h Reservation Persistence**
34. **FID 84h Namespace Write Protection Config**
35. **FID 21h Controller Data Queue**
36. **FID 78h Embedded Management Controller Address**
37. **FID 79h Host Management Agent Address**
38. **FID 24h Power Threshold**
39. **FID 25h Power Measurement**

Case 7: NSID of FFFFFFFFh to Namespace Specific Feature (OF-FYI, M)

Test Procedure:

1. Check the specification version supported by the DUT. If the DUT supports v1.4 or higher this test is applicable, otherwise this test is not applicable.
2. Check the MDS bit in CTRATT and store the value.
3. Configure the NVMe Host to issue a Get Feature command to the NVMe Controller for each supported Feature ID with SEL set to 011b
4. For each Feature ID indicated as Namespace Specific (i.e. Dword 0 bit 1 of the completion entry of the Get Features command is set to ‘1’) perform a Get Feature with SEL=000b and NSID =FFFFFFFFh.
5. For each Feature ID indicated as Namespace Specific (i.e. Dword 0 bit 1 of the completion entry of the Get Features command is set to ‘1’) perform a Get Feature with SEL=000b and NSID =0h.

© 2026 UNH–IOL

6. For each Feature ID indicated as Namespace Specific (i.e. Dword 0 bit 1 of the completion entry of the Get Features command is set to '1') perform a set Feature with NSID =FFFFFFFFh.
7. For each Feature ID indicated as Namespace Specific (i.e. Dword 0 bit 1 of the completion entry of the Get Features command is set to '1') perform a set Feature with NSID =0h.

Observable Results:

1. This test checks implementation of NVMe v1.3 ECN 002. DUT's claiming to support NVMe v1.3 or earlier may implement the expected behavior. DUT's claiming to support NVMe v1.4 or later must implement the expected behavior.
2. For DUT's supported NVMe v1.4 or later, verify that the Get Feature Commands with NSID of FFFFFFFFh or 0h complete with status code of 'Invalid Namespace or Format' (0Bh).
3. For DUT's supported NVMe v1.4 or later, verify that the Set Feature Commands with NSID of FFFFFFFFh complete with status Success. If the MDS bit recorded in step 2 is cleared to 0h. If the MDS bit is set to 1h with a status of 'Invalid Field in Command' (02h).
4. For DUT's supported NVMe v1.4 or later, verify that the Set Feature Commands with NSID of 0h complete with status code of 'Invalid Namespace or Format' (0Bh).

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.2.7 and expect the same observable results.

1. **FID 01h Arbitration**
2. **FID 02h Power Management**
3. **FID 03h LBA Range Type**
4. **FID 04h Temperature Threshold**
5. **FID 05h Error Recovery**
6. **FID 06h Volatile Write Cache**
7. **FID 07h Number of Queues**
8. **FID 08h Interrupt Coalescing**
9. **FID 09h Interrupt Vector Configuration**
10. **FID 0Ah Write Atomicity Normal**
11. **FID 0Bh Asynchronous Event Configuration**
12. **FID 0Eh Timestamp**
13. **FID 7Eh Controller Metadata**
14. **FID 80h Software Progress Marker**
15. **FID 81h Host Identifier**
16. **FID 82h Reservation Notification Mask**

Sub-case Procedure:

4. For FID 83h verify that the Set Feature and Get Features Commands with NSID of 0h completes with status code of 'Invalid Namespace or Format' (0Bh) and when the Get Features Commands have an NSID of FFFFFFFFh, the command completes with status code of 'Invalid Field in Command' (02h).

17. **FID 83h Reservation Persistence**

Sub-case Procedure:

4. For FID 83h verify that the Set Feature and Get Features Commands with NSID of 0h completes with status code of 'Invalid Namespace or Format' (0Bh) and when the Get Features Commands have an NSID of FFFFFFFFh, the command completes with status code of 'Invalid Field in Command' (02h).

18. **FID 24h Power Threshold**
19. **FID 25h Power Measurement**

Case 8: VWC Feature (OF, M)

Test Procedure:

1. Check the VWC field of the Identify Controller Data Structure.

© 2026 UNH–IOL

2. Perform a Get Feature with SEL=000b for the VWC Feature 06h.
3. Perform a Set Feature for the VWC Feature 06h.

Observable Results:

1. If a volatile write cache is not present, then a Set Features command specifying the Volatile Write Cache feature identifier shall fail with Invalid Field in Command status, otherwise the Set Feature command should complete with status 'Success'.
2. If a volatile write cache is not present, then a Get Features specifying the Volatile Write Cache feature identifier should fail with status 'Invalid Field in Command' (02h), otherwise the Get Feature command should complete with status 'Success'.

Case 9: FID 03h, NSID=FFFFFFFFh (OF-FYI, M)

Test Procedure:

1. Perform a Set Feature with FID 03h and NSID=FFFFFFFFh. If FID=03h Is not supported this test is not applicable.

Observable Results:

1. Verify that the Set Feature command is aborted with status 'Invalid Field in Command' (02h).

Case 10: Controller Feature Values, NSID=0h or FFFFFFFFh (OF-FYI, M)

Test Procedure:

1. For each changeable FID that applies to the controller, perform a Set Feature with a new value and NSID=FFFFFFFFh, followed by a Get Feature with NSID=FFFFFFFFh.
2. For each changeable FID that applies to the controller, perform a Set Feature with a new value (unique from the previous step) and NSID=0h, followed by a Get Feature with NSID=0h.
3. For each changeable FID that applies to the controller, perform a Set Feature with a new value (unique from the previous step) and NSID=FFFFFFFFh, followed by a Get Feature with NSID=0h.
4. For each changeable FID that applies to the controller, perform a Set Feature with a new value (unique from the previous step) and NSID=0h, followed by a Get Feature with NSID=FFFFFFFFh.

Observable Results:

1. Verify that each Set Feature and Get Feature command completed successfully and that each Get Feature command reported the value as set in the previous Set Feature command.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.2.10 and expect the same observable results.

1. **FID 01h Arbitration**
2. **FID 02h Power Management**
3. **FID 04h Temperature Threshold**
4. **FID 06h Volatile Write Cache**
5. **FID 08h Interrupt Coalescing**
6. **FID 09h Interrupt Vector Configuration**
7. **FID 0Bh Asynchronous Event Configuration**
8. **FID 0Ch Autonomous Power State Transition**
9. **FID 0Fh Keep Alive Timer**
10. **FID 10h Host Controlled Thermal Management**
11. **FID 12h Read Recovery Level Config**
12. **FID 13h Predictable Latency Mode Config**
13. **FID 14h Predictable Latency Mode Window**

© 2026 UNH–IOL

14. **FID 15h LBA Status Information Report Interval**
15. **FID 16h Host Behavior Support**
16. **FID 17h Sanitize Config**
17. **FID 18h Endurance Group Event Configuration**
18. **FID 19h I/O Command Set Profile**
19. **FID 1Ah Spinup Control**
20. **FID 80h Software Progress Marker**
21. **FID 24h Power Threshold**
22. **FID 25h Power Measurement**

Case 11: Multiple Set Features Commands for FID 03h (OF-FYI, M)

Test Procedure:

1. Perform a Set Feature with FID 03h for a specific set of values in the LBA Range Type Data Structure Entry. If FID=03h Is not supported, or not changeable, this test is not applicable.
2. Perform a Get Operation for FID 03h.
3. Repeat steps 1 and 2 10 times, with new values in the LBA Range Type Data Structure Entry each time.

Observable Results:

1. Verify in all cases that the Get Feature command returned the values from the most recent Set Feature command.

Case 12: Timestamp FID 0Eh (OF-FYI, M)

Test Procedure:

1. Check the Identify Controller Data Structure ONCS field Bit 6 to determine if the DUT supports the Timestamp feature. If the DUT does not support the Timestamp feature, then this test is not applicable.
2. Perform a Set Feature with FID 0Eh to set a timestamp value in the DUT, start a timer on the testing station.
3. Perform a Get Feature operation with FID=0Eh to request to current timestamp value.
4. Perform a Controller Level Reset.
5. Perform a Get Feature operation with FID=0Eh to request to current timestamp value.

Observable Results:

1. Verify in all cases that the Set/Get Feature operations for FID=0Eh completed successfully.
2. If the Synch bit (returned in response to the Get Feature command) is set to 0, indicating that the controller does not stop counting, verify that the timestamp returned after the Controller Level Reset matches the timer running on the testing station.
3. If the Timestamp Origin field is set to 000b verify that the timestamp shows the time since the Controller Level Reset.
4. If the Timestamp Origin field is set to 001b, verify that the timestamp reflects the value that was initialized in the Set Features command.

Case 13: Get Feature Namespace Specific FID Valid NSID (OF-FYI, M)

Test Procedure:

1. Check that the DUT claims to support NVMe v1.4 or higher. If the DUT does not support NVMe v1.4 or higher, then this test is not applicable.
2. Perform a Get Feature operation to a valid NSID with FID=03h, LBA Range Type, which is a namespace specific Feature.
3. Perform a Get Feature operation to a valid NSID with FID=05h, Error Recovery, which is a namespace specific Feature.
4. Perform a Get Feature operation to a valid NSID with FID=82h, Reservation Notification Mask, which is a namespace specific Feature.

© 2026 UNH–IOL

5. Perform a Get Feature operation to a valid NSID with FID=83h, Reservation Persistence, which is a namespace specific Feature.
6. Perform a Get Feature operation to a valid NSID with FID=84h, Namespace Write Protection Config, which is a namespace specific Feature.
7. If any of the Features are not supported, then proceed to the next step in the procedure.

Observable Results:

1. Verify that for each supported Feature ID the Completion Queue Entry for the Get Feature command has Dword 0 bit 1 set to '1' to indicate that the feature is namespace specific.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.2.13 and expect the same observable results.

1. **FID 03h LBA Range Type**
2. **FID 05h Error Recovery**
3. **FID 82h Reservation Notification Mask**
4. **FID 83h Reservation Persistence**
5. **FID 84h Namespace Write Protection Config**

Case 14: Get Feature Namespace Specific FID NSID=FFFFFFFFh (OF-FYI, M)

Test Procedure:

1. Check that the DUT claims to support NVMe v1.4 or higher. If the DUT does not support NVMe v1.4 or higher, then this test is not applicable.
2. Perform a Get Feature operation to NSID=FFFFFFFFh with FID=03h, LBA Range Type, which is a namespace specific Feature.
3. Perform a Get Feature operation to NSID=FFFFFFFFh with FID=05h, Error Recovery, which is a namespace specific Feature.
4. Perform a Get Feature operation to NSID=FFFFFFFFh with FID=82h, Reservation Notification Mask, which is a namespace specific Feature.
5. Perform a Get Feature operation to NSID=FFFFFFFFh with FID=83h, Reservation Persistence, which is a namespace specific Feature.
6. Perform a Get Feature operation to NSID=FFFFFFFFh FID=84h, Namespace Write Protection Config, which is a namespace specific Feature.
7. If any of the Features are not supported, then proceed to the next step in the procedure.

Observable Results:

1. Verify that if Feature ID 03h is supported, the associated Get Feature command fails with status 'Invalid Namespace or Format' (0Bh).
2. Verify that if Feature ID 82h or 83h is supported, the associated Get Feature command fails with status 'Invalid Field In Command' (02h).
3. Verify that for every other supported Feature ID the command fails with status 'Invalid Namespace or Format' (0Bh).

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.2.14 and expect the same observable results.

1. **FID 03h LBA Range Type**
2. **FID 05h Error Recovery**
3. **FID 82h Reservation Notification Mask**
4. **FID 83h Reservation Persistence**
5. **FID 84h Namespace Write Protection Config**

Case 15: Set Feature Namespace Specific FID NSID=FFFFFFFFh (OF-FYI, M)

Test Procedure:

© 2026 UNH–IOL

1. Check that the DUT claims to support NVMe v1.4 or higher. If the DUT does not support NVMe v1.4 or higher, then this test is not applicable.
2. Check the MDS bit in the Identify Controller Data Structure.
3. Perform a Set Feature operation to NSID=FFFFFFFFh with FID=03h, LBA Range Type, which is a namespace specific Feature.
4. Perform a Set Feature operation to NSID=FFFFFFFFh with FID=05h, Error Recovery, which is a namespace specific Feature.
5. Perform a Set Feature operation to NSID=FFFFFFFFh with FID=82h, Reservation Notification Mask, which is a namespace specific Feature.
6. Perform a Set Feature operation to NSID=FFFFFFFFh with FID=83h, Reservation Persistence, which is a namespace specific Feature.
7. Perform a Set Feature operation to NSID=FFFFFFFFh FID=84h, Namespace Write Protection Config, which is a namespace specific Feature.
8. If any of the Features are not supported, then proceed to the next step in the procedure.
9. If MDS=0, perform a Get Feature command to each supported FID attempted previously in the test procedure.

Observable Results:

1. Verify that if MDS=0, that the DUT sets the specified feature value for all namespaces attached to the controller processing the command if the feature is supported.
2. Verify that if MDS=1, that the Set Feature command for supported Feature ID fails with status 'Invalid Field in Command' (02h).

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.2.15 and expect the same observable results.

1. **FID 03h LBA Range Type**
2. **FID 05h Error Recovery**
3. **FID 82h Reservation Notification Mask**
4. **FID 83h Reservation Persistence**
5. **FID 84h Namespace Write Protection Config**

Case 16: I/O Command Set Profile Feature (OF-FYI, FYI)

Test Procedure:

1. Check bit 43 in CAP.CSS. If this bit is set to 0 this test is not applicable.
2. Perform a Get Feature operation with FID=19h, I/O Command Set Profile Feature.

Observable Results:

1. Verify that the Get Feature command completes successfully.

Case 17: I/O Command Set Combination 0h (OF-FYI, FYI)

Test Procedure:

1. Check bit 43 in CAP.CSS. If this bit is set to 0 this test is not applicable.
2. Perform a Set Feature operation with FID=19h, I/O Command Set Profile Feature, and include an IOCSCI value which corresponds to an I/O Command Set Combination with a value of 0h.

Observable Results:

1. Verify that the Set Feature command completes with status 'I/O Command Set Combination Rejected' (2Bh).

Case 18: I/O Command Set Combination Not Supported (OF-FYI, FYI)

Test Procedure:

1. Check bit 43 in CAP.CSS. If this bit is set to 0 this test is not applicable.

© 2026 UNH–IOL

2. Perform a Set Feature operation with FID=19h, I/O Command Set Profile Feature, and include an IOCSCI value which corresponds to an I/O Command Set Combination which is not supported by the DUT.

Observable Results:

1. Verify that the Set Feature command completes with status 'I/O Command Set Combination Rejected' (2Bh).

Case 19: Feature Value Reset to Default after Controller Level Reset (OF-FYI, M)

Test Procedure:

1. For a given Feature, perform a Get Feature command to determine the current value, and a Get Feature command to determine the default value.
2. If the current value is the same as the default value, perform a Set Feature command to change the current value to a value different than the default value. Verify that the value was changed by performing a Get Feature command for the current value. If the Feature is not changeable, that Feature can be skipped. If the current value is different than the default value, proceed to the next step.
3. Perform a Controller Level Reset.
4. Perform a Get Feature command to retrieve the current value.
5. Repeat the procedure for each of the following Features in the listed subcases that are changeable.

Observable Results:

1. Verify that the Feature Value for the current value returned after the Controller Level Reset matched the default value.

Possible Problems: None known.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.2.19 and expect the same observable results.

1. **FID 01h Arbitration**
2. **FID 02h Power Management**
3. **FID 03h LBA Range Type**
4. **FID 04h Temperature Threshold**
5. **FID 05h Error Recovery**
6. **FID 06h Volatile Write Cache**
7. **FID 07h Number of Queues**
8. **FID 08h Interrupt Coalescing**
9. **FID 09h Interrupt Vector Configuration**
10. **FID 0Ah Write Atomicity Normal**
11. **FID 0Bh Asynchronous Event Configuration**
12. **FID 0Ch Autonomous Power State Transition**
13. **FID 0Dh Host Memory Buffer**
14. **FID 0Eh Timestamp**
15. **FID 0Fh Keep Alive Timer**
16. **FID 10h Host Controlled Thermal Management**
17. **FID 11h Non-Operational Power State Config**
18. **FID 12h Read Recovery Level Config**
19. **FID 13h Predictable Latency Mode Config**
20. **FID 14h Predictable Latency Mode Window**
21. **FID 15h LBA Status Information Report Interval**
22. **FID 16h Host Behavior Support**
23. **FID 17h Sanitize Config**
24. **FID 18h Endurance Group Event Configuration**
25. **FID 19h I/O Command Set Profile**
26. **FID 1Ah Spinup Control**

© 2026 UNH–IOL

27. **FID 7Dh Enhanced Controller Metadata**
28. **FID 7Eh Controller Metadata**
29. **FID 7Fh Namespace Metadata**
30. **FID 80h Software Progress Marker**
31. **FID 81h Host Identifier**
32. **FID 82h Reservation Notification Mask**
33. **FID 83h Reservation Persistence**
34. **FID 84h Namespace Write Protection Config**
35. **FID 24h Power Threshold**
36. **FID 25h Power Measurement**

Case 20: Embedded Management Controller Address Set Feature (OF-FYI, FYI)

Test Procedure:

1. Perform a Command and Feature Lockdown log page (LID 14h), to determine if Embedded Management Controller Address Set Feature is locked down, if not allowed this test is not applicable.
2. Perform a Get Log Page command for the Log Identifier 18h, Management Address List to determine if the Embedded Management is supported. If not supported, then this test is not applicable.
3. Perform a Set Feature operation with FID=78h, Embedded Management Controller Address, and include an Embedded Management Controller Address (Uniform Resource Indicator RFC 3986).

Observable Results:

1. Verify that the Set Feature command completes with successful status.

Case 21: Embedded Management Controller Address Get Feature (OF-FYI, FYI)

Test Procedure:

1. Perform a Command and Feature Lockdown log page (LID 14h) to determine if Embedded Management Controller Address Get Feature locked down, if not allowed this test is not applicable.
2. Perform a Get Log Page command for the Log Identifier 18h, Management Address List to determine if the Embedded Management is supported. If not supported, then this test is not applicable.
3. Perform a Get Feature operation with FID=78h, Embedded Management Controller Address.

Observable Results:

1. Verify that the Get Feature command completes with successful status and that the contents of the Embedded Management Controller Address field is valid (Uniform Resource Indicator RFC 3986).

Case 22: Host Management Agent Address Set Feature (OF-FYI, FYI)

Test Procedure:

1. Perform a Command and Feature Lockdown log page (LID 14h) to determine if Host Management Agent Address Set Feature is locked down, if not allowed this test is not applicable
2. Perform a Get Log Page command for the Log Identifier 18h, Management Address List to determine if the Embedded Management is supported. If not supported, then this test is not applicable.
3. Perform a Set Feature operation with FID=79h, Host Management Agent Address, and include a Host Management Agent Address (Uniform Resource Indicator RFC 3986).

Observable Results:

1. Verify that the Set Feature command completes with successful status.

Case 23: Host Management Agent Address Get Feature (OF-FYI, FYI)

Test Procedure:

© 2026 UNH–IOL

1. Perform a Command and Feature Lockdown log page (LID 14h) to determine if Host Management Agent Address Get Feature locked down, if not allowed this test is not applicable.
2. Perform a Get Log Page command for the Log Identifier 18h, Management Address List to determine if the Embedded Management is supported. If not supported, then this test is not applicable.
3. Perform a Get Feature operation with FID=79h, Host Management Agent Address.

Observable Results:

1. Verify that the Get Feature command completes with successful status and that the contents of the Host Management Agent Address field is valid (Uniform Resource Indicator RFC 3986).

Case 24: Get Feature Performance Characteristics, Standard Performance (1Ch) (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify Controller Command and check the value of the ONCS bit 4 to determine SEL field.
2. Configure the NVMe Host to issue a Get Feature command to the NVMe Controller with a Feature Identifier of 1Ch with the SEL field set to 011b, and Attribute Index set to Standard Performance Attribute (00h).
3. Perform a Get Log page of the Feature Identifier Supported and Effects log page (12h).
4. Configure the NVMe Host to issue a Get Feature command to the NVMe Controller with a Feature Identifier of 1Ch with the SEL field set to 000b and Attribute Index set to Standard Performance Attribute (00h).

Observable Results:

1. Verify that the Get Feature command completes successfully.
2. Verify that if the NS Specific bit returned is cleared to ‘0’, that the NVM Subsystem Scope bit is set to ‘1’, or if the NS Specific bit returned is set to ‘1’, that the Namespace Scope bit is set to ‘1’.
3. Verify that the data buffer is returned, and if the Performance Characteristics feature is namespace or controller specific.

Case 25: Get Feature Performance Characteristics, Performance Attribute Identifier List (C0h) (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Feature command to the NVMe Controller with a Feature Identifier of 1Ch with the SEL field set to 011b and Attribute Index set to Performance Attribute Identifier List (C0h).
2. Perform a Get Log page of the Feature Identifier Supported and Effects log page (12h).
3. Configure the NVMe Host to issue a Get Feature command to the NVMe Controller with a Feature Identifier of 1Ch with the SEL field set to 000b, and Attribute Index set to Standard Performance Attribute (00h).

Observable Results:

1. Verify that the Get Feature command completes successfully.
2. Verify that if the NS Specific bit returned is cleared to ‘0’, that the NVM Subsystem Scope bit is set to ‘1’, or if the NS Specific bit returned is set to ‘1’, that the Namespace Scope bit is set to ‘1’.
3. Verify that the data buffer is returned and the Reserved fields are cleared to ‘0’, with the Performance Attribute Identifier List.

Case 26: Get Feature Performance Characteristics, Invalid Field (1Ch) (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Feature command to the NVMe Controller with a Feature Identifier of 1Ch and Attribute Index set to a Reserved field.

Observable Results:

1. Verify that the Get Feature command returns the status of ‘Invalid Field in Command’ (02h).

© 2026 UNH–IOL

Case 27: Set Feature Performance Characteristics, Invalid Field (1Ch) (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Set Feature command to the NVMe Controller with a Feature Identifier of 1Ch, and Attribute Index set to a Reserved field.

Observable Results:

1. Verify that the Set Feature command returns the status of 'Invalid Field in Command' (02h).

Case 28: Set Feature Performance Characteristics, Standard Performance Attribute (1Ch) (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Set Feature command to the NVMe Controller with a Feature Identifier of 1Ch and Attribute Index field set to Standard Performance Attribute (00h).

Observable Results:

1. Verify that the controller aborts the Set Feature command with the status code of 'Invalid Field in Command' (02h).

Case 29: Set Feature Performance Characteristics, Vendor Specific, USVSPA=0 (1Ch) (OF-FYI, FYI)

Test Procedure:

1. Have the Host perform a Get Feature on the Performance Characteristics feature (1Ch) with the Attribute Index set to Performance Attribute Indentifier List (C0h) in Command Dword 11, if the USVSPA field returned from the Get Feature is not equal to zero skip this test.
2. Configure the NVMe Host to issue a Set Feature command to the NVMe Controller which specifies an Attribute Index field set to a Vendor Specific Performance Attribute between C1h and FFh.

Observable Results:

1. Verify that the controller aborts the Set Feature command returns the status code of 'Invalid Field in Command' (02h).

Case 30: Set Feature Performance Attribute Identifier List (C0h) (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Set Feature command with Feature Identifier 1Ch, to the NVMe Controller which specifies an Attribute Index field set to Performance Attribute Identifier List (C0h).

Observable Results:

1. Verify that the controller aborts the Set Feature command returns the status code of 'Invalid Field in Command' (02h).

Case 31: Get Feature Performance Attribute Identifier List (C0h) (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command, if Bit 4 of the ONCS field is not set to '1', then this test is not applicable.
2. Configure the NVMe Host to issue a Get Feature command with Feature Identifier 1Ch, to the NVMe Controller which specifies an Attribute Index field set to Performance Attribute Identifier List (C0h), and the Select field is set to Current (000b)

© 2026 UNH–IOL

3. Configure the NVMe Host to issue a Get Feature command with Feature Identifier 1Ch, to the NVMe Controller which specifies an Attribute Index field set to Performance Attribute Identifier List (C0h), and the Select field is set to Default (001b)
4. Configure the NVMe Host to issue a Get Feature command with Feature Identifier 1Ch, to the NVMe Controller which specifies an Attribute Index field set to Performance Attribute Identifier List (C0h), and the Select field is set to Saved (010b)

Observable Results:

1. Verify that each of the Get Feature Commands successfully completes and that if a list is returned, that all fields are displayed and the Attribute Types matches the Select field for each instance in the table

Case 32: HOSTID Resets to 0h (OF-FYI, FYI)

Test Procedure:

1. Determine if the controller supports the Host Identifier feature identifier (81h), if not this test is not applicable.
2. Configure the NVMe Host to issue a Get Feature to FID 81h specifying the Enable Extended Host Identifier (EXHID) bit in Command Dword 11 (bit 00):
    a. If HOSTID is zero, issue a Set Feature to FID 81h (specifying the requested size via the EXHID bit) to set HOSTID to a valid non-zero value.
3. Perform an NVMe Controller Level Reset.
4. Configure the NVMe Host to issue a Get Feature to FID 81h specifying the same EXHID bit value used in step 2a.

Observable Results:

1. Verify the Get Feature in step 4 returns a HOSTID cleared to 0h for Drives 2.0 or greater.
2. Verify the data structure layout returned in step 4 based on the EXHID bit:
    a. If EXHID was cleared to ‘0’: Verify the 64-bit Host Identifier value is in bytes 07:00 and bytes 15:08 are reserved (cleared to 0h),
    b. If EXHID was set to ‘1’: Verify the controller reports a 128-bit identifier in bytes 15:00.

Case 33: HOSTID set to Non-Zero value (OF-FYI, FYI)

Test Procedure:

1. Determine if the controller supports the Host Identifier feature identifier (81h), if not this test is not applicable.
2. Configure the NVMe Host to issue a Get Feature to FID 81h specifying the Enable Extended Host Identifier (EXHID) bit in Command Dword 11 (bit 00). If HOSTID is zero, configure the host to issue a Set Feature to FID 81h (specifying the requested size via the EXHID bit) to set HOSTID to a valid non-zero value.
3. Configure the host to issue a Set Feature to FID 81h (specifying the same EXHID bit value used in step 2) to set HOSTID to a different valid non-zero value than currently set.
4. Configure the NVMe Host to issue a Get Feature to FID 81h specifying the same EXHID bit value used in step 2.

Observable Results:

1. Verify the Set Feature on step 3 aborted with a status code of ‘Command Sequence Error’ (0Ch) for drives 2.0 or greater.
2. Verify that HOSTID from the get feature in step 4 is the same as the HOSTID at the end of step 2.
3. Verify the data structure layout returned in step 4 based on the EXHID bit:
    a. If EXHID was cleared to ‘0h’: Verify the 64-bit Host Identifier value is in bytes 07:00 and bytes 15:08 are reserved (cleared to 0h).
    b. If EXHID was set to ‘1h’: Verify the controller reports a 128-bit identifier in bytes 15:00.

Case 34: HOSTID Not Saveable (OF-FYI, FYI)

Test Procedure:

© 2026 UNH–IOL

1. Determine if the Controller supports the Host Identifier feature identifier (81h), if not this test is not applicable.
2. Configure the NVMe Host to issue a set feature command to FID 0x81 with a valid value & SV set to 1h.

Observable Results:

1. Verify the command is aborted with feature aborted with status code of 'Feature Identifier Not Saveable' (0Dh) on drives 2.0 or greater

Possible Problems: If HOSTID is non-zero the command may be aborted with Command sequence error.

Case 35: Temperature Hysteresis Threshold, invalid TMPSEL (OF-FYI, FYI)

Test Procedure:

1. Have the host perform a get on the Identify Controller Data Structure, I/O Command Set Independent CNS 01h, if the Temperature Threshold Maximum Hysteresis (THPTHMH) in the Temperature Threshold Hysteresis Attributes (TMPTHHA). If the TMPHMH is cleared to 0 this test is not applicable.
2. Have the host perform a Set Feature on Temperature Threshold (04h), with Temperature Threshold Hysteresis (TMPTHH) set to 001b, Threshold Type Select (THSEL) set to Over Temperature Threshold (00b) and Threshold Temperature Select (TMPSEL) set to 09h.

Observable Results:

1. Verify that the Set Feature is aborted with a returned status code of 'Invalid Field in Command' (02h).

Case 36: Temperature Hysteresis Threshold, Greater than TMPTHMH (OF-FYI, FYI)

Test Procedure:

1. Have the host perform a get on the Identify Controller Data Structure, I/O Command Set Independent CNS 01h, if the Temperature Threshold Maximum Hysteresis (THPTHMH) in the Temperature Threshold Hysteresis Attributes (TMPTHHA). If the TMPHMH is cleared to 0 this test is not applicable. If the TMPHMH is not cleared to 0, then save this value.
2. Have the host perform a Set Feature on Temperature Threshold (04h), with Temperature Threshold Hysteresis (TMPTHH) set to a value TMPHMH+1, Threshold Type Select (THSEL) set to Over Temperature Threshold (00b).

Observable Results:

1. Verify that the Set Feature is aborted with a returned status code of 'Invalid Field in Command' (02h).

Case 37: Namespace Admin Label (1Fh), SEL=000b (OF-FYI, FYI)

Test Procedure:

1. Check the Bit 4 of the ONCS field to determine if the DUT supports setting the Save field in a Set Features command to a non-zero value and the Select field of the Get Features command to a non-zero value. If Bit 4 is set to 0 this test is not applicable.
2. Configure the NVMe Host to issue Get Log Page command for the Feature Identifiers Supported and Effects Log Page (LID 12h). If bit 0 FID Supported (FUSPP) of the Feature Identifier Supported field for Feature Identifier 1Fh is cleared to 0 this test is not applicable.
3. Specifying Feature Identifier 1Fh 'Namespace Admin Label', perform the following:
    1. Configure the NVMe Host to issue a Get Features command with SEL field set to 000b indicating the specified FID value in CDW10 to the controller.
    2. Configure the NVMe Host to issue a Set Features command indicating the specified FID value in CDW10 setting valid values for the feature.
    3. Configure the NVMe Host to issue a second Get Features command with SEL field set to 000b indicating the specified FID value in CDW10 to the controller.

Observable Results:

© 2026 UNH–IOL

1. Verify that after the completion of each command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
2. Verify that the controller returns the Feature Information as specified in the NVMe Specification for each Get Features command issued by the NVMe Host.
3. Verify that the Feature Information returned in the second Get Features command matches the values which were set by the NVMe Host using the Set Features command.
4. Verify that all received responses have all Reserved fields set to 0.

Case 38: Namespace Admin Label (1Fh), SEL=001b (OF-FYI, FYI)

Test Procedure:

1. Check the Bit 4 of the ONCS field to determine if the DUT supports setting the Save field in a Set Features command to a non-zero value and the Select field of the Get Features command to a non-zero value. If Bit 4 is set to 0 this test is not applicable.
2. Configure the NVMe Host to issue Get Log Page command for the Feature Identifiers Supported and Effects Log Page (LID 12h). If bit 0 FID Supported (FUSPP) of the Feature Identifier Supported field for Feature Identifier 1Fh is cleared to 0 this test is not applicable.
3. Specifying Feature Identifier 1Fh ‘Namespace Admin Label’, perform the following:
    a. Configure the NVMe Host to issue a Get Features command with SEL field set to 000b indicating the specified FID value in CDW10 to the controller.
    b. Configure the NVMe Host to issue a Get Features command with SEL field set to 001b indicating the specified FID value in CDW10 to the controller.
    c. Configure the NVMe Host to issue a Set Features command indicating the specified FID value in CDW10 setting valid values for the feature if that feature is changeable.
    d. Configure the NVMe Host to issue a second Get Features command with SEL field set to 000b indicating the specified FID value in CDW10 to the controller.
    e. Configure the NVMe Host to issue a second Get Features command with SEL field set to 001b indicating the specified FID value in CDW10 to the controller.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
2. Verify that the controller returns the Feature Information for each FID as specified in section 5.12 of the NVMe Specification for each Get Features command issued by the NVMe Host.
3. Verify that features values returned in step b do not match the features value returned in d.
4. Verify that feature values returned in step b match the features returned in step e.
5. Verify that all received responses have all Reserved fields set to 0.

Case 39: Namespace Admin Label (1Fh), SEL=010b (OF-FYI, FYI)

Test Procedure:

1. Check the Bit 4 of the ONCS field to determine if the DUT supports setting the Save field in a Set Features command to a non-zero value and the Select field of the Get Features command to a non-zero value. If Bit 4 is set to 0 this test is not applicable.
2. Configure the NVMe Host to issue Get Log Page command for the Feature Identifiers Supported and Effects Log Page (LID 12h). If bit 0 FID Supported (FUSPP) of the Feature Identifier Supported field for Feature Identifier 1Fh is cleared to 0 this test is not applicable.
3. Specifying Feature Identifier 1Fh ‘Namespace Admin Label’, perform the following:
    1. Configure the NVMe Host to issue a Get Features command with SEL set to 000b indicating the specified FID value in CDW10 to the controller.
    2. Configure the NVMe Host to issue a Set Features command with the SV bit set to 1 indicating the specified FID value in CDW10 setting changed values for the feature to the controller, for all changeable values.

© 2026 UNH–IOL

3. Configure the NVMe Host to issue a second Get Features command with SEL=010b indicating the specified FID value in CDW10 to the controller.
4. Perform a Controller reset.
5. Configure the NVMe Host to issue a second Get Features command with SEL=010b indicating the specified FID value in CDW10 to the controller.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
2. Verify that the controller returns the Feature Information for each FID as specified in section 5.12 of the NVMe Specification for each Get Features command issued by the NVMe Host.
3. Verify that the Feature Information returned in the second Get Features command matches the values which were set by the NVMe Host using the Set Features command, *and were saved across the reset event.*
4. Verify that all received responses have all Reserved fields set to 0.

Case 40: Namespace Admin Label (1Fh), SEL=011b (OF-FYI, FYI)

Test Procedure:

1. Check the Bit 4 of the ONCS field to determine of the DUT supports setting the Save field in a Set Features command to a non-zero value and the Select field of the Get Features command to a non-zero value. If Bit 4 is set to 0h this test is not applicable.
2. Configure the NVMe Host to issue Get Log Page command for the Feature Identifiers Supported and Effects Log Page (LID 12h). If bit 0 FID Supported (FUSPP) of the Feature Identifier Supported field for Feature Identifier 1Fh is cleared to 0h this test is not applicable.
3. Specifying Feature Identifier 1Fh 'Namespace Admin Label', perform the following:
    a. Configure the NVMe Host to issue a Get Features command with SEL=011b indicating the specified FID value in CDW10 to the controller. Record the value of Dword 0 bits 0, 1, and 2 returned in the Completion Entry.
    b. Perform a Set Feature command with the SV field set to 1h for the specified FID.
    c. Perform a Get Feature command with the SEL field set to 000b, with NSID set to an active Namespace.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
2. Verify that in step 3b,
    a. If bit 0 of Dword 0 of the completion entry for the Get Feature Command sent in step 3a, is set to 1h (i.e. the feature is saveable), verify that the Set Feature Command in step 3b completes successfully.
3. Verify that in step 3c,
    a. If bit 1 of Dword 0 of the completion entry is set to 0h (i.e. the Feature Identifier is not namespace specific), the DUT returns the controller specific value for that FID and the command completes successfully.
    b. If bit 1 of Dword 0 of the completion entry is set to 1h (i.e. the Feature Identifier is namespace specific), the DUT returns the namespace specific value for that FID and the command completes successfully.

Case 41: Namespace Admin Label (1Fh), SEL=Reserved Value (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure ONCS Bit 4 to determine if the product under test supports the Save field. If the product under test does not support the Save field, this test is not applicable.
2. Configure the NVMe Host to issue Get Log Page command for the Feature Identifiers Supported and Effects Log Page (LID 12h). If bit 0 FID Supported (FUSPP) of the Feature Identifier Supported field for Feature Identifier 1Fh is cleared to 0h this test is not applicable.

© 2026 UNH–IOL

3. For Feature Identifier 1Eh ‘Namespace Admin Label’, perform a Get Features command with the SEL field set to 111b.

Observable Results:

1. Verify that after the completion of each command, the controller returns Error Status ‘Invalid Field in Command’ (02h).

Case 42: Namespace Admin Label (1Fh) (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue Get Log Page command for the Feature Identifiers Supported and Effects Log Page (LID 12h). If bit 0 FID Supported (FUSPP) of the Feature Identifier Supported field for Feature Identifier 1Fh is cleared to 0h this test is not applicable.
2. For each Namespace in the NVM Subsystem:
    a. Configure the NVMe Host to issue a Get Features command to the NVMe Controller with a Feature Identifier of 1Fh.
    b. Configure the NVMe Host to issue a Set Features command to the NVMe Controller with a Feature Identifier of 1Fh specifying the Save (SV) bit set to 1h and a Namespace Admin Label of a known UTF-8 String that differs from the value returned by the previous Get Features command.
    c. Configure the NVMe Host to issue a Get Features command to the NVMe Controller with a Feature Identifier of 1Fh.
    d. Configure the NVMe Host to issue a Set Features command to the NVMe Controller with a Feature Identifier of 1Fh specifying the Save (SV) bit set to 1h and a Namespace Admin Label of a known UTF-8 String that differs from the previous Set Features command which is 256 bytes in length.
    e. Configure the NVMe Host to issue a Get Features command to the NVMe Controller with a Feature Identifier of 1Fh.

Observable Results:

1. Verify all commands completed successfully.
2. For each Namespace verify that as a result of each Set Features command the subsequent Get Features command returns the value specified in the Set Features command.

Case 43: Namespace Admin Label (1Fh) Saveable (OF-FYI, M)

Test Procedure:

1. Check bit 4 of the ONCS field to determine if the DUT supports Save field in the Set Features command. If Bit 4 is set to 1h this test is not applicable.
2. Configure the NVMe Host to issue Get Log Page command for the Feature Identifiers Supported and Effects Log Page (LID 12h).
3. For each Namespace in the NVM Subsystem:
    a. Configure the NVMe Host to issue a Get Features command to the NVMe Controller with a Feature Identifier of 1Fh.
    b. Configure the NVMe Host to issue a Set Features command to the NVMe Controller with a Feature Identifier of 1Fh specifying the Save (SV) bit set to 1h and a Namespace Admin Label of a known UTF-8 String that differs from the value returned by the previous Get Features command.

Observable Results:

1. Verify that the Get Log Page command completed successfully, and bit 0 FID Supported (FUSPP) of the Feature Identifier Supported field for Feature Identifier 1Fh is cleared to 0h.
2. For each Namespace verify that the Get Features and the Set Features commands issued both completed with status: ‘Invalid Field in Command’ (02h).

© 2026 UNH–IOL

Case 44: Reservation Persistence (83h), not Saveable (OF-FYI, FYI)

Test Procedure:

1. Check bit 4 of the ONCS field to determine if the DUT supports Save field in the Set Features command. If Bit 4 is set to 0 this test is not applicable.
2. Configure the NVMe Host to issue a Get Features command to the NVMe Controller with a Feature Identifier of 83h and SEL field set to 011b.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the Saveable bit returned by the Get Feature to FID 83h is cleared to 0.

Case 45: Get Feature LBA Range Type (03h) Data Structure (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify Controller Command and check the value of the ONCS bit 4. If the DUT does not support the SEL Field, this test is not applicable.
2. Configure the NVMe Host to issue a Get Feature command to the NVMe Controller with a Feature Identifier of 03h and SEL field set to 001b.

Observable Results:

1. Verify all commands complete successfully.
2. Verify that the default value for FID 03h has the Number of LBA Ranges field cleared to 0h and the LBA Range Type data structure contains:
    1. a single entry with the Type field cleared to 0h,
    2. Attributes field set to 1h
    3. Starting LBA field cleared to 0h.
    4. Number of Logical Blocks field set to indicate the number of LBAs in the namespace; and
    5. GUID field cleared to 0h or set to a globally unique identifier.

Case 46: Get Feature Host Metadata, GDHM (FYI)

Test Procedure:

1. Check the NVME Specification version supported by DUT using the value reported in the Identify Controller Data Structure VER field (83:80). If the DUT supports v1.4 or lower, then this test may not be applicable. If the controller supports FIDs 7Dh, 7Eh, and 7Fh, then this test is applicable irrespective of the NVMe Base Specification version.
2. Have the host perform a Controller Level Reset.
3. Have the host perform a Get Feature on Host Metadata (7Dh, 7Eh, & &Fh), with the Generate Default Host Metadata (GDHM) field cleared to ‘0h’ and save off the values.
4. Have the host perform a second Get Feature on Host Metadata (7Dh, 7Eh, & &Fh), with the GDHM field cleared to ‘0h’, and save off the values.
5. Have the host Perform a Get Feature on Host Metadata (7Dh, 7Eh, & &Fh), with the GDHM field set to ‘1h’, and save off the values.
6. Configure the NVMe Host to perform a Controller Level Reset.
7. Have the host perform a second Get Feature on Host Metadata (7Dh, 7Eh, & &Fh), with the GDHM field cleared to ‘0h’, and save off the values.

Observable Results:

1. Verify that all Get Feature commands complete successfully.
2. Verify that Host Metadata fields are equal from procedure steps 3 & 4.
3. Verify that the Host Metadata fields are different the values saved in procedure step 4 and step 5.
4. Verify that that the Host Metadata fields are equal from procedure steps 4 & 7.

© 2026 UNH–IOL

Possible Problems: There is the potential that the controller will return no vendor specific strings.

Case 47: Feature Values after Controller Level Reset for Features that are not Saveable and are Persistent: (OF-FYI)

Test Procedure:

1. For a given Feature, perform a Get Feature command to determine the current value, and a Get Feature command to determine the default value.
2. If the current value is the same as the default value, perform a Set Feature command to change the current value to a value different from the default value. Verify that the value was changed by performing a Get Feature command for the current value. If the feature is savable, the feature is not applicable to this test. If the current value is different from the default value, proceed to the next step.
3. Perform a Controller Level Reset.
4. Perform a Get Feature command with SEL cleared to 0h to retrieve the current value.

Observable Results:

1. Verify that the Feature Value for the current value returned after the Controller Level Reset matched the set value.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.2.47 and expect the same observable results.

1. **FID 03h LBA Range Type**

   FID 03h is persistent across Controller Level Resets, and is not applicable to this test case. It is being left in the test plan for historical and logical reasons, to not have to re-enumerate the subsequent tests. Removed from testing in version 25.0.

2. **FID 10h Host Controlled Thermal Management**
3. **FID 12h Read Recovery Level Config**
4. **FID 17h Sanitize Config**
5. **FID 19h I/O Command Set Profile**
6. **FID 1Ah Spinup Control**
7. **FID 1Bh Power Loss Signaling Config**
8. **FID 1Ch Performance Characteristics**
9. **FID 1Dh Flexible Data Placement**
10. **FID 1Eh Flexible Data Placement Events**
11. **FID 1Fh Namespace Admin Label**
12. **FID 78h Embedded Management Controller Address**
13. **FID 79h Host Management Agent Address**
14. **FID 80h Software Progress Marker**
15. **FID 83h Reservation Persistence**
16. **FID 24h Power Threshold**
17. **FID 25h Power Measurement**

Case 48: NSID != 0 on Feature values that apply to NVM Subsystem Scope, Domain Scope, or Endurance Group Scope: (OF-FYI)

Test Procedure:

1. For a given Feature, perform a Get Feature command with a non-zero NSID.
2. For a given Feature, perform a Set Feature command with a non-zero NSID.

Observable Results:

1. Verify all commands are aborted with a status code of ‘Invalid Field in Command’ (02h).

© 2026 UNH–IOL

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.2.48 and expect the same observable results.

1. **FID 12h Read Recovery Level Config**
2. **FID 17h Sanitize Config**
3. **FID 18h Endurance Group Event Configuration**
4. **FID 1Ah Spinup Control**
5. **FID 1Bh Power Loss Signaling Config**
6. **FID 1Ch Performance Characteristics**
7. **FID 1Dh Flexible Data Placement**
8. **FID 78h Embedded Management Controller Address**
9. **FID 79h Host Management Agent Address**

Case 49: Configurable Device Personality Get Features CDP with Unsupported Personality Identifier (FYI)

Test Procedure:

1. Verify the controller supports the CDP capability by checking bit 0 (FSUPP) for FID 22h in the Feature Identifiers Supported and Effects log page (LID 12h).
2. Configure the NVMe Host to issue a Get Features command for FID 22h (CDP), and in Command Dword 13, set the PERID field (bits 07:00) to a value identified as unsupported in step 2 (e.g., a reserved value such as BFh), and the Select Field value of 000b (Current).

Observable Results:

1. Verify that the Get Features command is aborted by the controller, and the completion queue entry contains a status code of ‘Invalid Field in Command’ (02h).

Case 50: Set Features with Invalid All Personalities Request (FYI)

Test Procedure:

1. Verify the controller supports the CDP capability by checking bit 0 (FSUPP) for FID 22h in the Feature Identifiers Supported and Effects log page (LID 12h).
2. Configure the NVMe Host to issue a Set Features command targeting FID 22h (CDP), in Command Dword 13, set the PERID field (Bits 07:00) to FFh (All Personalities), and in Command Dword 13, set the CHPS bit (Bit 09) to ‘1h’.

Observable Results:

1. Verify that the Get Features command is aborted by the controller, and the completion queue entry contains a status code of ‘Invalid Field in Command’ (02h).

Case 51: Host Metadata Descriptor Limit (FYI)

Test Procedure:

1. Determine if the controller supports Enhanced Controller Metadata (FID 7Dh). If not, this test is not applicable.
2. Configure the NVMe Host to issue a Set Features command for FID 7Dh with the Element Action (EA) field in Command Dword 11 set to 10b (Add Entry Multiple).
3. Provide a Host Metadata data structure in the data buffer that contains a number of Metadata Element Descriptors which, when combined with any existing descriptors for this feature, results in a total count greater than 255.

Observable Results:

1. Verify the Set Features command in step 3 is aborted with a status code of ‘Invalid Field in Command’ (02h).

© 2026 UNH–IOL

Test 1.3 – Get Log Page Command (OF, M)

Purpose: To verify that an NVMe Controller can properly execute a Get Log Page command.

References:

Old Ref: [1] NVMe Specification 5.14, NVMe v1.3 TP 4007a and TP 4030
NVM Express Base specification 2.0a : 5.16

Resource Requirements:

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

Last Modification: July 15, 2019

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

The desired log page is specified in the Log Page Identifier (LID) field of CDW10. Valid Log Identifiers are described in Table 3 and Table 4.

© 2026 UNH–IOL

Table 3 – Log Page Identifiers

Log Identifier	Optional/ Mandatory	Log Page
00h	Mandatory	Supported Log Page
01h	Mandatory	Error Information
02h	Mandatory	SMART / Health Information
03h	Mandatory	Firmware Slot Information
04h	Optional	Changed Namespace List
05h	Optional	Command Effects Log
06h	Optional	Device Self Test
07h	Optional	Telemetry Host-Initiated
08h	Optional	Telemetry Controller Initiated
09h	Optional	Endurance Group Information
0Ah	Optional	Predictable Latency Per NVM Set
0Bh	Optional	Predictable Latency Event Aggregate
0Ch	Optional	Asymmetric Namespace Access
0Dh	Optional	Persistent Event Log
0Eh		Refer to NVM Command Set
0Fh	Optional	Endurance Group Event Aggregate
10h	Optional	Media Unit Status
11h	Optional	Supported Capacity Configuration List
12h	Mandatory	Feature identifiers supported and effects
13h	Mandatory	NVMe-MI commands supported and effects
14h	Optional	Command and feature lockdown
15h	Optional	Boot Partition
16h	Optional	Rotational Media Information
17h – 6Fh	N/A	Reserved
18h	Optional	Management Address List
1Ah	Optional	Reachability Groups
1Bh	Optional	Reachability Associations
1Ch	Optional	Changed Allocated Namespace List
1Dh-1Fh	Reserved	
20h	Optional	FDP Configurations
21h	Optional	Reclaim Unit Handle Usage
22h	Optional	FDP Statistics
23h	Optional	FDP Events
24h-6Fh	Reserved	
70h		Discovery
71h-7Fh		Reserved for NVMe over Fabrics

© 2026 UNH–IOL

Table 4 – NVM Command Set Specific Log Page Identifiers

Log Identifier	Optional/Mandatory	Log Page
80h	Optional	Reservation Notification
81h	Optional	Sanitize Status
82h – BEh	N/A	I/O Command Set Specific
BFh		Refer to Zone Namespace Command Set
C0h to FFh		Vendor Specific

A completion queue entry is posted to the Admin Completion Queue when the log has been posted to the memory buffer indicated in PRP Entry 1.

Test Setup: See Appendix A.

Case 1: Supported LIDs (OF, M)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command specifying each of the Mandatory Log Page Identifiers described in Table 3 and 4 above, to the NVMe Controller.

Observable Results:

1. Verify that a completion queue entry is posted to the Admin Completion Queue when the log has been posted to the memory buffer indicated in PRP Entry 1.
2. In each case verify that the information returned in the Log Page information is formatted according to the respective descriptions for the log pages in section 5.14 of the NVMe Specification.
3. Verify that all received responses have all Reserved fields set to 0.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.3.1 and expect the same observable results.

1. **LID 00h Supported Log Pages**
2. **LID 01h Error Information**
3. **LID 02h SMART/Health Information**
4. **LID 03h Firmware Slot Information**
5. **LID 04h Changed Attached Namespace List**
6. **LID 05h Commands Supported and Effects**
7. **LID 06h Device Self-test**
8. **LID 07h Telemetry Host-Initiated**
9. **LID 08h Telemetry Controller-Initiated**
10. **LID 09h Endurance Group Information**
11. **LID 0Ah Predictable Latency Per NVM Set**
12. **LID 0Bh Predictable Latency Event Aggregate**
13. **LID 0Ch Asymmetric Namespace Access**
14. **LID 0Dh Persistent Event Log**
15. **LID 0Eh LBA Status Information**
16. **LID 0Fh Endurance Group Event Aggregate**
17. **LID 10h Media Unit Status**
18. **LID 11h Supported Capacity Configuration List**
19. **LID 12h Feature Identifiers Supported and Effects**
20. **LID 13h NVMe-MI Commands Supported and Effects**

Sub-case Procedure:

© 2026 UNH–IOL

1. LID 13h is only mandatory to support if the NVMe-MI Send and NVMe-MI Receive commands are supported. If these commands are not supported by the controller under test, then this subcase is not applicable.

21. LID 14h Command and Feature Lockdown
22. LID 15h Boot Partition
23. LID 16h Rotational Media Information
24. LID 70h Discovery
25. LID 80h Reservation Notification
26. LID 81h Sanitize Status
27. LID BFh Changed Zone List

Sub-case Procedure:

1. This test is only applicable for devices that support the Zoned Namespace I/O Command Set.

28. LID 83h Downloadable Program Types List

Sub-case Procedure:

1. This test is only applicable for devices that support the Computational Programs Command Set.

29. LID 84h Memory Range Set List

Sub-case Procedure:

1. This test is only applicable for devices that support the Computational Programs Command Set.

30. LID 1Ch Changed Allocated Namespace List
31. LID 18h Management Address List
32. LID 25h Power Management

Case 2: Unsupported Vendor Specific LIDs (OF, M)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command specifying a single log identifier (LID) from the vendor specific LID range (C0h-FFh) that is not supported by the NVMe Controller. It may be necessary to query product owner for an unsupported LID from the vendor specific range, or consult product data sheet. If the DUT supports all Vendor Specific LIDs, then this test is not applicable.

Observable Results:

1. Verify that each command completes with an error code of Invalid Log Page 09h.
2. Verify that all received responses have all Reserved fields set to 0.

Case 3: Reserved LIDs (OF, M)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command specifying log identifiers of 00h, and 6Fh.

Observable Results:

1. Verify that each command completes with an error code of Invalid Log Page 09h. If the DUT claims support for NVMe specification version 1.3 or earlier, they DUT may also return an error code of ‘Invalid Field in Command’ (02h).
2. Verify that all received responses have all Reserved fields set to 0h.

Possible Problems: LID 00h is no longer reserved for devices that are NVM Express Base Specification, Revision 2.0 or greater. If a device properly supports LID 00h, then the LID 00h portion of this test is not applicable. If the device is NVM Express Base Specification, Revision 2.0 or greater and does not support LID 00h, then this test shall return a failing result.

© 2026 UNH–IOL

Case 4: NUMD/MDTS Conflict (OF, M)

Test Procedure:

1. Check the MDTS value reported by the DUT in the Identify Controller Data Structure. If MDTS is set to 0h (i.e. unlimited) then this test is not applicable.
2. Check the NVME Specification version supported by DUT using the value reported in the Identify Controller Data Structure VER field (83:80). If the DUT supports v1.2.1 or higher go to step 3. If the DUT supports v1.2a or lower (or VER is set to 0h, go to step 4).
3. Set the NUMDU and NUMDL values to be greater than the MDTS value provided by the DUT in the Identify Controller Data Structure. Configure the NVMe Host to issue a Get Log Page command specifying each of the Mandatory Log Page Identifiers (01h – Error Information, 02h – SMART/Health Information, 03h – Firmware Slot Information).
4. Set the NUMD values to be greater than the MDTS value provided by the DUT in the Identify Controller Data Structure. Configure the NVMe Host to issue a Get Log Page command specifying each of the Mandatory Log Page Identifiers. If NUMD cannot be set to a value greater than MDTS, or if MDTS = 0 (i.e. unlimited) then this test is not applicable.

Observable Results:

1. Verify that NVMe Controller returns Error 'Invalid Field in Command' (02h) in response to the received Get Log Page command.

Case 5: Get Error Information after Error (OF, M)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command to LID 01h, Error Information, and record the number of Error log entries.
2. Configure the NVMe Host to issue an Identify Command with a Reserved CNS value.
3. Check that the M bit is set in the Status Field of the completion queue entry associated with the Identify Command. If the M bit is not set to 1, this test is not applicable.
4. Configure the NVMe Host to issue a Get Log Page command to LID 01h, Error Information, and record the number of Error log entries.
5. Configure the NVMe Host to issue an Identify Command with a Reserved CNS value.
6. Check that the M bit is set in the Status Field of the completion queue entry associated with the Identify Command. If the M bit is not set to 1, this test is not applicable.
7. Configure the NVMe Host to issue a Get Log Page command to LID 01h, Error Information, and record the number of Error log entries.

Observable Results:

1. If the M bit is set, verify that NVMe Controller reported an error condition in the Error Information log, and the number of Error Log Entries is incremented by one or more. The reported error condition may or may not be directly related to the Identify Command with Reserved CNS value.

Case 6: SMART Temperature Threshold (OF, M)

Test Procedure:

1. Check the NVMe version (VER field) supported by the DUT.
2. Configure the NVMe Host to issue a Set Feature command to set the Critical Temperature Threshold below the current temperature of the DUT.
3. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h).

Observable Results:

1. Verify that NVMe Controller reported the Critical Temperature warning in the SMART/Health Information Log.

© 2026 UNH–IOL

2. If the DUT claims support for NVMe v1.4 or higher, verify that the Get Log Page – SMART / Health Status – Critical Warning Bit 1 is set to 1 only when a temperature is greater than or equal to an over temperature threshold; or less than or equal to an under temperature threshold.

Case 7: Data Units Read Count – Compare (OF, M)

Test Procedure:

1. Check ONCS bit 0 to determine if the DUT supports the Compare Command. If the Compare Command is not supported, this test is not applicable.
2. Configure the NVMe Host to issue a Identify Namespace Data Structure Command CNS=00 and record the value for LBA Data Size (LBADS) for LBA Format 0.
3. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Read value.
4. Perform 1000 NVMe READ command of LBADS bytes.
5. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Read value, which is expected to increase by LBADS/512. (i.e. if LBADS is 512, then the Data Units Read value should increase by 1).
6. Perform 1000 NVMe Compare command of LBADS bytes.
7. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Read value, which is expected to increase by LBADS/512. (i.e. if LBADS is 512, then the Data Units Read value should increase by 1). .
8. Perform 1000 NVMe READ command of 2x LBADS bytes.
9. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Read value, which is expected to increase by LBADS/512. (i.e. if LBADS is 512, then the Data Units Read value should increase by 2).
10. Perform 1000 NVMe Compare command of 2x LBADS bytes.
11. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Read value, which is expected to increase by LBADS/512. (i.e. if LBADS is 512, then the Data Units Read value should increase by 2).
12. Perform 1000 NVMe READ command of 4x LBADS bytes.
13. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Read value, which is expected to increase by LBADS/512. (i.e. if LBADS is 512, then the Data Units Read value should increase by 4).
14. Perform 1000 NVMe Compare command of 4x LBADS bytes.
15. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Read value, which is expected to increase by LBADS/512. (i.e. if LBADS is 512, then the Data Units Read value should increase by 4). .

Observable Results:

1. Verify that NVMe Controller reported the Data Units Read value that increased by 14x (LBADS/512) in accord with the NVMe READ and Compare commands that were performed.

Case 8: Data Units Written (FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue a Identify Namespace Data Structure Command CNS=00 and record the value for LBA Data Size (LBADS) for LBA Format 0.
2. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Written value.
3. Perform 1000 NVMe Write commands of LBADS bytes.
4. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Written value, which is expected to increase by LBADS/512. (i.e. if LBADS is 512, then the Data Units Written value should increase by 1).
5. Perform 1000 NVMe Write commands of 2x LBADS bytes.

© 2026 UNH–IOL

6. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Written value, which is expected to increase by 2x LBADS/512. (i.e. if LBADS is 512, then the Data Units Written value should increase by 2).
7. Perform 1000 NVMe Write command of 4x LBADS bytes.
8. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Written value, which is expected to increase by 4x LBADS/512. (i.e. if LBADS is 512, then the Data Units Written value should increase by 4).

Observable Results:

1. Verify that NVMe Controller reported the Data Units Written value that increased by 7x (LBADS/512) in accord with the NVMe Write Commands that were performed.

Case 9: Power Cycle Count (M)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h) and record the Power Cycles value.
2. Power Cycle the DUT.
3. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h) and record the Power Cycles value.

Observable Results:

1. Verify that NVMe Controller reported the Power Cycles value, and that the value incremented properly after performed a power cycle.

Case 10: NUMD Greater than Log Page Conflict (FYI)

Test Procedure:

1. Check the NVME Specification version supported by DUT using the value reported in the Identify Controller Data Structure VER field (83:80). If the DUT supports v1.2.1 or higher go to step 2. If the DUT supports v1.2a or lower (or VER is set to 0h, go to step 3).
2. Set the NUMDU and NUMDL values to be greater than the Log Page Size required for each of the Mandatory Log Page Identifiers.
3. Set the NUMD values to be greater than the Log Page Size required for each of the Mandatory Log Page Identifiers.

Observable Results:

1. Verify that NVMe Controller sends the requested Log Page, and that the size of the Log Page is defined by NUMD or NUMDU/NUMDL. Verify that valid data exists only between bytes 0 to the Log Page Size. Verify that bytes beyond MDTS up to NUMD or NUMDU/NUMDL are filled with undefined data.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.3.10 and expect the same observable results.

1. **LID 00h Supported Log Pages**
2. **LID 01h Error Information**
3. **LID 02h SMART/Health Information**
4. **LID 03h Firmware Slot Information**
5. **LID 04h Changed Attached Namespace List**
6. **LID 05h Commands Supported and Effects**
7. **LID 06h Device Self-test**
8. **LID 07h Telemetry Host-Initiated**
9. **LID 08h Telemetry Controller-Initiated**
10. **LID 09h Endurance Group Information**
11. **LID 0AhPredictable Latency Per NVM Set**

© 2026 UNH–IOL

12. **LID 0Bh Predictable Latency Event Aggregate**
13. **LID 0Ch Asymmetric Namespace Access**
14. **LID 0Dh Persistent Event Log**
15. **LID 0Eh LBA Status Information**
16. **LID 0Fh Endurance Group Event Aggregate**
17. **LID 10h Media Unit Status**
18. **LID 11h Supported Capacity Configuration List**
19. **LID 12h Feature Identifiers Supported and Effects**
20. **LID 13h NVMe-MI Commands Supported and Effects**
21. **LID 14h Command and Feature Lockdown**
22. **LID 15h Boot Partition**
23. **LID 16h Rotational Media Information**
24. **LID 70h Discovery**
25. **LID 80h Reservation Notification**
26. **LID 81h Sanitize Status**
27. **LID Downloadable Program Types List**

Sub-case Procedure:

1. This test is only applicable for devices that support the Computational Programs Command Set.

28. **LID 83h Downloadable Program Types List**

Sub-case Procedure:

1. This test is only applicable for devices that support the Computational Programs Command Set.

29. **LID 84h Memory Range Set List**

Sub-case Procedure:

1. This test is only applicable for devices that support the Computational Programs Command Set.

30. **LID 1Ch Changed Allocated Namespace List**
31. **LID 18h Management Address List**
32. **LID 25h Power Management**

Case 11: Telemetry Host Initiated Valid Offset Create=1 (OF-FYI, M)

Test Procedure:

1. Check the Bit 3 of the LPA field Identify Controller Data Structure to determine of the DUT supports Telemetry Host-Initiated and Telemetry Controller–Initiated log pages. If Bit 3 is set to 0 this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command specifying Telemetry Host-Initiated Log Page to the NVMe Controller with a Log Page Offset Lower value that is a multiple of 512 bytes, and the Create Telemetry Host-Initiated Data bit is set to 1.

Observable Results:

1. Verify that NVMe Controller sends the requested Log Page following the Telemetry Host-Initiated Log Page including the Telemetry Host-Initiated Data Block 1. Note that the Telemetry contents are vendor defined.
2. Verify that the Telemetry Host-Initiated Data Area 2 Last Block field is greater than or equal to the Telemetry Host-Initiated Data Area 1 Last Block field
3. Verify that the Telemetry Host-Initiated Data Area 3 Last Block field is greater than or equal to the Telemetry Host-Initiated Data Area 2 Last Block field.
4. If present, verify that the IEEE OUI Identifier field is a valid IEEE/RAC assigned identifier that is registered at http://standards.ieee.org/develop/regauth/oui/public.html.

Case 12: Telemetry Host Initiated Valid Offset Create=0 (OF-FYI, M)

Test Procedure:

© 2026 UNH–IOL

1. Check the Bit 3 of the LPA field Identify Controller Data Structure to determine of the DUT supports Telemetry Host-Initiated and Telemetry Controller–Initiated log pages. If Bit 3 is set to 0 this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command specifying Telemetry Host-Initiated Log Page to the NVMe Controller with a Log Page Offset Lower value that is a multiple of 512 bytes, and the Create Telemetry Host-Initiated Data bit is set to 1.
3. Configure the NVMe Host to issue a Get Log Page command specifying Telemetry Host-Initiated Log Page to the NVMe Controller with a Log Page Offset Lower value that is a multiple of 512 bytes, and the Create Telemetry Host-Initiated Data bit is set to 0.
4. Configure the NVMe Host to issue a Get Log Page command specifying Telemetry Host-Initiated Log Page to the NVMe Controller with a Log Page Offset Lower value that is a multiple of 512 bytes, and the Create Telemetry Host-Initiated Data bit is set to 1.

Observable Results:

1. Verify that in steps 2, 3, and 4 the NVMe Controller sends the requested Log Page following the Telemetry Host-Initiated Log Page format defined in the NVMe specification.
2. Verify that the Log Page delivered by the NVMe Controller in Step 3 was not updated from the Log Page delivered by the Controller in Step 2, and that the Telemetry Host-Initiated Data Generation Number does not increments.

Case 13: Telemetry Host Initiated Invalid Offset (OF-FYI, M)

Test Procedure:

1. Check the Bit 3 of the LPA field Identify Controller Data Structure to determine of the DUT supports Telemetry Host-Initiated and Telemetry Controller–Initiated log pages. If Bit 3 is set to 0h this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command specifying Telemetry Host-Initiated Log Page to the NVMe Controller with a Log Page Offset Lower value that is not a multiple of 512 bytes.

Observable Results:

1. Verify that NVMe Controller responds to the Get Log Page command with an error of ‘Invalid Field in Command’ (02h).

Case 14: Telemetry Controller Initiated Valid Offset (OF-FYI, M)

Test Procedure:

1. Check the Bit 3 of the LPA field Identify Controller Data Structure to determine of the DUT supports Telemetry Host-Initiated and Telemetry Controller–Initiated log pages. If Bit 3 is set to 0 this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command specifying Telemetry Controller-Initiated Log Page to the NVMe Controller with a Log Page Offset Lower value that is a multiple of 512 bytes.

Observable Results:

1. Verify that NVMe Controller sends the requested Log Page following the Telemetry Controller-Initiated Log Page format defined in the NVMe specification.

Case 15: Telemetry Controller Initiated Invalid Offset (OF-FYI, M)

Test Procedure:

1. Check the Bit 3 of the LPA field Identify Controller Data Structure to determine of the DUT supports Telemetry Host-Initiated and Telemetry Controller–Initiated log pages. If Bit 3 is set to 0h this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command specifying Telemetry Controller-Initiated Log Page to the NVMe Controller with a Log Page Offset Lower value that is not a multiple of 512 bytes.

Observable Results:

1. Verify that NVMe Controller responds to the Get Log Page command with an error of ‘Invalid Field in Command’ (02h).

© 2026 UNH–IOL

Case 16: Telemetry Host Initiated Valid 0 Offset Create=1 (OF-FYI, M)

Test Procedure:

1. Check the Bit 3 of the LPA field Identify Controller Data Structure to determine of the DUT supports Telemetry Host-Initiated and Telemetry Controller–Initiated log pages. If Bit 3 is set to 0 this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command specifying Telemetry Host-Initiated Log Page to the NVMe Controller with a Log Page Offset of 0, and the Create Telemetry Host-Initiated Data bit is set to 1.

Observable Results:

1. Verify that NVMe Controller sends the requested Log Page using the correct formatting of the first 512 bytes of the Telemetry Host-Initiated Log Page including the reserved bytes.

Case 17: Data Units Written Does not Increment for Write Uncorrectable (OF, M)

Test Procedure:

1. Configure the NVMe Host to issue a Identify Namespace Data Structure Command CNS=00 and record the value for LBA Data Size (LBADS) for LBA Format 0.
2. Configure the NVMe Host to issue a Identify Controller Data Structure Command CNS=01 and record the value for ONCS Bit 1. If ONCS Bit 1 is set to 0, then the DUT does not support the Write Uncorrectable Command and this test is not applicable. If ONCS Bit 1 is set 1, then proceed to the next step.
3. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Written value.
4. Perform 1000 NVMe Write Uncorrectable commands of LBADS bytes.
5. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Written value, which is expected to be unchanged.
6. Perform 1000 NVMe Write Uncorrectable commands of 2x LBADS bytes.
7. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Written value, which is expected to increase by 2x LBADS/512. (i.e. if LBADS is 512, then the Data Units Written value, which is expected to be unchanged.
8. Perform 1000 NVMe Write Uncorrectable commands of 4x LBADS bytes.
9. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Written value, which is expected to increase by 4x LBADS/512. (i.e. if LBADS is 512, then the Data Units Written value, which is expected to be unchanged.
10. Perform a Write operation to all LBAs that a Write Uncorrectable operation was performed to enable those blocks to be read in future test operations.

Observable Results:

1. Verify that NVMe Controller reported the Data Units Written value that did not increase when the Write Uncorrectable commands were performed, and this was properly reflected in each SMART/Health Information Log Page returned by the DUT.

Case 18: Persistent Event Log (OF-FYI, M)

Test Procedure:

1. Check Bit 4 of the Log Page Attributes field in the Identify Controller Data Structure. If Bit 4 is set to 0h, then this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page Command for Log Identifier 0Dh “Persistent Event”.
3. Read the supported events bitmap returned in Log Page 0Dh, then cause the testing station to perform one of the events supported by the DUT. Common events supported may be:
    a. Set Feature
    b. Sanitize
    c. Format NVM
    d. Change Namespace

© 2026 UNH–IOL

e. Power on or Reset.

4. If no events that can be performed by the test station are supported, then this test is not applicable.
5. Configure the NVMe Host to issue a Get Log Page Command for Log Identifier 0Dh “Persistent Event”.
6. Perform a Controller Level Reset.
7. Configure the NVMe Host to issue a Get Log Page Command for Log Identifier 0Dh “Persistent Event”.

Observable Results:

1. Verify that all Get Log Page commands completed with status ‘Success’.
2. Verify that the Log page contents indicated that a supported event was performed, after the testing station performed one of the supported events.
3. Verify that the Log Page contents were not reset after the Controller Level Reset.

Case 19: Data Units Read Count – Verify (OF-FYI, M)

Test Procedure:

1. Check ONCS bit 7 to determine if the DUT supports the Verify Command. If the Verify Command is not supported, this test is not applicable.
2. Configure the NVMe Host to issue a Identify Namespace Data Structure Command CNS=00 and record the value for LBA Data Size (LBADS) for LBA Format 0.
3. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Read value.
4. Perform 1000 Verify commands of LBADS bytes, using the same protection information as the namespace was formatted with.
5. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Read value, which is expected to increase by LBADS/512. (i.e. if LBADS is 512, then the Data Units Read value should increase by 1).
6. Perform 1000 Verify commands of 2x LBADS bytes, using the same protection information as the namespace was formatted with.
7. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Read value, which is expected to increase by LBADS/512. (i.e. if LBADS is 512, then the Data Units Read value should increase by 2).
8. Perform 1000 Verify commands of 4x LBADS bytes, using the same protection information as the namespace was formatted with.
9. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Read value, which is expected to increase by LBADS/512. (i.e. if LBADS is 512, then the Data Units Read value should increase by 4).

Observable Results:

1. Verify that NVMe Controller reported the Data Units Read value that increased by 7x LBADS/512 in accord with the NVMe Verify commands that were performed.

Case 20: Data Units Read Count – Read Only (OF, M)

Test Procedure:

1. Configure the NVMe Host to issue a Identify Namespace Data Structure Command CNS=00 and record the value for LBA Data Size (LBADS) for LBA Format 0.
2. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Read value.
3. Perform 1000 NVMe READ command of LBADS bytes.
4. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Read value, which is expected to increase by LBADS/512. (i.e. if LBADS is 512, then the Data Units Read value should increase by 1). .
5. Perform 1000 NVMe READ command of 2x LBADS bytes.

© 2026 UNH–IOL

6. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Read value, which is expected to increase by LBADS/512. (i.e. if LBADS is 512, then the Data Units Read value should increase by 2).
7. Perform 1000 NVMe READ command of 4x LBADS bytes.
8. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Read value, which is expected to increase by LBADS/512. (i.e. if LBADS is 512, then the Data Units Read value should increase by 4).

Observable Results:

1. Verify that NVMe Controller reported the Data Units Read value that increased by 7x (LBADS/512) in accord with the NVMe READ commands that were performed.

Case 21: Data Units Written Does not Increment for Write Zeroes (OF, M)

Test Procedure:

1. Configure the NVMe Host to issue a Identify Namespace Data Structure Command CNS=00 and record the value for LBA Data Size (LBADS) for LBA Format 0.
2. Configure the NVMe Host to issue a Identify Controller Data Structure Command CNS=01 and record the value for ONCS Bit 1. If ONCS Bit 1 is set to 0, then the DUT does not support the Write Uncorrectable Command and this test is not applicable. If ONCS Bit 1 is set 1, then proceed to the next step.
3. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Written value.
4. Perform 1000 NVMe Write Zeroes commands of LBADS bytes.
5. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Written value, which is expected to be unchanged.
6. Perform 1000 NVMe Write Zeroes commands of 2x LBADS bytes.
7. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Written value, which is expected to be unchanged.
8. Perform 1000 NVMe Write Zeroes commands of 4x LBADS bytes.
9. Configure the NVMe Host to issue a Get Log Page command to LID 02h, SMART/Health Information (02h). and record the Data Units Written value, which is expected to be unchanged.

Observable Results:

1. Verify that NVMe Controller reported the Data Units Written value that did not increase when the Write Zeroes commands were performed, and this was properly reflected in each SMART/Health Information Log Page returned by the DUT.

Case 22: Invalid LPOL offset (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command specifying each of the Mandatory Log Page Identifiers described in Table 3 and 4 above, to the NVMe Controller. For each mandatory Log Page requested, the LPOL field in the Get Log Page Request should be set to a value greater than the size of the log page requested.

Observable Results:

1. Verify that the Get Log Page command is aborted with status ‘Invalid Field in Command’ (02h).

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.3.22 and expect the same observable results.

1. **LID 00h Supported Log Pages**
2. **LID 01h Error Information**
3. **LID 02h SMART/Health Information**
4. **LID 03h Firmware Slot Information**

© 2026 UNH–IOL

5. **LID 04h Changed Attached Namespace List**
6. **LID 05h Commands Supported and Effects**
7. **LID 06h Device Self-test**
8. **LID 07h Telemetry Host-Initiated**
9. **LID 08h Telemetry Controller-Initiated**
10. **LID 09h Endurance Group Information**
11. **LID 0Ah Predictable Latency Per NVM Set**
12. **LID 0Bh Predictable Latency Event Aggregate**
13. **LID 0Ch Asymmetric Namespace Access**
14. **LID 0Dh Persistent Event Log**
15. **LID 0Eh LBA Status Information**
16. **LID 0Fh Endurance Group Event Aggregate**
17. **LID 10h Media Unit Status**
18. **LID 11h Supported Capacity Configuration List**
19. **LID 12h Feature Identifiers Supported and Effects**
20. **LID 13h NVMe-MI Commands Supported and Effects**
21. **LID 14h Command and Feature Lockdown**
22. **LID 15h Boot Partition**
23. **LID 16h Rotational Media Information**
24. **LID 70h Discovery**
25. **LID 80h Reservation Notification**
26. **LID 81h Sanitize Status**
27. **LID 82h Program List**

Sub-case Procedure:

1. This test is only applicable for devices that support the Computational Programs Command Set.

28. **LID 83h Downloadable Program Types List**

Sub-case Procedure:

1. This test is only applicable for devices that support the Computational Programs Command Set.

29. **LID 84h Memory Range Set List**

Sub-case Procedure:

1. This test is only applicable for devices that support the Computational Programs Command Set.

30. **LID 1Ch Changed Allocated Namespace List**
31. **LID 18h Management Address List**
32. **LID 25h Power Management**

Case 23: Invalid LPOU offset (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command specifying each of the Mandatory Log Page Identifiers described in Table 3 and 4 above, to the NVMe Controller. For each mandatory Log Page requested, the LPOU field in the Get Log Page Request should be set to a value greater than the size of the log page requested.

Observable Results:

1. Verify that the Get Log Page command is aborted with status ‘Invalid Field in Command’ (02h).

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.3.23 and expect the same observable results.

1. **LID 00h Supported Log Pages**
2. **LID 01h Error Information**

© 2026 UNH–IOL

3. **LID 02h SMART/Health Information**
4. **LID 03h Firmware Slot Information**
5. **LID 04h Changed Attached Namespace List**
6. **LID 05h Commands Supported and Effects**
7. **LID 06h Device Self-test**
8. **LID 07h Telemetry Host-Initiated**
9. **LID 08h Telemetry Controller-Initiated**
10. **LID 09h Endurance Group Information**
11. **LID 0Ah Predictable Latency Per NVM Set**
12. **LID 0Bh Predictable Latency Event Aggregate**
13. **LID 0Ch Asymmetric Namespace Access**
14. **LID 0Dh Persistent Event Log**
15. **LID 0Eh LBA Status Information**
16. **LID 0Fh Endurance Group Event Aggregate**
17. **LID 10h Media Unit Status**
18. **LID 11h Supported Capacity Configuration List**
19. **LID 12h Feature Identifiers Supported and Effects**
20. **LID 13h NVMe-MI Commands Supported and Effects**
21. **LID 14h Command and Feature Lockdown**
22. **LID 15h Boot Partition**
23. **LID 16h Rotational Media Information**
24. **LID 70h Discovery**
25. **LID 80h Reservation Notification**
26. **LID 81h Sanitize Status**
27. **LID 82h Downloadable Program Types List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives.

28. **LID 83h Memory Range Set List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives.

29. **LID 84h Memory Range Set List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives.

30. **LID 1Ch Changed Allocated Namespace List**
31. **LID 18h Management Address List**
32. **LID 25h Power Management**

Case 24: Domain Identifier (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to query the Identifier Controller Data Structure MDS bit, Bit 10 (MDS) and Bit 4 (Endurance Groups) of the CTRATT field.
2. If the CTRATT Bit 4 (Endurance Groups) is set to 0, this test is not applicable.
3. Configure the NVMe Host to issue a Get Log Page command for the Endurance Group Log (LID=09h).

Observable Results:

1. Verify that if the MDS bit is set to 0, that the Domain Identifier in the Endurance Group Log page is set to 0 also.

© 2026 UNH–IOL

2. Verify that if the MDS bit is set to 1, that the Domain Identifier in the Endurance Group Log page is set to a non zero value.

Case 25: Feature Identifiers Supports and Effects Log Page (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Feature Identifiers Supported and Effects Log Page (LID=12h). If this LID is not supported then this test is not applicable.

Observable Results:

1. Verify that the FID Scope (FSP) field for any given feature, shall have no more than 1 bit set to '1', or else be set to 0.
2. Verify that if the FID Supported bit (FSUPP) is set to 0 for any given feature, then all fields in the FID Supported and Effects Data Structure for that feature shall be set to 0.

Case 26: NVMe-MI Commands Supported and Effects Log Page (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the NVMe-MI Commands Supported and Effects Log Page (LID=13h). If this LID is not supported in version 1.4 or earlier then this test is not applicable.

Observable Results:

1. Verify the Commands Support and Effects Log Page (LID=13h) completes successfully if the DUT is 2.0 or later.
2. Verify that the Command Scope (CSP) field for any given command, shall have no more than 1 bit set to '1', or else be set to 0.
3. Verify that if the Command Supported bit (CSUPP) is set to 0 for any given feature, then all fields in the NVME-MI Commands Supported and Effects Data Structure for that command shall be set to 0.

Possible Problems: None known.

Case 27: Persistent Event Log Log Revision (OF-FYI, M)

Test Procedure:

1. If the DUT does not support NVMe v1.4 or higher, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page Command for Log Identifier 0Dh "Persistent Event".
3. Read the Log Revision field in the Persistent Event Log Header.

Observable Results:

1. Verify that all Get Log Page commands completed with status 'Success'.
2. Verify that the Log Revision is set to 03h

Possible Problems: None known.

Case 28: Persistent Event Log Generation Number (OF-FYI, M)

Test Procedure:

1. If the DUT does not support NVMe v2.0 or higher, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page Command for Log Identifier 0Dh "Persistent Event".
3. Read the Generation Number field in the Persistent Event Log Header.

Observable Results:

1. Verify that all Get Log Page commands completed with status 'Success'.
2. Verify that the Generation Number is no larger than FFFFh.

© 2026 UNH–IOL

Possible Problems: None known.

Case 29: Log Page Offset, Offset Type=0 (FYI, M)

Test Procedure:

1. Check Bit 2 of the Log Page Attributes (LPA) field of the Identify Controller Data structure to determine if the DUT supports the Log Page Offset. If the LPA bit 2 is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command specifying each of the Mandatory Log Page Identifiers described in Table 3 and Table 4 above, to the NVMe Controller, using an Offset Type (OT) bit that is cleared to '0h'.

Observable Results:

1. Verify that the Identify completes successfully.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.3.29 and expect the same observable results.

1. **LID 00h Supported Log Pages**
2. **LID 01h Error Information**
3. **LID 02h SMART/Health Information**
4. **LID 03h Firmware Slot Information**
5. **LID 04h Changed Attached Namespace List**
6. **LID 05h Commands Supported and Effects**
7. **LID 06h Device Self-test**
8. **LID 07h Telemetry Host-Initiated**
9. **LID 08h Telemetry Controller-Initiated**
10. **LID 09h Endurance Group Information**
11. **LID 0Ah Predictable Latency Per NVM Set**
12. **LID 0Bh Predictable Latency Event Aggregate**
13. **LID 0Ch Asymmetric Namespace Access**
14. **LID 0Dh Persistent Event Log**
15. **LID 0Eh LBA Status Information**
16. **LID 0Fh Endurance Group Event Aggregate**
17. **LID 10h Media Unit Status**
18. **LID 11h Supported Capacity Configuration List**
19. **LID 12h Feature Identifiers Supported and Effects**
20. **LID 13h NVMe-MI Commands Supported and Effects**

    Sub-case Procedure:

    1. LID 13h is only mandatory to support if the NVMe-MI Send and NVMe-MI Receive commands are supported. If these commands are not supported by the controller under test, then this subcase is not applicable.

21. **LID 14h Command and Feature Lockdown**
22. **LID 15h Boot Partition**
23. **LID 16h Rotational Media Information**
24. **LID 70h Discovery**
25. **LID 80h Reservation Notification**
26. **LID 81h Sanitize Status**
27. **LID 82h Downloadable Program Types List**

    Sub-case Procedure:

    1. This test is only applicable for Computational Storage drives.

28. **LID 83h Memory Range Set List**

© 2026 UNH–IOL

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives.

29. **LID 84h Memory Range Set List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives

30. **LID 1Ch Changed Allocated Namespace List**
31. **LID 18h Management Address List**
32. **LID 25h Power Management**

Case 30: Log Page Offset, Offset Type=1 (FYI, FYI)

Test Procedure:

1. Check Bit 2 of the Log Page Attributes (LPA) field of the Identify Controller Data structure to determine if the DUT supports the Log Page Offset. If the LPA bit 2 is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page Identify command for LID 00h. If there are no LIDs that have the IOS bit set to '1', this test is not applicable.
3. Configure the NVMe Host to issue a Get Log Page command specifying each of the LIDs returned in step 2 that have the IOS bit set to '1', to the NVMe Controller, using an Offset Type (OT) set to '1'.

Observable Results:

1. Verify that the Get Log Page completes successfully

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.3.30 and expect the same observable results.

1. **LID 00h Supported Log Pages**
2. **LID 01h Error Information**
3. **LID 02h SMART/Health Information**
4. **LID 03h Firmware Slot Information**
5. **LID 04h Changed Attached Namespace List**
6. **LID 05h Commands Supported and Effects**
7. **LID 06h Device Self-test**
8. **LID 07h Telemetry Host-Initiated**
9. **LID 08h Telemetry Controller-Initiated**
10. **LID 09h Endurance Group Information**
11. **LID 0Ah Predictable Latency Per NVM Set**
12. **LID 0Bh Predictable Latency Event Aggregate**
13. **LID 0Ch Asymmetric Namespace Access**
14. **LID 0Dh Persistent Event Log**
15. **LID 0Eh LBA Status Information**
16. **LID 0Fh Endurance Group Event Aggregate**
17. **LID 10h Media Unit Status**
18. **LID 11h Supported Capacity Configuration List**
19. **LID 12h Feature Identifiers Supported and Effects**
20. **LID 13h NVMe-MI Commands Supported and Effects**
21. **LID 14h Command and Feature Lockdown**
22. **LID 15h Boot Partition**
23. **LID 16h Rotational Media Information**
24. **LID 70h Discovery**
25. **LID 80h Reservation Notification**
26. **LID 81h Sanitize Status**
27. **LID 82h Program List**

© 2026 UNH–IOL

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives.

28. LID 83h Downloadable Program Types List

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives.

29. LID 84h Memory Range Set List

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives.

30. LID 1Ch Changed Allocated Namespace List
31. LID 18h Management Address List
32. LID 25h Power Management

Case 31: Log Page Offset, Invalid Field in Command (FYI, M)

Test Procedure:

1. Check Bit 2 of the Log Page Attributes (LPA) field of the Identify Controller Data structure to determine if the DUT supports the Log Page Offset. If the LPA bit 2 is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command specifying log identifier of 00h, determine that a specified LID has the Index Offset Supported (IOS) bit cleared to 0 in the LID Supported and Effects Data Structure.
3. Configure the NVMe Host to issue a Get Log Page command on the LID having IOS cleared to 0, with the Offset Type field set to 1

Observable Results:

1. Verify that the first Get Log Page completes successfully
2. Verify that the second Get Log Page is aborted with a status code of 'Invalid Field in Command'

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.3.31 and expect the same observable results.

1. LID 00h Supported Log Pages
2. LID 01h Error Information
3. LID 02h SMART/Health Information
4. LID 03h Firmware Slot Information
5. LID 04h Changed Attached Namespace List
6. LID 05h Commands Supported and Effects
7. LID 06h Device Self-test
8. LID 07h Telemetry Host-Initiated
9. LID 08h Telemetry Controller-Initiated
10. LID 09h Endurance Group Information
11. LID 0Ah Predictable Latency Per NVM Set
12. LID 0Bh Predictable Latency Event Aggregate
13. LID 0Ch Asymmetric Namespace Access
14. LID 0Dh Persistent Event Log
15. LID 0Eh LBA Status Information
16. LID 0Fh Endurance Group Event Aggregate
17. LID 10h Media Unit Status
18. LID 11h Supported Capacity Configuration List
19. LID 12h Feature Identifiers Supported and Effects
20. LID 13h NVMe-MI Commands Supported and Effects
21. LID 14h Command and Feature Lockdown

© 2026 UNH–IOL

22. **LID 15h Boot Partition**
23. **LID 16h Rotational Media Information**
24. **LID 70h Discovery**
25. **LID 80h Reservation Notification**
26. **LID 81h Sanitize Status**
27. **LID 82h Program List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives

28. **LID 83h Downloadable Program Types List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives

29. **LID 84h Memory Range Set List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives

30. **LID 1Ch Changed Allocated Namespace List**
31. **LID 18h Management Address List**
32. **LID 25h Power Management**

Case 32: LPOL, OT=0 (FYI, M)

Test Procedure:

1. Check Bit 2 of the Log Page Attributes (LPA) field of the Identify Controller Data structure to determine if the DUT supports the Log Page Offset. If the LPA bit 2 is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command specifying each of the Mandatory Log Page Identifiers described in Table 3 and Table 4 above, to the NVMe Controller. For each mandatory Log Page requested, the OT bit is cleared to 0. Have the bits 1:0 set to a non-zero value.

Observable Results:

1. Verify that the first Get Log Page command completes successfully.

Possible Problems: Controller may return status of 'Invalid Field in Command' (02h).

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.3.32 and expect the same observable results.

1. **LID 00h Supported Log Pages**
2. **LID 01h Error Information**
3. **LID 02h SMART/Health Information**
4. **LID 03h Firmware Slot Information**
5. **LID 04h Changed Attached Namespace List**
6. **LID 05h Commands Supported and Effects**
7. **LID 06h Device Self-test**
8. **LID 07h Telemetry Host-Initiated**
9. **LID 08h Telemetry Controller-Initiated**
10. **LID 09h Endurance Group Information**
11. **LID 0Ah Predictable Latency Per NVM Set**
12. **LID 0Bh Predictable Latency Event Aggregate**
13. **LID 0Ch Asymmetric Namespace Access**
14. **LID 0Dh Persistent Event Log**
15. **LID 0Eh LBA Status Information**
16. **LID 0Fh Endurance Group Event Aggregate**

© 2026 UNH–IOL

17. **LID 10h Media Unit Status**
18. **LID 11h Supported Capacity Configuration List**
19. **LID 12h Feature Identifiers Supported and Effects**
20. **LID 13h NVMe-MI Commands Supported and Effects**
21. **LID 14h Command and Feature Lockdown**
22. **LID 15h Boot Partition**
23. **LID 16h Rotational Media Information**
24. **LID 70h Discovery**
25. **LID 80h Reservation Notification**
26. **LID 81h Sanitize Status**
27. **LID 82h Program List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives

28. **LID 83h Downloadable Program Types List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives

29. **LID 84h Memory Range Set List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives

30. **LID 1Ch Changed Allocated Namespace List**
31. **LID 18h Management Address List**
32. **LID 25h Power Management**

Case 33: Invalid LPOU & LPOL, OT=0 (OF-FYI, M)

Test Procedure:

1. Check Bit 2 of the Log Page Attributes (LPA) field of the Identify Controller Data structure to determine if the DUT supports the Log Page Offset. If the LPA bit 2 is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command specifying each of the Mandatory Log Page Identifiers described in Table 3 and Table 4 above, to the NVMe Controller, with the OT bit cleared to 0h and the offset (LPOU and LPOL fields) should be set to a value greater than the size of the log page requested.

Observable Results:

1. Verify that the Get Log Page command is aborted with status ‘Invalid Field in Command’ (02h).

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.3.33 and expect the same observable results.

1. **LID 00h Supported Log Pages**
2. **LID 01h Error Information**
3. **LID 02h SMART/Health Information**
4. **LID 03h Firmware Slot Information**
5. **LID 04h Changed Attached Namespace List**
6. **LID 05h Commands Supported and Effects**
7. **LID 06h Device Self-test**
8. **LID 07h Telemetry Host-Initiated**
9. **LID 08h Telemetry Controller-Initiated**
10. **LID 09h Endurance Group Information**
11. **LID 0Ah Predictable Latency Per NVM Set**
12. **LID 0Bh Predictable Latency Event Aggregate**
13. **LID 0Ch Asymmetric Namespace Access**

© 2026 UNH–IOL

14. **LID 0Dh Persistent Event Log**
15. **LID 0Eh LBA Status Information**
16. **LID 0Fh Endurance Group Event Aggregate**
17. **LID 10h Media Unit Status**
18. **LID 11h Supported Capacity Configuration List**
19. **LID 12h Feature Identifiers Supported and Effects**
20. **LID 13h NVMe-MI Commands Supported and Effects**

Sub-case Procedure:

1. LID 13h is only mandatory to support if the NVMe-MI Send and NVMe-MI Receive commands are supported. If these commands are not supported by the controller under test, then this subcase is not applicable.

21. **LID 14h Command and Feature Lockdown**
22. **LID 15h Boot Partition**
23. **LID 16h Rotational Media Information**
24. **LID 70h Discovery**
25. **LID 80h Reservation Notification**
26. **LID 81h Sanitize Status**
27. **LID 82h Program List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives

28. **LID 83h Downloadable Program Types List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives

29. **LID 84h Memory Range Set List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives

30. **LID 1Ch Changed Allocated Namespace List**
31. **LID 18h Management Address List**
32. **LID 25h Power Management**

Case 34: Invalid LPOU & LPOL, OT=1 (OF-FYI, M)

Test Procedure:

1. Check Bit 2 of the Log Page Attributes (LPA) field of the Identify Controller Data structure to determine if the DUT supports the Log Page Offset. If the LPA bit 2 is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command specifying each of the Mandatory Log Page Identifiers described in Table 3 and Table 4 above, to the NVMe Controller. For each mandatory Log Page with the OT bit set to 1h and the host specifies an index (LPOU and LPOL fields) in the Get Log Page Request be set to a value greater than the number of entries in the list of data structures in the log page requested.

Observable Results:

1. Verify that the Get Log Page command is aborted with status ‘Invalid Field in Command’ (02h).

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.3.34 and expect the same observable results.

1. **LID 00h Supported Log Pages**
2. **LID 01h Error Information**
3. **LID 02h SMART/Health Information**
4. **LID 04h Changed Attached Namespace List**
5. **LID 03h Firmware Slot Information**

© 2026 UNH–IOL

6. **LID 05h Commands Supported and Effects**
7. **LID 06h Device Self-test**
8. **LID 07h Telemetry Host-Initiated**
9. **LID 08h Telemetry Controller-Initiated**
10. **LID 09h Endurance Group Information**
11. **LID 0Ah Predictable Latency Per NVM Set**
12. **LID 0Bh Predictable Latency Event Aggregate**
13. **LID 0Ch Asymmetric Namespace Access**
14. **LID 0Dh Persistent Event Log**
15. **LID 0Eh LBA Status Information**
16. **LID 0Fh Endurance Group Event Aggregate**
17. **LID 10h Media Unit Status**
18. **LID 11h Supported Capacity Configuration List**
19. **LID 12h Feature Identifiers Supported and Effects**
20. **LID 13h NVMe-MI Commands Supported and Effects**
21. **LID 14h Command and Feature Lockdown**
22. **LID 15h Boot Partition**
23. **LID 16h Rotational Media Information**
24. **LID 70h Discovery**
25. **LID 80h Reservation Notification**
26. **LID 81h Sanitize Status**
27. **LID 82h Program List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives

28. **LID 83h Downloadable Program Types List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives

29. **LID 84h Memory Range Set List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives

30. **LID 1Ch Changed Allocated Namespace List**
31. **LID 18h Management Address List**
32. **LID 25h Power Management**

Case 35: Invalid LPOU & LPOL, OT=1, IOS=0 (OF-FYI, M)

Test Procedure:

1. Check Bit 2 of the Log Page Attributes (LPA) field of the Identify Controller Data structure to determine if the DUT supports the Log Page Offset. If the LPA bit 2 is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command specifying log identifier of 00h, determine that a specified LID has the Index Offset Supported (IOS) bit cleared to 0h in the LID Supported and Effects Data Structure.
3. Configure the NVMe Host to issue a Get Log Page command on the LIDs having IOS set to ‘0h’, with the Offset Type field set to 1h.

Observable Results:

1. Verify that the Get Log Page command is aborted with status ‘Invalid Field in Command’ (02h).

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.3.35 and expect the same observable results.

© 2026 UNH–IOL

1. **LID 00h Supported Log Pages**
2. **LID 01h Error Information**
3. **LID 02h SMART/Health Information**
4. **LID 03h Firmware Slot Information**
5. **LID 04h Changed Attached Namespace List**
6. **LID 05h Commands Supported and Effects**
7. **LID 06h Device Self-test**
8. **LID 07h Telemetry Host-Initiated**
9. **LID 08h Telemetry Controller-Initiated**
10. **LID 09h Endurance Group Information**
11. **LID 0Ah Predictable Latency Per NVM Set**
12. **LID 0Bh Predictable Latency Event Aggregate**
13. **LID 0Ch Asymmetric Namespace Access**
14. **LID 0Dh Persistent Event Log**
15. **LID 0Eh LBA Status Information**
16. **LID 0Fh Endurance Group Event Aggregate**
17. **LID 10h Media Unit Status**
18. **LID 11h Supported Capacity Configuration List**
19. **LID 12h Feature Identifiers Supported and Effects**
20. **LID 13h NVMe-MI Commands Supported and Effects**
21. **LID 14h Command and Feature Lockdown**
22. **LID 15h Boot Partition**
23. **LID 16h Rotational Media Information**
24. **LID 70h Discovery**
25. **LID 80h Reservation Notification**
26. **LID 81h Sanitize Status**
27. **LID 82h Program List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives

28. **LID 83h Downloadable Program Types List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives

29. **LID 84h Memory Range Set List**

Sub-case Procedure:

1. This test is only applicable for Computational Storage drives

30. **LID 1Ch Changed Allocated Namespace List**
31. **LID 18h Management Address List**
32. **LID 25h Power Management**

Case 36: Telemetry Host-initiated, Not Extended (OF-FYI, M)

Test Procedure:

1. Check the Bit 3 of the LPA field Identify Controller Data Structure to determine of the DUT supports Telemetry Host-Initiated and Telemetry Controller–Initiated log pages. If Bit 3 is set to 0 this test is not applicable
2. Check the Bit 6 of the LPA field Identify Controller Data Structure to determine of the DUT supports Data Area 4 of Host and Controller log, if bit 6 is set to ‘1’, skip test
3. Configure the NVMe Host to issue a Telemetry Log Page Command for Host-Initiated Data (LID=07h).

Observable Results:

© 2026 UNH–IOL

1. Verify that the data beyond the last block in the Telemetry Host-Initiated Data Area 3 Last Block is undefined

Case 37: Telemetry Host-Initiated, bit 6=1, ETDAS=0 (OF-FYI, M)

Test Procedure:

1. Check the Bit 3 of the LPA field Identify Controller Data Structure to determine of the DUT supports Telemetry Host-Initiated and Telemetry Controller–Initiated log pages. If Bit 3 is set to 0 this test is not applicable
2. Check the Bit 6 of the LPA field Identify Controller Data Structure to determine of the DUT supports Data Area 4 of Host and Controller log, if set to '0' skip test
3. Check the Extended Telemetry Data Area 4 Supported (ETDAS) field in the Host Behavior Support feature is, if set to 1 skip test
4. Configure the NVMe Host to issue a Telemetry Log Page Command for Host-Initiated Data (LID=07h).

Observable Results:

1. Verify that if Bit 6 is set to '1' and ETDAS is set to '0', that the last block in the returned Telemetry Host-Initiated Data Area 3, from the Get Log Page command, is undefined.

Case 38: Telemetry Host-Initiated, bit 6=1, ETDAS=1 (OF-FYI, FYI)

Test Procedure:

1. Check the Bit 3 of the LPA field Identify Controller Data Structure to determine of the DUT supports Telemetry Host-Initiated and Telemetry Controller–Initiated log pages. If Bit 3 is set to 0h this test is not applicable.
2. Check the Bit 6 of the LPA field Identify Controller Data Structure to determine of the DUT supports Data Area 4 of Host and Controller log, if set to '0h' skip test.
3. Check the Extended Telemetry Data Area 4 Supported (ETDAS) field in the Host Behavior Support feature, if set to '0h' skip test.
4. Configure the NVMe Host to issue a Telemetry Log Page Command for Host-Initiated Data (LID=07h).

Observable Results:

1. Verify that if Bit 6 is set to '1h' and ETDAS is set to '1h', then the data beyond that the last block in the returned Telemetry Host-Initiated Data Area 4, from the Get Log Page command, is undefined.

Case 39: Telemetry Host-Initiated, Last Block field (OF-FYI, FYI)

Test Procedure:

1. Check the Bit 3 of the LPA field Identify Controller Data Structure to determine of the DUT supports Telemetry Host-Initiated and Telemetry Controller–Initiated log pages. If Bit 3 is set to 0 this test is not applicable
2. Check the Bit 6 of the LPA field Identify Controller Data Structure to determine of the DUT supports Data Area 4 of Host and Controller log, if Bit 6 is cleared to '0' skip this test
3. Configure the NVMe Host to issue a Telemetry Log Page Command for Host-Initiated Data (LID=07h).

Observable Results:

1. Verify that the Get Log Page Command completed successfully and retrieve the Telemetry Host-Initiated Data Area 4 Last Block field
2. Verify that if Bit 6 is set to '1' that the value of the Telemetry Host-Initiated Data Area 4 Last Block field is greater than or equal to the value in the Telemetry Host-initiated Data Area 3 Last Block field

Case 40: Telemetry Controller-initiated, LPA bit 6=0 (OF-FYI, M)

Test Procedure:

1. Check the Bit 3 of the LPA field Identify Controller Data Structure to determine of the DUT supports Telemetry Host-Initiated and Telemetry Controller–Initiated log pages. If Bit 3 is set to 0 this test is not applicable
2. Check the Bit 6 of the LPA field Identify Controller Data Structure to determine of the DUT supports Data Area 4 of Host and Controller log, if bit 6 is set to '1', skip test

© 2026 UNH–IOL

3. Configure the NVMe Host to issue a Telemetry Log Page Command for Controller-Initiated Data (LID=08h).

Observable Results:

1. Verify that the data beyond the last block in the Telemetry Controller-Initiated Data Area 3 Last Block is undefined

Case 41: Telemetry Controller-initiated, LPA bit 6=1 (OF-FYI, M)

Test Procedure:

1. Check the Bit 3 of the LPA field Identify Controller Data Structure to determine of the DUT supports Telemetry Host-Initiated and Telemetry Controller–Initiated log pages. If Bit 3 is set to 0 this test is not applicable
2. Check the Bit 6 of the LPA field Identify Controller Data Structure to determine of the DUT supports Data Area 4 of Host and Controller log, if bit 6 is set to '0', skip test
3. Configure the NVMe Host to issue a Telemetry Log Page Command for Controller-Initiated Data (LID=08h).

Observable Results:

1. Verify that the data beyond the last block in the Telemetry Controller-Initiated Data Area 4 Last Block is undefined

Case 42: Telemetry Controller-initiated, LPA bit 6=1, ETDAS=0 (OF-FYI, M)

Test Procedure:

1. Check the Bit 3 of the LPA field Identify Controller Data Structure to determine of the DUT supports Telemetry Host-Initiated and Telemetry Controller–Initiated log pages. If Bit 3 is set to 0 this test is not applicable
2. Check the Bit 6 of the LPA field Identify Controller Data Structure to determine of the DUT supports Data Area 4 of Host and Controller log, if bit 6 is set to '0', skip test
3. Check the Extended Telemetry Data Area 4 Supported (ETDAS) field in the Host Behavior Support feature, if set to '1', skip this test
4. Configure the NVMe Host to issue a Telemetry Log Page Command for Controller-Initiated Data (LID=08h).

Observable Results:

1. Verify that the data beyond the last block in the Telemetry Controller-Initiated Data Area 3 Last Block is undefined

Case 43: Telemetry Host-initiated, LPA bit 6=1, ETDAS=0 (OF-FYI, M)

Test Procedure:

1. Check the Bit 3 of the LPA field Identify Controller Data Structure to determine of the DUT supports Telemetry Host-Initiated and Telemetry Controller–Initiated log pages. If Bit 3 is set to 0 this test is not applicable
2. Check the Bit 6 of the LPA field Identify Controller Data Structure to determine of the DUT supports Data Area 4 of Host and Controller log, if bit 6 is set to '0', skip test
3. Check the Extended Telemetry Data Area 4 Supported (ETDAS) field in the Host Behavior Support feature, if set to '1', skip this test
4. Configure the NVMe Host to issue a Telemetry Log Page Command for Host Initiated Data (LID=07h).

Observable Results:

1. Verify that the data beyond the last block in the Telemetry Host-Initiated Data Area 3 Last Block is undefined

Case 44: Telemetry Controller-initiated, LPA bit 6=1 (OF-FYI, M)

Test Procedure:

1. Check the Bit 3 of the LPA field Identify Controller Data Structure to determine of the DUT supports Telemetry Host-Initiated and Telemetry Controller–Initiated log pages. If Bit 3 is set to 0 this test is not applicable

© 2026 UNH–IOL

2. Check the Bit 6 of the LPA field Identify Controller Data Structure to determine of the DUT supports Data Area 4 of Host and Controller log, if bit 6 is set to '0', skip test
3. Configure the NVMe Host to issue a Telemetry Log Page Command for Controller-Initiated Data (LID=08h).

Observable Results:

1. Verify that the value of the last block in the Telemetry Controller-Initiated Data Area 4 Last Block field is greater than or equal to the Telemetry Controller-Initiated Data Area 3 Last Block

Case 45: Telemetry Controller-Initiated, LPA bit 6=1, ETDAS=1 (OF-FYI, FYI)

Test Procedure:

1. Check the Bit 3 of the LPA field Identify Controller Data Structure to determine of the DUT supports Telemetry Host-Initiated and Telemetry Controller–Initiated log pages. If Bit 3 is set to 0 this test is not applicable
2. Check the Bit 6 of the LPA field Identify Controller Data Structure to determine of the DUT supports Data Area 4 of Host and Controller log, if bit 6 is set to '0', skip test
3. Check the Extended Telemetry Data Area 4 Supported (ETDAS) field in the Host Behavior Support feature, if set to '0', skip this test
4. Configure the NVMe Host to issue a Telemetry Log Page Command for Controller-Initiated Data (LID=08h).

Observable Results:

1. Verify that if Data Area 4 is created, then Data Area 3 is non-zero length and populated

Case 46: Media Unit Status Invalid Field in Command (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform an Identify Namespace Data Structure Command (CNS=18h) in order to get the Domain List associated with this controller.
3. Perform Get Log Page for the Media Unit Status Log Page (10h) with an invalid Domain Identifier not reported in the Domain List

Observable Results:

1. Verify that the controller shall abort the Get Log Page for the Media Unit Status Log Page command with 'Invalid Field in Command'

Case 47: Supported Capacity Configuration List, Invalid Field in Command (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform an Identify Namespace Data Structure Command (CNS=18h) in order to get the Domain List associated with this controller.
3. Perform Get Log Page for the Supported Capacity Configuration List Log Page (11h) with an invalid Domain Identifier not reported in the Domain List

Observable Results:

1. Verify that the controller shall abort the Get Log Page for the Supported Capacity Configuration List Log Page command with 'Invalid Field in Command'

Case 48: Predictable Latency Event Aggregate, Invalid Field in Command (OF-FYI, FYI)

Test Procedure:

© 2026 UNH–IOL

1. Check Bit 5 (Predictable Latency Mode) of the CTRATT field of the Identify Controller Data Structure. If Bit 5 is set to 0, then this test is not applicable.
2. Perform Get Log Page for the Predictable Latency Event Aggregate Log Page (0Bh)

Observable Results:

1. Verify that the Log page returned does not contain an entry (i.e., an NVM Set Identifier) that is cleared to 0h, if there are entries in the log.

Case 49: Media Unit Status, Selected Configuration=0 (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform an Identify Namespace Data Structure Command (CNS=18h) in order to get the Domain List associated with this controller.
3. Perform Get Log Page for the Media Unit Status Log Page (10h) with a valid Domain Identifier reported in the Domain List

Observable Results:

1. Verify that if the Selected Configuration field is cleared to ‘0’, that the ENDGID, NVMSETID, Capacity Adjustment Factor and MUC S fields are all cleared to ‘0’.

Case 50: Media Unit Status, Channel Ids ordered (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform an Identify Namespace Data Structure Command (CNS=18h) in order to get the Domain List associated with this controller.
3. Perform Get Log Page for the Media Unit Status Log Page (10h) with a valid Domain Identifier reported in the Domain List

Observable Results:

1. Verify that the Channel Identifiers in the Media Unit Status Descriptor of the Media Unit Status Log Page are in ascending order by value and each Channel Identifier only appears once.

Case 51: Media Unit Status, Domain Identifier=0 (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Check Bit 10 (Multi-Domain SubSystem) of the CTRATT field of the Identify Controller Data Structure.
3. Perform an Identify Namespace Data Structure Command (CNS=18h) in order to get the Domain List associated with this controller.
4. Perform Get Log Page for the Media Unit Status Log Page (10h) with a valid Domain Identifier reported in the Domain List

Observable Results:

1. Verify that if Multi-Domain Subsystem is not supported, then the Domain Identifier Field is cleared to ‘0’.

Case 52: Media Unit Status, ENDGID <= ENDGIDMAX (OF-FYI, FYI)

Test Procedure:

© 2026 UNH–IOL

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform an Identify Controller Data Structure Command (CNS=01h) and retrieve the Endurance Group Identifier Maximum (ENDGIDMAX)
3. Perform an Identify Namespace Data Structure Command (CNS=18h) in order to get the Domain List associated with this controller.
4. Perform Get Log Page for the Media Unit Status Log Page (10h) with a valid Domain Identifier reported in the Domain List and retrieve the Endurance Group Identifier (ENDGID).

Observable Results:

1. Verify that the ENDGID value is less than or equal to the ENDGIDMAX value

Case 53: Media Unit Status, NVMSETID <= NSETIDMAX (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform an Identify Controller Data Structure Command (CNS=01h) and retrieve the NVM Set Identifier Maximum (NSETIDMAX)
3. Perform an Identify Namespace Data Structure Command (CNS=18h) in order to get the Domain List associated with this controller.
4. Perform Get Log Page for the Media Unit Status Log Page (10h) with a valid Domain Identifier reported in the Domain List and retrieve the NVM Set Identifier (NVMSETID).

Observable Results:

1. Verify that the Identify Controller Data Structure command (CNS=01h) completed successfully
2. Verify that the NVMSETID value is less than or equal to the NSETIDMAX value

Case 54: Media Unit Status, same Endurance Group, same Capacity Adjustment Factor (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform an Identify Namespace Data Structure Command (CNS=18h) in order to get the Domain List associated with this controller.
3. Perform Get Log Page for the Media Unit Status Log Page (10h) with a valid Domain Identifier reported in the Domain List and step through each Media Unit Status Descriptor.

Observable Results:

1. Verify that for each Media Unit Status Descriptor that indicate the same Endurance Group Identifier, the Capacity Adjustment Factor fields have the same values

Case 55: Media Unit Status, Percentage Used (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform an Identify Namespace Data Structure Command (CNS=18h) in order to get the Domain List associated with this controller.
3. Perform Get Log Page for the Media Unit Status Log Page (10h) with a valid Domain Identifier reported in the Domain List and step through each Media Unit Status Descriptor.

Observable Results:

© 2026 UNH–IOL

1. Verify that for each Media Unit Status Descriptor, if the Percentage Used field is greater than a value of 254, it is represented as 255

Case 56: Media Unit Status, Channel Identifiers Offset (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform an Identify Namespace Data Structure Command (CNS=18h) in order to get the Domain List associated with this controller.
3. Perform Get Log Page for the Media Unit Status Log Page (10h) with a valid Domain Identifier reported in the Domain List and step through each Media Unit Status Descriptor.

Observable Results:

1. Verify that for each Media Unit Status Descriptor, the Channel Identifiers Offset is a non-zero value and a multiple of 16

Case 57: Support Capacity Configuration Lists, Capacity Configuration Identifier (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform Get Log Page for the Supported Capacity Configuration List Log Page (11h).

Observable Results:

1. Verify that the Capacity Configuration Identifier for each Capacity Configuration Descriptor in the Support Capacity Configuration List only appears once

Case 58: Support Capacity Configuration Lists, Endurance Group Configuration Identifier (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform Get Log Page for the Supported Capacity Configuration List Log Page (11h).

Observable Results:

1. Verify that the Endurance Group Configuration Identifier for each Endurance Group Configuration Descriptor, in each Capacity Configuration Descriptor in the Support Capacity Configuration List, only appears once

Case 59: Support Capacity Configuration Lists, Multiple Domains (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Check Bit 10 (Multi-Domain SubSystem) of the CTRATT field of the Identify Controller Data Structure, if MDS is set to 1 this test is not applicable
3. Perform Get Log Page for the Supported Capacity Configuration List Log Page (11h).

Observable Results:

1. Verify that the Domain Identifier field is cleared to ‘0’ for each Capacity Configuration Descriptors

© 2026 UNH–IOL

Case 60: Support Capacity Configuration Lists, NVM Set Identifier (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform Get Log Page for the Supported Capacity Configuration List Log Page (11h).

Observable Results:

1. Verify that each NVM Set Identifier only appear once in the Endurance Group Configuration Descriptor

Case 61: Support Capacity Configuration Lists, Channel Identifier (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform Get Log Page for the Supported Capacity Configuration List Log Page (11h).

Observable Results:

1. Verify that each Channel Identifier only appear once in the Channel Configuration Descriptor

Case 62: Support Capacity Configuration Lists, Endurance Group ENDGID (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform an Identify Controller Data Structure Command (CNS=01h) and retrieve the Endurance Group Identifier Maximum (ENDGIDMAX)
3. Perform Get Log Page for the Supported Capacity Configuration List Log Page (11h) and retrieve the Endurance Group Identifier (ENDGID) field

Observable Results:

1. Verify that the Endurance Group Identifier (ENDGID) is greater than 1h and less than or equal to the ENDGIDMAX value

Case 63: Support Capacity Configuration Lists, NVM Set Identifiers (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform an Identify Controller Data Structure Command (CNS=01h) and retrieve the NVM Set Identifier Maximum (NSETIDMAX)
3. Perform Get Log Page for the Supported Capacity Configuration List Log Page (11h) and retrieve the NVM Set ‘n’ Identifier field

Observable Results:

1. Verify that the NVM Set ‘n’ Identifier field is greater or equal to 0h and less than or equal to the NSETIDMAX value

Case 64: Support Capacity Configuration Lists, NVM Set EGSET (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.

© 2026 UNH–IOL

2. Perform an Identify Controller Data Structure Command (CNS=01h) and retrieve the NVM Set Identifier Maximum (NSETIDMAX)
3. Perform Get Log Page for the Supported Capacity Configuration List Log Page (11h) and retrieve the NVM Set EGSET-1 Identifier field

Observable Results:

1. Verify that the NVM Set EGSET-1 Identifier field is greater or equal to 0h and less than or equal to the NSETIDMAX value

Case 65: Channel Configuration Descriptor, NVM Set EGSET (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform Get Log Page for the Supported Capacity Configuration List Log Page (11h)

Observable Results:

1. Verify that the Channel Configuration Descriptors are listed in ascending order by Media Unit Identifier and each Media Unit Identifier appears only once.

Case 66: Channel Configuration Descriptor, MUDL=0 (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform Get Log Page for the Supported Capacity Configuration List Log Page (11h)

Observable Results:

1. Verify that the Media Unit Descriptor Length (MUDL) field in Channel Configuration Descriptors is cleared to ‘0’.

Case 67: NVM Capacity Model with Endurance Group support and no NVM Set support (OF-FYI, FYI)

Test Procedure:

1. Check Bit 2 and 4 (NVM Set and Endurance Group) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is set to 0 and Bit 4 are not set to 1, then continue with this test, if either is set differently this test is not applicable.
2. Perform a Get Log on the Media Unit Status (10h)
3. Perform a get on the I/O Command Set Independent Identify Namespace data structure (CNS 08h)

Observable Results:

1. Verify that NVMSETID fields are cleared to zero in all data structures.

Case 68: NVM Capacity Model, Endurance Group not supported (OF-FYI, FYI)

Test Procedure:

1. Check Bit 4 (Endurance Group) of the CTRATT field of the Identify Controller Data Structure. If Bit 4 is set to 0, then continue with this test.
2. Perform a Get Log Page on the Media Unit Status (LID 10h)
3. Perform a Get Log Page on Supported Capacity Configuration List (LID 11h)
4. Perform a get on the Identify Controller Data Structure, I/O Command Set Independent CNS 01h)
5. Perform a get on the I/O Command Set Independent Identify Namespace data structure (CNS 08h)

© 2026 UNH–IOL

Observable Results:

1. Verify that ENDGID fields are cleared to zero in all data structures.

Case 69: Get Media Unit Status log page, ANA enabled (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform an Identify Controller Data Structure command. Record Bit 3 of the CMIC field. If Bit 3 is set to 0, this test is not applicable.
3. Perform a Get Log Page for Log Page ID 0Ch with the RGO bit set to 0.
4. Perform a Get Log Page on the Media Unit Status (LID 10h)

Observable Results:

1. Verify that the Get Log Page command for Log Page ID 0Ch completed successfully.
2. Verify that the ANA State of the ANA Group Descriptor returned in the Get Log Page 0Ch
3. Verify that the returned status for the Get Log Page on the Media Unit Status (10h) is either Asymmetric Access Inaccessible, Asymmetric Access Persistent loss or Asymmetric Access Transition respectively for the ANA States: ANA Inaccessible State, ANA Persistent Loss State or ANA Change state

Case 70: Supported Capacity Configuration List log page, ANA enabled (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform an Identify Controller Data Structure command. Record Bit 3 of the CMIC field. If Bit 3 is set to 0, this test is not applicable.
3. Perform a Get Log Page for Log Page ID 0Ch with the RGO bit set to 0.
4. Perform a Get Log Page on the Supported Capacity Configuration List (LID 11h)

Observable Results:

1. Verify that the Get Log Page command for Log Page ID 0Ch completed successfully.
2. Verify that the ANA State of the ANA Group Descriptor returned in the Get Log Page 0Ch
3. Verify that the returned status for the Get Log Page on the Supported Capacity Configuration List (11h) is either Asymmetric Access Inaccessible, Asymmetric Access Persistent loss or Asymmetric Access Transition respectively for the ANA States: ANA Inaccessible State, ANA Persistent Loss State or ANA Change state

Case 71: Telemetry Host-Initiated, Establish Context and Read 512 header support (OF-FYI, M)

Test Procedure:

1. Check the Version field in the Identify Controller Data Structure, if the device claims to support NVMe version 1.4 or less, this test is not applicable
2. Check Bit 4 of the Log Page Attributes field in the Identify Controller Data Structure. If Bit 4 is set to 0, then this test is not applicable
3. Configure the NVMe Host to issue a Get Log Page Command for Log Identifier 00h “Supported Log Pages”.
4. Read the LID specific Parameter field from the Persisten Event Log Page 0Dh

Observable Results:

1. Verify that the get log page command completes successfully
2. Verify that the Establish Context and Read 512 Bytes of Header Supported field is set to 1.

© 2026 UNH–IOL

Case 72: Telemetry Host-Initiated, MCDS and Data Area size (OF-FYI, FYI)

Test Procedure:

1. Check the Bit 3 of the LPA field Identify Controller Data Structure to determine of the DUT supports Telemetry Host-Initiated and Telemetry Controller–Initiated log pages. If Bit 3 is set to 0 this test is not applicable.
2. Check bit 0 of the LID specific field for the Telemetry Host-Initiated Log Page, if set to 0, then the Maximum Created Data Area Support (MCDAS) is not supported and this test is not applicable.
3. Configure the NVMe Host to issue three separate Get Log Page command specifying Telemetry Host-Initiated Log Page to the NVMe Controller with a Log Page Offset Lower value that is a multiple of 512 bytes, and the Create Telemetry Host-Initiated Data bit is set to 1 and the Maximum Created Data Area (MCDA) Log Specific Parameter to the following values:
    a. 001b (Data Area 1)
    b. 010b (Data Areas 1 and 2)
    c. 011b (Data Areas 1, 2 and 3)

Observable Results:

1. Verify that for each of the Get Log Page commands sent, the controller does not return more Data Areas than specified in the Log Specific Field.

Case 73: Telemetry Host-Initiated, Generation Number (OF-FYI, FYI)

Test Procedure:

1. Check Bit 4 of the Log Page Attributes field in the Identify Controller Data Structure. If Bit 4 is set to 0h, then this test is not applicable.
2. Send a Get Log Page command for the Log Identifier 00h and read back the LID Specific parameter for LID 0Dh (Persistent Event Log). If bit 0 (Establish Context and Read 512 Bytes of Header Supported) is set to 0h, this test is not applicable.
3. Configure the NVMe Host to issue a Get Log Page command for the Log Identifier 0Dh “Persistent Event”.

Observable Results:

1. Verify that all Get Log Page commands completed with status ‘Success’.
2. Verify the Generation Number (byte 372) in the Persistent Event Log Page is a value less than or equal to FFFFh.

Possible Problems: Log Page 00h is mandatory for 2.0 drives, but reserved in 1.4 or less devices. It is possible that devices may support these fields but are not advertising 2.0 as their version. As a result, an Invalid Field status may be returned for step 2, in which case this test is not applicable.

Case 74: Management Address List (OF-FYI, FYI)

Test Procedure:

1. Send a Get Log Page command for the Log Identifier 18h, Management Address List, and read back the Management Address List data structure containing Management Address Descriptors.
2. If the Get Log Page is completed with a status success, then read and display each of the Management Address Descriptors in the Management Address List log.

Observable Results:

1. Verify that the Get Log Page command completed with status ‘Success’. If the Get Log Page command completes with status of ‘Invalid Log Page’ (09h) then this log page is not supported by the device. NVMe Controllers compliant with version 2.0 or earlier may return ‘Invalid Field In Command’ (02h).
2. If the Get Log Page Command completes with status ‘Success’. Verify that the last Management Address Type is either the eighth entry or the MAT field is set to FFh. Display all non-reserved Management Address Descriptors fields in the Management Address List.

© 2026 UNH–IOL

Case 75: Get Error Information enhancement (OF-FYI, FYI)

Test Procedure:

1. If the DUT does not support NVMe v2.0 or higher, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command to LID 01h, Error Information log, and record the number of Error log entries. If the Log Page Version value is not greater than or equal to 1h this test is not applicable.
3. Configure the NVMe Host to issue an Identify Command with a Reserved CNS value.
4. Configure the NVMe Host to issue a Get Log Page command to LID 01h, Error Information, and record the number of Error log entries.

Observable Results:

1. Verify that the Command Set Indicator (CSI) is set to a value of 0h and verify the OpCode is for the command that the error is associated with in the Get Log Page.

Case 76: Format NVM Completion Event, Persistent Event Log (OF-FYI, FYI)

Test Procedure:

1. Check the OACS field of the Identify Controller Data structure to determine if the DUT supports the Format NVM command. If the command is not supported this test is not applicable.
2. Check Bit 4 of the Log Page Attributes field in the Identify Controller Data Structure. If Bit 4 is set to 0h, then this test is not applicable.
3. Record the values in the DPS field of the Identify Namespace Data structure and all other pertinent attributes of the namespace, in order to put the namespace back to its original state.
4. Configure the NVMe Host to issue a Format NVM Command with an LBAF Supported by the DUT, SES=000b and FFFFFFFFh for the namespace ID if FNA bit 3 is cleared to ‘0h’. If FNA bit 3 is set to ;1; issue a Format NVM Command to the current active namespace. Note the status returned from the Format Command.
5. Configure the NVMe Host to issue a Get Log Page Command for Log Identifier 0Dh “Persistent Event Log”.
6. Verify that the Persistent Event Format structure’s Event Type Revision field is set to 02h and the Event Type field is set to 08h. If not set to these two values respectively this test is not applicable.
7. Perform a Format NVM operation to restore the DPS settings for the Namespace that were recorded in Step 3.

Observable Results:

1. Verify that the Format and Get Log Page commands complete with status of ‘Success’ (00h).
2. Verify that the Format NVM Completion Event Data Format structure’s Status field in Bits (15:1) has the same status as that reported from the Format command.

Case 77: Supported Capacity Configuration List (11h) appear ascending order and once (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform Get Log Page for the Supported Capacity Configuration List Log Page (11h)

Observable Results:

1. Verify that the Capacity Configuration Descriptors, the Endurance Group Configuration Descriptors, NVM Set Identifiers and Channel Configuration Descriptors are listed in ascending order and appear only once.

Case 78: NSID Specified on Controller or NVM Subsystem Scope Log Pages (OF-FYI, M)

Test Procedure:

© 2026 UNH–IOL

1. Configure the NVMe host to issue an Identify command specifying CNS 1h (Identify Controller Data Structure, I/O Command Set Independent) to read the value of the Log Page Attributes (LPA) Field. If LPA bit 5 is set to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page Command for Log Identifier 00h "Supported Log Pages" to determine which log pages are supported by the device under test.
3. For all of the supported log pages that are Controller or NVM Scope only, send a Get Log Page command to that feature, specifying an NSID of 0h and then FFFFFFFFh.
    a. The following is a list of Log Pages that could be used in this test if supported:
        i. Supported Log Pages (00h)
        ii. Error Information (01h)
        iii. Firmware Slot Information (03h)
        iv. Changed Namespace List (04h)
        v. Commands Supported and Effects (05h)
        vi. Device Self-test (06h)
        vii. Telemetry Host-Initiated (07h)
        viii. Telemetry Controller-Initiated (08h)
        ix. Endurance Group Information (09h)
        x. Predictable Latency Per NVM Set (0Ah)
        xi. Predictable Latency Event Aggregate (0Bh)
        xii. Asymmetric Namespace Access (0Ch)
        xiii. Persistent Event Log (0Dh)
        xiv. LBA Status Information (0Eh)
        xv. Endurance Group Event Aggregate (0Fh)
        xvi. Media Unit Status (10h)
        xvii. Supported Capacity Configuration List (11h)
        xviii. Feature Identifiers Supported and Effects (12h)
        xix. NVMe-MI Commands Supported and Effects (13h)
        xx. NVMe-MI Commands Supported and Effects (14h)
        xxi. Boot Partition (15h)
        xxii. Management Address List (15h)
        xxiii. Discovery (70h)
        xxiv. Host Discovery (71h)
        xxv. AVE Discovery (72h)
        xxvi. Pull Model DDC Request (73h)
        xxvii. Reservation Notification (80h)
        xxviii. Sanitize Status (81h)
4. Send a second Get Log Page command specifying the same LIDs used in the previous step, with an NSID specifying a valid and attached namespace.

Observable Results:

1. Verify that all the Get Log Page commands in step 3 complete successfully.
2. Verify that all the Get Log Page commands in step 4 complete with the status of 'Invalid Field in Command' (02h).

Possible Problems: This test is part of the should to shall conversion TP4136. As a result, the "shall" that constitutes this test does not exist until after a later version of the NVMe Base Specification 2.0.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.3.78 and expect the same observable results.

1. **LID 1Ch Changed Allocated Namespace List**
2. **LID 25h Power Management**

© 2026 UNH–IOL

Case 79: NSID Controller or NVM Subsystem SMART/Health Log Page (OF-FYI, M)

Test Procedure:

1. Configure the NVMe host to issue an Identify command specifying CNS 1h (Identify Controller Data Structure, I/O Command Set Independent) to read the value of the Log Page Attributes (LPA) Field. If LPA bit 0 is set to 1, this test is not applicable.
2. Configure the NVMe host to send a Get Log Page command with LID 02h, specifying an NSID of 0h.
3. Configure the NVMe host to send a Get Log Page command with LID 02h, with an NSID specifying a valid and attached namespace.

Observable Results:

1. Verify that the first get log page command completes successfully.
2. Verify that that the Get Log Page command in step 3 completes with the status of 'Invalid Field in Command' (02h).

Case 80: Command Supported Effects Log page – Cancel Command Not Supported (M)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID 05h). If bit 0 of the Commands Supported and Effects Data Structure for the Cancel command (Opcode 18h) is set to 1 then this test is not applicable.
2. Configure the NVMe Host to issue a Write command with a known data pattern to write to one LBA.
3. Configure the NVMe Host to issue a Cancel command specifying:
    a. Command Identifier (CID) to be the CID of the Write command;
    b. Submission Queue Identifier (SQID) to be the I/O queue the Write command was issued.
    c. The Action Code to be 'Single Command Cancel' (00b); and
    d. The NSID set to FFFFFFFFh;
4. Configure the NVMe Host to issue a Read command to the LBA which was written to in step 2.

Observable Results:

1. Verify the Cancel command returns with status 'Invalid Command Opcode' (01h).
2. Verify all other commands completed successfully.
3. Verify that the known data pattern was read correctly from the LBA exactly as it was written.

Case 81: Command Supported Effects Log Page – Cancel Command Indicates Support (M)

Test Procedure:

1. Configure the NVMe Host to issue a Write command with a known data pattern to write to one LBA.
2. Configure the NVMe Host to issue a Cancel command specifying:
    a. Command Identifier (CID) to be the CID of the Write command;
    b. Submission Queue Identifier (SQID) to be the I/O queue the Write command was issued;
    c. The Action Code to be 'Single Command Cancel' (00b); and
    d. The NSID set to FFFFFFFFh;
3. Configure the NVMe Host to issue a Read command to the LBA which was written to in step 2.
4. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID 05h). Record Command Supported (CSUPP), bit 0 of the Commands Supported and Effects Data Structure for the Cancel command (Opcode 18h).

Observable Results:

1. If the Cancel command completed successfully:
    a. Verify that all other commands complete successfully (excluding the Write command).
    b. Verify that the Write command is aborted or completes successfully
    c. Verify that the CSUPP bit is for the Cancel command is set to 1
2. If the Cancel command completed with status 'Invalid Command Opcode' (01h):

© 2026 UNH–IOL

a. Verify that the Get Log Page command either:
   i. Returned 'Invalid Field if the device is version pre-2.0';
   ii. Returned 'Invalid Log Identifier' for version 2.0 or later; or
   iii. Returned 'Success' and the CSUPP bit for the Cancel command is cleared to 0.
b. Verify all other commands completed successfully.

3. Any other result is a failure.

Case 82: Error Information Log Page NVM Command Set Identifier (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command specifying CNS value 01h to the controller to receive back an Identify Controller data structure.
2. Read the VER field, if the device under test reports version NVMe 1.4 or less, this test may not be applicable.
3. Configure the NVMe Host to issue a Get Log Page command to LID 01h
4. Read byte 63 of the Error Information Log Page that is returned from the previous step. If byte 63 is set to 0, then this test is not applicable.
5. Configure the NVMe Host to issue an Identify Command with a reserved CNS value, specifying an NVM namespace.
6. Configure the NVMe Host to issue a second Get Log Page command to LID 01h

Observable Results:

1. Verify that all commands except for the Identify Command in step 4 complete successfully.
2. Verify that byte 30 of log the page returned in step 5 is set to 0h, NVM Command Set.
3. Verify that byte 31 is set to 06h.

Possible Problems: If the Device Under Test supports NVMe 1.4 or less, then this test may not be applicable. If the manufacturer decides to set byte 63 of the Error Information Log Page to 1h regardless of the NVMe version, then this test is applicable.

Case 83: Get Log Page Commands Ignore CSI field or if CC.CSS is Set to 110b (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to perform a read on the CC.CSS register. If CC.CSS is not set to 0, then clear the CC.EN register field and then clear the CC.CSS register field to 0. If the CC.CSS registry field is not able to be set to 0 then skip this test. Re-enable the controller by setting the CC.EN bit to 1.
2. Configure the NVMe Host to perform Get Log Page commands on Log Page Identifiers that are Command Set Indicator (CSI) specific with unsupported Command Sets, with the CS.CSS register field set to 110b.

Observable Results:

1. Verify that the Get Log Page succeeds, thus ignoring the CSI setting in the Get Log Page command.

Case 84: Command Supported Effects Log page – Command Submission and Execution Relaxations (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID=05h). If this LID is not supported in NVMe Base version 1.4 or earlier then this test is not applicable. Configure the NVMe Host to perform Get Log Page commands on Log Page Identifiers that are Command Set Indicator (CSI) specific with unsupported Command Sets, with the CS.CSS register field set to 110b.
2. For each Commands Supported and Effects Data Structure in the log page, record the CSER and CSE values.

Observable Results:

1. Verify that the Get Log Page command completes successfully.

© 2026 UNH–IOL

2. Verify that if the Command Submission and Execution Relaxations (CSER) field is set to 01b for any given command, then the Command Submission and Execution (CSE) field for that command shall be set to 001b.

Case 85: Command Supported Effects Log page – Command Not Supported (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID=05h). If this LID is not supported in NVMe Base version 1.4 or earlier, then this test is not applicable.
2. For each Commands Supported and Effects Data Structure in the log page, record the CSUPP value.

Observable Results:

1. Verify that the Get Log Page command completes successfully.
2. Verify that if the Command Supported (CSUPP) bit is cleared to 0 for any given command, then the entire Commands Supported and Effects Data Structure for that command shall be cleared to 0

Case 86: Get Log Page Command 0Dh, Persistent Event Log PIT Field != 0 (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If the DUT claims support for a version less than 2.0, then this test is not applicable. If Bit 4 of the Log Page Attributes (LPA) field is cleared to 0, then this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command for the Persistent Event Log Page (LID 0Dh) with LSP set to 10b to release any previous context.
3. Configure the NVMe Host to issue a Get Log Page command for the Persistent Event Log Page (LID 0Dh) with LSP set to 11b to Establish Context and Read 512 Bytes of Header.
4. If the Total Number of Events (TNEV) value of the Persistent Event Log Header is cleared to 0, configure the NVMe Host to create a new Persistent Event.
5. Configure the NVMe Host to issue another Get Log Page command with LSP cleared to 0 to read the Port Identifier Type (PIT) field in the first Event Header Additional Information field (byte 515) of the log page.

Observable Results:

1. Verify that the Identify command completes successfully.
2. Verify that the Get Log Page commands complete successfully.
3. Verify that the Port Identifier Type (PIT) field is not cleared to 0.

Case 87: Persistent Event Log Page (LID 0Dh) - Sanitize Media Verification Event (Event Type 0Eh) (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bit 3 of SANICAP, VERS, is cleared to 00b, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command for the Persistent Event Log Page (LID 0Dh) with LSP set to 10b to release any previous context.
3. Configure the NVMe Host to issue a Get Log Page command for the Persistent Event Log Page (LID 0Dh) with LSP set to 11b to establish context and read 512 bytes of header.
4. Verify that bit 14 is set to 1b in the Supported Events Bitmap in the Persistent Event Log Page Header returned from the Get Log Page command in the previous step. If bit 14 is cleared to 00b, this test is not applicable.
5. Configure the NVMe Host to issue a Sanitize command with Enter Media Verification State (EMVS) set to 1b and the Sanitize Action (SANACT) set to either Block Erase (010b) or Crypto Erase (100b).
6. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.

© 2026 UNH–IOL

7. Configure the NVMe Host to issue a Get Log Page command for the Persistent Event Log Page (LID 0Dh) with LSP cleared to 0h to read log data and record all reported Persistent Events.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that the Persistent Event Log Page from step 6 contains a Sanitize Media Verification Event (Event Type 0Eh), and the Persistent Event Log Event Format Header has the following attributes
    a. Event Type field to 0Eh;
    b. Event Type Revision field to 01h; and
    c. Event Length field to 0h

Case 88: Persistent Event Log Page (LID 0Dh) - Sanitize Completion Event (Event Type 0Ah) (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bit 3 of SANICAP, VERS, is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command for the Persistent Event Log Page (LID 0Dh) with ACT set to 10b to release any previous context.
3. Configure the NVMe Host to issue a Get Log Page command for the Persistent Event Log Page (LID 0Dh) with ACT set to 11b to establish context and read 512 bytes of header.
4. Verify that bit 10 (Sanitize Completion Event Support) is set to 1b in the Supported Events Bitmap (SEB) in the Persistent Event Log Header returned from the Get Log Page command in the previous step. If bit 10 is cleared to 0h, this test is not applicable.
5. Configure the NVMe Host to issue a Get Log Page for the Persistent Event Log Page (LID 0Dh) with ACT set to 10b (Release Context) to clear the context established in step 3.
6. Create a Sanitize Completion event by initiating a sanitize operation (i.e., the sanitization target transitions to the Restricted Processing state or the Unrestricted Processing state, as described in section 8.1.26.4):
    a. Configure the NVMe Host to issue a Get Log Page command for the Persistent Event Log Page (LID 0Dh) with ACT set to 11b to establish context and read 512 bytes of header.
    b. Configure the NVMe Host to issue a Sanitize Namespace command targeting a valid namespace with the SANACT set to 100b (Crypto Erase) and Allow Unrestricted Sanitize Exit (AUSE) set to 1b.
    c. Periodically issue a Get Log Page for the Sanitize Status Log Page (LID 81h) for the Namespace Identifier (NSID) specified in step 6.b until Sanitize Progress (SPROG) is set to FFFFh.
7. Configure the NVMe Host to issue a Get Log Page command for the Persistent Event Log Page (LID 0Dh) with the Log Specific Parameter (LSP) cleared to 000b to read log data.

Observable Results:

1. Verify that all commands completed successfully.
2. Verify that the Persistent Event Log Page from step 7 contains a Sanitize Completion Event (Event Type 0Ah), and the Persistent Event Log Event Format Header has the following attributes:
    a. Event Type field to 0Ah;
    b. Event Type Revision field to 02h; and
    c. NSID field (Bytes 11:08) in the event data corresponds to the NSID specified in step 6.b.

Case 89: Sanitize Status (Log Page Identifier 81h) Unallocated NSID (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. Check the Optional Admin Command Support (OACS) bit 3 to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to obtian the Allocated Namespace ID list (CNS 10h) with a Namespace Identifier (NSID) cleared to 0h to the controller in order to obtain the Allocated Namespace ID list.

© 2026 UNH–IOL

3. Configure the NVMe Host to issue a Namespace Management command with a valid NSID, a Select field value 0h (Create), and valid values for the attached data structure to the DUT to create a new namespace.
4. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h) and an NSID created in step 3, if available.

Observable Results:

1. Verify that all commands in steps 1, 2, and 3 completed successfully.
2. Verify that the Get Log Page command in step 4 aborts with the status code 'Invalid Namespace or Format' (0Bh).

Case 90: Sanitize Status Log Page Target-Specific Field Validations (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID 05h). If bit 0 of the Commands Supported and Effects data structure for the Sanitize Namespace command (opcode 8Ch) is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h), specifying an allocated NSID. Record the values of the following fields:
   - •Estimated Time for Overwrite (ETO)
   - •Estimated Time for Block Erase (ETBE)
   - •Estimated Time for Overwrite with No Media Modification (ETODMM)
   - •Estimated Time for Block Erase with No Media Modification (ETBENMM)
   - •Estimated Time for Crypto Erase with No Media Modification (ETCENMM)
   - •Maximum Namespace Sanitize Operations In Progress (MNSOIP)
3. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status log page (LID 81h) configured to target the global NVM subsystem (FFFFFFFFh). Record the value of the Sanitize Target Namespace Identifier (STNSID) field.

Observable Results:

1. Verify that each field recorded in the Get Log Page command in step 2 returned a value of FFFFFFFFh.
2. Verify that when targeting the global NVM subsystem, the STNSID field is cleared to 0h.

Case 91: Sanitize Namespace Status List Log Page Contents Verification (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID 05h). If bit 0 of the Commands Supported and Effects data structure for the Sanitize Namespace command (opcode 8Ch) is cleared to 0h then this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Namespace Status List (LID 7Fh):
   a. Once with Log Specific Parameter (LSP) set to 0b (In Progress); and
   b. Once with LSP set to 1b (Successfully Completed).
3. Configure the NVMe Host to issue a Sanitize Namespace command targeting an allocated namespace with Sanitize Action (SANACT) set to Crypto Erase (100b).
4. While the sanitize operation is actively running, configure the NVMe Host to issue a Get Log Page command for the Sanitize Namespace Status List (LID 7Fh) with LSP set to 0b (In Progress). Record the list.
5. Periodically issue a Get Log Page command for the Sanitize Status Log Page (LID 81h) specifying the NSID of Namespace A until the Sanitize Progress (SPROG) field is set to FFFFh.
6. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Namespace Status List (LID 7Fh) with LSP set to 1b (Successfully Completed). Record the list.
7. Configure the NVMe Host to issue a final Get Log Page command for the Sanitize Status Log Page (LID 81h) specifying the NSID of the namespace used in step 3.

Observable Results:

© 2026 UNH–IOL

1. Verify that the NSID intended for testing is not present in either list returned from the Get Log Page commands issued in step 2.
2. Verify that while the operation is in progress after step 4, the target namespace NSID is successfully reported in the log page list, and its associated status fields indicate that the Sanitize State is ‘Sanitizing’ (010b).
3. Verify that in Step 6, the target namespace was present in the Successfully Completed list returned by the controller.
4. Verify that after successful completion, the Sanitize Operation Status (SOS) associated with the target returns ‘Sanitized’ (001b).

© 2026 UNH–IOL

Test 1.4 – Create/Delete I/O Submission and Completion Queue Commands (M)

Purpose: To verify that an NVMe Controller can properly create and delete I/O Submission and Completion Queues.

References:

Old Ref: [1] NVMe Specification 5.3, 5.4, 5.5, 5.6, NVMe v1.3 ECN 001
NVM Express Base specification 2.0a : 5.4, 5.5, 5.6, 5.7

Resource Requirements:

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

Last Modification: May 4, 2020

Discussion: The Create I/O Completion Queue command is used to create I/O Completion Queues while the Delete I/O Completion Queue command is used to delete I/O Submission Queues. Likewise, the Create I/O Submission Queue command is used to create I/O Submission Queues while the Delete I/O Submission Queue command is used to delete I/O Submission Queues. Host software shall ensure that any associated I/O Submission Queue is deleted prior to deleting a Completion Queue.

The Create I/O Submission Queue and Create I/O Completion Queue commands use the PRP Entry 1, Command Dword 10, and Command Dword 11 fields. The Delete I/O Submission Queue and Delete I/O Completion Queue commands use the Command Dword 10. All other command specific fields are reserved.

A completion queue entry is posted to the Admin Completion Queue when the specified queue has been created or deleted.

Test Setup: See Appendix A.

Case 1: Basic Operation (M)

Test Procedure:

1. Configure the NVMe Host to issue a Create I/O Completion Queue command to the controller.
2. Configure the NVMe Host to issue a Create I/O Submission Queue command to the controller.
3. Configure the NVMe Host to issue 10 Read commands assigned to the queues created in steps 1 and 2 to the controller.
4. Configure the NVMe Host to issue a Delete I/O Submission Queue command specifying the Submission Queue ID of the queue created in step 2 to the controller.
5. Configure the NVMe Host to issue a Delete I/O Completion Queue command specifying the Completion Queue ID of the queue created in step 1 to the controller.

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 queues are properly created and deleted by reading the completion queue entries posted for each command issued by the NVMe Host (all statuses should be Success).
3. Verify that all received responses have all Reserved fields set to 0.

Case 2: Create I/O Completion Queue with QID=0h, exceeds Number of Queues reported, Identifier Already in Use (M)

Test Procedure:

1. Configure the NVMe Host to issue a Get Features command for the Feature Identifier 07h, Number of Queues, record the value reported.
2. Configure the NVMe Host to issue a Create I/O Completion Queue command to the controller, with QID=0h.

© 2026 UNH–IOL

3. Configure the NVMe Host to issue a Create I/O Completion Queue command to the controller with QID=greater than the value reported in the Number of Queues field in the Get Feature response. If the DUT supports the maximum possible number of Queues than this test case is not applicable.
4. Configure the NVMe Host to issue 2 Create I/O Completion Queue commands with identical QID. The first command is expected to complete successfully, the second is expected to return an error of Invalid Queue Identifier.

Observable Results:

1. Verify that after the completion of each command, the controller returns an error of Invalid Queue Identifier

Case 3: Delete I/O Completion Queue before deleting Corresponding Submission Queue (M)

Test Procedure:

1. Configure the NVMe Host to issue a Create I/O Completion Queue command to the controller.
2. Configure the NVMe Host to issue a Create I/O Submission Queue command to the controller.
3. Configure the NVMe Host to issue 10 Read commands assigned to the queues created in steps 1 and 2 to the controller.
4. Configure the NVMe Host to issue a Delete I/O Completion Queue command specifying the Completion Queue ID of the queue created in step 1 to the controller.
5. Configure the NVMe Host to issue a Delete I/O Submission Queue command specifying the Submission Queue ID of the queue created in step 2 to the controller.
6. Configure the NVMe Host to issue a Delete I/O Completion Queue command specifying the Completion Queue ID of the queue created in step 1 to the controller.

Observable Results:

1. Verify that the Delete I/O Completion Queue command in step 4, caused the controller to returns an error of Invalid Queue Deletion.

Case 4: Create I/O Completion Queue with Invalid Queue Size (M)

Test Procedure:

1. Configure the NVMe Host to read the Capabilities Register CAP.MQES field to obtain the Maximum Queue Size supported by the DUT. Record the value.
2. Configure the NVMe Host to issue a Create I/O Completion Queue command to the controller with a QSIZE of 0h.
3. Configure the NVMe Host to issue a Create I/O Completion Queue command to the controller with a QSIZE of greater than the Queue size (MQES) supported by the DUT.

Observable Results:

1. Verify that in each case the Create I/O Completion Queue, caused the controller to returns an error of Invalid Queue Size.

Case 5: Create I/O Submission Queue with Invalid Queue Size (M)

Test Procedure:

1. Configure the NVMe Host to read the Capabilities Register CAP.MQES field to obtain the Maximum Queue Size supported by the DUT. Record the value.
2. Configure the NVMe Host to issue a Create I/O Completion Queue command to the controller.
3. Configure the NVMe Host to issue a Create I/O Submission Queue command to the controller with a QSIZE of 0h.
4. Configure the NVMe Host to issue a Create I/O Submission Queue command to the controller with a QSIZE of greater than the Queue size (MQES) supported by the DUT.

Observable Results:

© 2026 UNH–IOL

1. Verify that in each case the Create I/O Submission Queue, caused the controller to returns an error of Invalid Queue Size.

Case 6: Create I/O Submission Queue Physically Contiguous (M)

Test Procedure:

1. Configure the NVMe Host to read the Capabilities Register CAP.CQR field. If CAP.CQR is set to zero, this test case is not applicable. If CAP.CQR is set to 1, proceed to the next step.
2. Configure the NVMe Host to issue a Create I/O Completion Queue command to the controller.
3. Configure the NVMe Host to issue a Create I/O Submission Queue command to the controller with the PC bit set to 0.

Observable Results:

1. Verify that the transmitted Create I/O Submission Queue, caused the controller to returns an error of Invalid Field in Command.

Case 7: Create I/O Submission Queue Invalid CQID of 0h (M)

Test Procedure:

1. Configure the NVMe Host to issue a Create I/O Completion Queue command to the controller.
2. Configure the NVMe Host to issue a Create I/O Submission Queue command to the controller with the CQID set to 0h.

Observable Results:

1. Verify that the transmitted Create I/O Submission Queue with CQID of 0h, caused the controller to returns an error of 1h - Invalid Queue Identifier.

Case 8: Create I/O Completion Queue Invalid Interrupt Vector (M)

Test Procedure:

1. Configure the NVMe Host to read the MSICAP.MC.MME field and the MSIXCAP.MXC.TS.
2. Configure the NVMe Host to issue a Create I/O Completion Queue command to the controller with an Interrupt Vector (IV) value greater than the number of messages supported by the DUT as indicated in either MSICAP.MC.MME or MSIXCAP.MXC.TS.

Observable Results:

1. Verify that the transmitted Create I/O Completion Queue, caused the controller to return an error of Invalid Interrupt Vector.

Case 9: Create I/O Submission Queue Invalid CQID Outside Supported Range (M)

Test Procedure:

1. Perform a Get Feature command for Feature ID 07h to obtain the NCQA value.
2. Configure the NVMe Host to issue a Create I/O Completion Queue command to the controller.
3. Configure the NVMe Host to issue a Create I/O Submission Queue command to the controller with the CQID set to value outside the range supported by the DUT, as indicated in NCQA.

Observable Results:

1. Verify that the transmitted Create I/O Submission Queue with CQID outside the supported range, caused the controller to return an error of 1h - Invalid Queue Identifier.

© 2026 UNH–IOL

Case 10: Create I/O Submission Queue Invalid CQID within supported range, but Queue not created (M)

Test Procedure:

1. Perform a Get Feature command for Feature ID 07h to obtain the NCQA value.
2. Configure the NVMe Host to issue a Create I/O Completion Queue command to the controller.
3. Configure the NVMe Host to issue a Create I/O Submission Queue command to the controller with the CQID set to a value which is within the supported range (as indicated by NCQA) but for a queue which has not been created.

Observable Results:

1. Verify that the transmitted Create I/O Submission Queue with CQID with an invalid value within the supported range as indicated in NCQA (3h is used in some test implementations), caused the controller to return an error of 0h - Completion Queue Invalid.

Case 11: Set Feature After Queues Created (M)

Test Procedure:

1. If the DUT does not support NVMe v1.4 or higher, this test is not applicable.
2. Configure the NVMe Host to issue a Create I/O Completion Queue command to the controller.
3. Configure the NVMe Host to issue a Create I/O Submission Queue command to the controller.
4. Configure the NVMe Host to issue 10 Read commands assigned to the queues created in steps 1 and 2 to the controller.
5. Configure the NVMe Host to issue a Delete I/O Submission Queue command specifying the Submission Queue ID of the queue created in step 2 to the controller.
6. Configure the NVMe Host to issue a Delete I/O Completion Queue command specifying the Completion Queue ID of the queue created in step 1 to the controller.
7. Configure the NVMe Host to issue a Set Feature Command for the Number of Queues Feature, FID=07h.

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 queues are properly created and deleted by reading the completion queue entries posted for each command issued by the NVMe Host (all statuses should be Success).
3. Verify that all received responses have all Reserved fields set to 0h.
4. Verify that the Set Feature Command failed with status ‘Command Sequence Error’ (0Ch).

Possible Problems: None known.

© 2026 UNH–IOL

Test 1.5 – Abort Command (M)

Purpose: To determine the conditions under which an NVMe system can successfully abort a given command.

References:

Old Ref: [1] NVMe Specification 5.1, NVMe v1.3 ECN 003
NVM Express Base specification 2.0a : 5.1

Resource Requirements:

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

Last Modification: December 1, 2019

Discussion: The Abort command is used to abort a specific command previously issued to the Admin Submission Queue or an I/O Submission Queue. Host software may have multiple Abort commands outstanding, subject to the constraints of the Abort Command Limit indicated in the Identify Controller data structure. An abort is a best effort command; the command to abort may have already completed, currently be in execution, or may be deeply queued. It is implementation specific if/when a controller chooses to complete the command when the command to abort is not found. Whether the command was successfully aborted or not is determined by examining bit 0 of command Dword 0 of the completion queue entry for the Abort command. If the command was successfully aborted, then bit 0 of Dword 0 is cleared to '0h'. If the command was not aborted, then bit 0 of Dword 0 is set to '1h'. The completion queue entry for the Abort command is only posted to the Admin Completion Queue after the completion queue entry for the command specified in the Abort command has been posted to its corresponding Admin or I/O Completion Queue.

The Abort command uses the Command Dword 10 field. All other command specific fields are reserved.

Since the Abort Command is a best effort command, this test is designed to determine under what conditions a command can successfully be aborted, and is therefore considered an informative test.

Test Setup: See Appendix A.

Case 1: Abort I/O Command (M)

Test Procedure:

1. Configure the NVMe Host to issue 10 Read commands to the controller.
2. Configure the NVMe Host to issue an Abort command to the controller specifying the CID of one of the issued Read commands.

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. Determine whether the requested command was successfully aborted by examining bit 0 of Dword 0 of the completion queue entry for the Abort command.
    a. If bit 0 of Dword 0 is cleared to '0h' in the completion queue entry for the Abort command, the command was aborted. If the READ command was successfully aborted, verify that the status of the aborted command is Command Abort Requested (07h), and that the Completion Queue Entry for the aborted READ command is posted to the I/O Completion Queue *before* the Completion Queue Entry of the Abort command is posted to the Admin Completion Queue.
    b. If bit 0 of Dword 0 is set to 1h, the command was not aborted. Verify that a completion Queue Entry for the Abort command is posted to the Admin Completion Queue.
3. Verify that all received responses have all Reserved fields set to 0h.

Case 2: Abort Admin Command (M)

Test Procedure:

© 2026 UNH–IOL

1. Configure the NVMe Host to issue 10 Get Log Page commands to the controller. Any supported Log Page ID can be used.
2. Configure the NVMe Host to issue an Abort command to the controller specifying the CID of one of the issued Get Log Page commands.

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. Determine whether the requested command was successfully aborted by examining bit 0 of Dword 0 of the completion queue entry for the Abort command.
    a. If bit 0 of Dword 0 is cleared to '0h' in the completion queue entry for the Abort command, the command was aborted. If the Get Log Page command was successfully aborted, verify that the status of the aborted command is Command Abort Requested (07h), and that the Completion Queue Entry for the aborted Get Log Page command is posted to the Admin Completion Queue before the Completion Queue Entry of the Abort command is posted to the Admin Completion Queue.
    b. If bit 0 of Dword 0 is set to 1h, the command was not aborted. Verify that a completion Queue Entry for the Abort command is posted to the Admin Completion Queue.
3. Verify that all received responses have all Reserved fields set to 0h.

Possible Problems: Cannot test ABORT Limit Exceeded execution time limit (there's a chance the 1st ABORT finishes before the last one).

Case 3: Abort I/O Command - Immediate Abort Not Performed Bit (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Write command with a known data pattern to the device under test.
2. Configure the NVMe Host to issue an Abort command to the controller specifying the CID of the previous Write command.
3. Repeat steps 1 to 2 nine additional times.

Observable Results:

1. For each Abort command performed, determine the Abort Action Type by examining the 'Immediate Abort Not Performed' field, bit 0 of Dword 0 of the completion queue entry. If bit 0 is cleared to 0h the Abort was successful and an Immediate Abort was performed.
    a. Verify that the corresponding Get Log Page command completed with status 'Command Abort Requested' (07h).
    b. Verify that the Completion Queue Entry for the aborted Write command is posted to the I/O Completion Queue before the Completion Queue Entry of the Abort command is posted to the A Completion Queue.

Case 4: Abort Admin Command – Immediate Abort Not Performed Bit (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a supported Get Log Page command.
2. Configure the NVMe Host to issue an Abort command to the controller specifying the CID of the previous Get Log Page command.
3. Repeat steps 1 to 2 nine additional times.

Observable Results:

1. For each Abort command performed, determine the Abort Action Type by examining the 'Immediate Abort Not Performed' field, bit 0 of Dword 0 of the completion queue entry. If bit 0 is cleared to 0h the Abort was successful and an Immediate Abort was performed.
    a. Verify that the corresponding Get Log Page command completed with status 'Command Abort Requested' (07h).

© 2026 UNH–IOL

b. Verify that the Completion Queue Entry for the aborted Write command is posted to the Admin Completion Queue before the Completion Queue Entry of the Abort command is posted to the Admin Completion Queue.

© 2026 UNH–IOL

Test 1.6 – Format NVM Command (OF, M)

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

References:

Old Ref: [1] NVMe Specification 5.23, NVMe v1.3 ECN 002, NVMe v1.3 ECN 005
NVM Express Base specification 2.0a : 5.14

Resource Requirements:

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

Last Modification: April 8, 2019

Discussion: The Format NVM command is used to low level format the NVM media. This is used when the host wants to change the LBA data size and/or metadata size. A low level format may destroy all data and metadata associated with all namespaces or only the specific namespace associated with the command. After the Format NVM command successfully completes, the controller shall not return any user data that was previously contained in an affected namespace.

The settings specified in the Format NVM command are reported as part of the Identify Namespace data structure. If the controller supports multiple namespaces, then the host may specify the value of FFFFFFFFh for the namespace in order to apply the format operation to all namespaces accessible by the controller regardless of the value of the Format NVM Attribute field in the Identify Controller data structure, if the FNA bit in the OACS field is set to zero.

The Format NVM command uses the Command Dword 10 field. All other command specific fields are reserved.

Test Setup: See Appendix A.

Case 1: Valid LBAF, SES=000b (OF, M)

Test Procedure:

1. Check the OACS field of the Identify Controller Data structure to determine if the DUT supports the Format NVM command. If the command is not supported this test is not applicable.
2. Record the values in the DPS field of the Identify Namespace Data structure.
3. Configure the NVMe Host to issue a Write command to the controller in order to write a known data pattern to an LBA range.
4. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was written in step3.
5. Configure the NVMe Host to issue a Format NVM Command with an LBAF Supported by the DUT, SES=000b and FFFFFFFFh for the namespace ID if FNA bit 3 is cleared to ‘0’. If FNA bit 3 is set to ;1; issue a Format NVM Command to the current active namespace.
6. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was written in step3.
7. Perform a Format NVM operation to restore the DPS settings for the Namespace that were recorded in Step 2.

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 same data pattern which was written to the NVM in step 3 is read from the NVM in step 4.
3. Verify that the data pattern written in step 4 is not returned by the controller in step 6.
4. Verify that all received responses have all Reserved fields set to 0.

Case 2: Valid LBAF, SES=001b (OF-FYI, M)

Test Procedure:

© 2026 UNH–IOL

1. Check the OACS field of the Identify Controller Data structure to determine if the DUT supports the Format NVM command. If the command is not supported this test is not applicable.
2. Record the values in the DPS field of the Identify Namespace Data structure
3. Configure the NVMe Host to issue a Write command to the controller in order to write a known data pattern to an LBA range.
4. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was written in step3.
5. Configure the NVMe Host to issue a Format NVM Command with an LBAF Supported by the DUT, SES=001b and FFFFFFFFh for the namespace ID if FNA bit 3 is cleared to '0'. If FNA bit 3 is 1 issue the Format NVM Command to the current active namespace
6. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was written in step3.
7. Perform a Format NVM operation to restore the DPS settings for the Namespace that were recorded in Step 2.

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 same data pattern which was written to the NVM in step 3 is read from the NVM in step 4.
3. Verify that the data pattern written in step 3 is not returned by the controller in step 6.
4. Verify that all received responses have all Reserved fields set to 0.

Case 3: Valid LBAF, SES=010b (OF-FYI, M)

Test Procedure:

1. Check the OACS field of the Identify Controller Data structure to determine if the DUT supports the Format NVM command. If the command is not supported this test is not applicable.
2. Record the values in the DPS field of the Identify Namespace Data structure. Note that in NVMe v1.4 the DPS value is reported in the DPS field of the Identify Namespace Data Structure
3. Check Bit 2 of the FNA field of the Identify Controller Data structure to determine if the DUT supports cryptographic erase. If the DUT does not support cryptographic erase this test case is not applicable.
4. Configure the NVMe Host to issue a Write command to the controller in order to write a known data pattern to an LBA range.
5. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was written in step4.
6. Configure the NVMe Host to issue a Format NVM Command with an LBAF Supported by the DUT, SES=010b and FFFFFFFFh for the namespace ID if FNA bit 3 is cleared to '0'. If FNA bit 3 is set to '1' issue a Format NVM Command to the current active namespace.
7. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was written in step4.
8. Perform a Format NVM operation to restore the DPS settings for the Namespace that were recorded in Step 2.

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 same data pattern which was written to the NVM in step 4 is read from the NVM in step 5.
3. Verify that the data pattern written in step 4 is not returned by the controller in step 7.
4. Verify that all received responses have all Reserved fields set to 0.

Case 4: Valid LBAF, SES=111b (reserved value) (OF-FYI, M)

Test Procedure:

1. Check the OACS field of the Identify Controller Data structure to determine if the DUT supports the Format NVM command. If the command is not supported this test is not applicable.

© 2026 UNH–IOL

2. Record the values in the DPS field of the Identify Namespace Data structure. Note that in NVMe v1.4 the DPS value is reported in the DPS field of the Identify Namespace Data Structure In earlier versions of the specification the DPS value is reported in the Formatted LBA Size field of the Identify Namespace data structure.
3. Configure the NVMe Host to issue a Write command to the controller in order to write a known data pattern to an LBA range.
4. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was written in step 4.
5. Configure the NVMe Host to issue a Format NVM Command with an LBAF Supported by the DUT, SES=111b and FFFFFFFFh for the namespace ID if FNA bit 3 is cleared to '0'. If FNA bit 3 is set to '1' issue a Format NVM Command to the current active namespace.
6. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was written in step 4.
7. Perform a Format NVM operation to restore the DPS settings for the Namespace that were recorded in Step 2.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.6.4 and expect the same observable results.

1. SES 011B

Sub-case Procedure:

5. Configure the NVMe Host to issue a Format NVM Command with an LBAF Supported by the DUT, SES=011b and FFFFFFFFh for the namespace ID if FNA bit 3 is cleared to '0h'. If FNA bit 3 is set to '1h' issue a Format NVM Command to the current active namespace.

2. SES 110B

Sub-case Procedure:

5. Configure the NVMe Host to issue a Format NVM Command with an LBAF Supported by the DUT, SES=110b and FFFFFFFFh for the namespace ID if FNA bit 3 is cleared to '0h'. If FNA bit 3 is set to '1h' issue a Format NVM Command to the current active namespace.

3. SES 101B

Sub-case Procedure:

5. Configure the NVMe Host to issue a Format NVM Command with an LBAF Supported by the DUT, SES=101b and FFFFFFFFh for the namespace ID if FNA bit 3 is cleared to '0h'. If FNA bit 3 is set to '1h' issue a Format NVM Command to the current active namespace.

4. SES 111B

Sub-case Procedure:

5. Configure the NVMe Host to issue a Format NVM Command with an LBAF Supported by the DUT, SES=111b and FFFFFFFFh for the namespace ID if FNA bit 3 is cleared to '0h'. If FNA bit 3 is set to '1h' issue a Format NVM Command to the current active namespace.

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 same data pattern which was written to the NVM in step 3 is read from the NVM in step 4.
3. Verify that the data pattern written in step 3 *is* returned by the controller in step 6.
4. Verify that the Format NVM command completes with error status code 'Invalid Field' (02h).
5. Verify that all received responses have all Reserved fields set to 0h.

Case 5: Unsupported LBAF, SES=000b (OF-FYI, M)

Test Procedure:

© 2026 UNH–IOL

1. Check the OACS field of the Identify Controller Data structure to determine if the DUT supports the Format NVM command. If the command is not supported this test is not applicable.
2. Check the NLBAF field in the Identify Namespace Data Structure to determine the number of LBA formats supported by the DUT. If the DUT is at NVMe version 1.4 or less and if NLBAF is set to 16, then LBAF cannot be set to an invalid value and this test is not applicable.
3. Record the values in the DPS field of the Identify Namespace Data structure.
4. Configure the NVMe Host to issue a Write command to the controller in order to write a known data pattern to an LBA range.
5. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was written in step 4.
6. Configure the NVMe Host to issue a Format NVM Command with an invalid LBAF.
7. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was written in step 4.
8. Perform a Format NVM operation to restore the DPS settings for the Namespace that were recorded in Step 2.

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 same data pattern which was written to the NVM in step 4 is read from the NVM in step 5.
3. Verify that the data pattern written in step 4 *is* returned by the controller in step 7.
4. Verify that the Format NVM command completes with error status code ‘Invalid Format’ (0Ah).
5. Verify that all received responses have all Reserved fields set to 0h.

Case 6: Unsupported LBAF, SES=111b (reserved value) (OF-FYI, M)

Test Procedure:

1. Check the OACS field of the Identify Controller Data structure to determine if the DUT supports the Format NVM command. If the command is not supported this test is not applicable.
2. Check the NLBAF field in the Identify Namespace Data Structure to determine the number of LBA formats supported by the DUT. If NLBAF is set to 16, then LBAF cannot be set to an invalid value and this test is not applicable.
3. Record the values in the DPS field of the Identify Namespace Data structure
4. Configure the NVMe Host to issue a Write command to the controller in order to write a known data pattern to an LBA range.
5. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was written in step 4.
6. Configure the NVMe Host to issue a Format NVM Command with an invalid LBAF and SES=111b.
7. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was written in step 4.
8. Perform a Format NVM operation to restore the DPS settings for the Namespace that were recorded in Step 2.

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 same data pattern which was written to the NVM in step 4 is read from the NVM in step 5.
3. Verify that the data pattern written in step 4 *is* returned by the controller in step 7.
4. Verify that the Format NVM command completes with error status code of ‘Invalid Field’ (02h) or ‘Invalid Format’ (0Ah).
5. Verify that all received responses have all Reserved fields set to 0h.

Case 7: Valid LBAF, SES=000b, PI is non-zero (M)

Test Procedure:

© 2026 UNH–IOL

1. Check the OACS field of the Identify Controller Data structure to determine if the DUT supports the Format NVM command. If the command is not supported this test is not applicable.
2. Check the DPC field of the Identify Namespace Data Structure to determine if the DUT supports End to End Data Protection, record DPC value.
3. Record the values in the DPS field of the Identify Namespace Data structure
4. Configure the NVMe Host to issue a Write command to the controller in order to write a known data pattern to an LBA range.
5. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was written in the previous step.
6. Configure the NVMe Host to issue a Format NVM Command with LBAF with non-zero metadata size supported by the DUT, SES=000b and FFFFFFFFh for the namespace ID, and PI=001b. If FNA bit 3 is set to '1h', issue a Format NVM Command to the current active namespace.
7. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was written in step 4.
8. Perform a Format NVM operation to restore the DPS settings for the Namespace that were recorded in Step 2.
9. Repeat for PI=010b and PI=011b.

Observable Results:

1. If End to End Data Protection Information is supported:
    a. 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.
    b. Verify that the same data pattern which was written to the NVM in step 4 is read from the NVM in step 5.
    c. Verify that the data pattern written in step 4 is not returned by the controller in step 7.
2. If End to End Data Protection is not supported, verify that the command completed with error status code 'Invalid Field in Command' (02h) or 'Invalid Format' (0Ah).
3. Verify that all received responses have all Reserved fields set to 0h.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.6.7 and expect the same observable results.

1. **Separate Buffer**
2. **Extended Buffer**
3. **MS = 0**

Sub-case Procedure:

6. Configure the NVMe Host to issue a Format NVM Command with LBAF with zero metadata size supported by the DUT, SES=000b and FFFFFFFFh for the namespace ID, and PI=001b. If FNA bit 3 is set to '1', issue a Format NVM Command to the current active namespace.

Case 8: Valid LBAF, SES=000b, PI=Type 3 (OF-FYI, FYI)

Test Procedure:

1. Check the OACS field of the Identify Controller Data structure to determine if the DUT supports the Format NVM command. If the command is not supported this test is not applicable.
2. Record the values in the DPS field of the Identify Namespace Data structure
3. 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.
4. Configure the NVMe Host to issue a Format NVM Command with an LBAF with non-zero metadata size supported by the DUT, SES=000b, PIL=1 or 0 depending on value supported by the DUT, PI=Type 3 and a valid namespace ID. Note: For implementations that are compliant with 1.0 or greater of the NVM Command Set Specification, this field shall be cleared to '0h'.
5. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was formatted in step 4, and bit 0 of PRCHK field set to 1h.

© 2026 UNH–IOL

6. Configure the NVMe Host to issue a Compare command to the controller specifying the same LBA range which was formatted in step 4, and bit 0 of PRCHK field set to 1h.
7. Configure the NVMe Host to issue a Write command to the controller specifying the same LBA range which was formatted in step 4, and bit 0 of PRCHK field set to 1h.
8. Check bit 3 of the ONCS, if bit 3 equals zero then check the WZSL field is set to zero then skip this procedure step. If bit 3 of ONCS is set to 1, then continue with this procedure step. Configure the NVMe Host to issue a Write Zeroes command to the controller specifying the same LBA range which was formatted in step 4, and bit 0 of PRCHK field set to 1h.
9. Perform a Format NVM operation to restore the DPS settings for the Namespace that were recorded in Step 2.

Observable Results:

1. Verify for version 1.4 and earlier devices that the Read, Compare, Write, and Write Zeroes commands all completed with status Invalid Protection Information or Invalid Field in Command. This functionality may be implemented by DUTs that support NVMe v1.3 or earlier but must be implemented by DUTs that support NVMe v1.4 or completed with a status of 'Successful Completion' (00h).
2. Verify for version 2.0 or later devices that the Read, Compare, Write and Write Zeroes commands all complete with a status of Invalid Protection Information, or complete with a status of 'Successful Completion' (00h).
3. Verify that all received responses have all Reserved fields set to 0h.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.6.8 and expect the same observable results.

1. **Separate Buffer**
2. **Extended Buffer**

Case 9: NSID=FFFFFFFFh, no attached namespaces, SES=000 (OF-FYI, FYI)

Test Procedure:

1. Check the OACS field and FNA field of the Identify Controller Data structure. If OACS field indicates that the DUT does not support the Format NVM command this test is not applicable. If bit 0 of the FNA field is set to 0, this test is not applicable. If FNA bit 3 is set to '1', then this test is not applicable.
2. Record the values in the DPS field of the Identify Namespace Data structure.
3. Configure the NVMe Host to issue a Write command to an attached namespace in order to write a known data pattern to an LBA range. If there is no namespaces attached, create a namespace and attach it.
4. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was written in step 4, then detach all namespaces.
5. Configure the NVMe Host to issue a Format NVM Command with an LBAF Supported by the DUT, SES=000b, NSID=FFFFFFFFh. Reattach the namespace written to in step 3.
6. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was written in step 4.
7. Perform a Format NVM operation to restore the DPS settings for the Namespace that were recorded in Step 2.

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 same data pattern which was written to the NVM in step 3 is read from the NVM in step 4.
3. Verify that the data pattern written in step 3 is not returned by the controller in step 6.
4. Verify that all received responses have all Reserved fields set to 0.

Case 10: NSID=FFFFFFFFh, no attached namespaces, SES!=000 (OF-FYI, FYI)

Test Procedure:

1. Check the OACS field and FNA field of the Identify Controller Data structure. If OACS field indicates that the DUT does not support the Format NVM command this test is not applicable. If bit 1 of the FNA field is set to 0, this test is not applicable. If FNA bit 3 is set to '1', then this test is not applicable.

© 2026 UNH–IOL

2. Record the values in the DPS field of the Identify Namespace Data structure.
3. Configure the NVMe Host to issue a Write command to an attached namespace in order to write a known data pattern to an LBA range.
4. Configure the NVMe Host to issue a Read command to the same namespace specifying the same LBA range which was written in step 3.
5. Configure the NVMe Host to issue a Namespace Attachment command to detach all namespaces from the controller.
6. Configure the NVMe Host to issue a Format NVM Command with an LBAF Supported by the DUT, NSID=FFFFFFFFh, and SES configured based on the following:
    a. If FNA bit 2 is set to 1 in the Identify Controller Data Structure, Cryptographic Erase is supported. Set SES to 010b.
    b. Otherwise (if FNA bit 2 is cleared to 0), set SES to 001b.
7. Configure the NVMe Host to issue a Namespace Attachment command to attach all namespaces back to the controller.
8. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA range which was written in step 3.
9. Perform a Format NVM command to restore the DPS settings for the namespace that were recorded in step 2.

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 same data pattern which was written to the NVM in step 3 is read from the NVM in step 4.
3. Verify that the data pattern written in step 3 is not returned by the controller in step 8.
4. Verify that all received responses have all Reserved fields set to 0.

Case 11: NSID=FFFFFFFFh Supported (OF-FYI, M)

Test Procedure:

1. Check the OACS field and FNA field of the Identify Controller Data structure. If OACS field indicates that the DUT does not support the Format NVM command this test is not applicable. If bit 3 of the FNA field is set to 1, this test is not applicable.
2. Configure the NVMe Host to issue a Format NVM Command with an LBAF Supported by the DUT, NSID=FFFFFFFFh.

Observable Results:

1. Verify that the Format Command completes with status Success.

Case 12: NSID=FFFFFFFFh Not Supported (OF-FYI, FYI)

Test Procedure:

1. Check the OACS field and FNA field of the Identify Controller Data structure. If OACS field indicates that the DUT does not support the Format NVM command this test is not applicable. If bit 3 of the FNA field is set to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Format NVM Command with an LBAF Supported by the DUT, NSID=FFFFFFFFh.

Observable Results:

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

Possible Problems: None known.

Case 13: Valid LBAFEE Value (OF-FYI, FYI)

Test Procedure:

© 2026 UNH–IOL

1. Configure the host to issue a Get Feature command for FID 16h. If FID 16h is not supported this test is not applicable.

Observable Results:

1. Verify that LBAFEE in the Host Behavior Support Data Structure is set to either 0h or 1h.

Case 14: LBA Format & Namespace Granularity (LBAFEE=1h, & ELBAS=1h) (OF-FYI, FYI)

Test Procedure:

1. Configure the host to issue a Get Feature for FID 16h. If FID 16h is not supported or LBAFEE in the Host Behavior Support – Data Structure is not set to 1h this test is not applicable.
2. Configure the host to issue an Identify Controller command, if ELBAS in the identify controller is not set to 1h this test is not applicable.
3. Configure the host to issue an Identify Namespace command with CNS 0h.
4. Configure the host to issue an Identify Namespace command with CNS 16h (Namespace Granularity List).

Observable Results:

1. Verify that the Identify Namespace command returns a total of NLBAF + NULBAF that is equal to 64 or less.
2. If Namespace Granularity List is supported verify that Number of Descriptors returned is equal to 64 or less.

Case 15: LBA Formats & Namespace Granularity (LBAFEE=0h) (OF-FYI, FYI)

Test Procedure:

1. Configure the host to issue a Get Feature for FID 16h. If FID 16h is not supported or LBAFEE in the Host Behavior Support – Data Structure is not set to 0h this test is not applicable.
2. Configure the host to issue an Identify Namespace command with CNS 0h.
3. Configure the host to issue an Identify Namespace command with CNS 16h.

Observable Results:

1. Verify that the Identify Namespace command returns a total of NLBAF + NULBAF that is equal to 16 or less.
2. If Namespace Granularity List is supported verify that Number of Descriptors value returned is equal to 16 or less.

Case 16: Format (LBAFEE=0h & ELBAS=1h) (OF-FYI, FYI)

Test Procedure:

1. Configure the host to issue an Identify command with CNS=01h, if ELBAS in the CTRATT field is set to 0h this test is not applicable.
2. Configure the host to Issue an Identify command with CNS=00h and NSID=FFFFFFFFh to obtain NLBAF, DPC, DPS, and the supported LBAF. If NLBAF is set to 0 this test is not applicable.
3. Configure the host to issue an Identify command with CNS=05h CSI=00h. If no ELBAF reports a PIF indicating:
    a. 16b Guard Protection Information with the STS field set to a non-zero value;
    b. 32b Guard Protection Information; or
    c. 64b Guard Protection Information

    then this test is not applicable

4. Configure the host to issue a Set Feature command for FID 16h to set LBAFEE to 0h.
5. Configure the NVMe Host to issue a Format command to an LBAF with a non-zero PIF value.
6. Configure the host to issue a Set Feature command for FID 16h to set LBAFEE to 1h.
7. Configure the NVMe Host to issue a Format command to an LBAF with a non-zero PIF value.

© 2026 UNH–IOL

Observable Results:
1. Verify that the Format command in step 5 aborts with Status code of "Invalid Namespace or Format".
2. Verify that all other commands complete successfully.

Possible Problems: If no currently created namespaces exist that support extended protection, this test may require deleting and creating a namespace.

Case 17: Format on Namespace that is Write Protected (OF-FYI, FYI)

Test Procedure:
1. Check Namespace Write Protection Capabilities (NWPC) in the Identify Controller Data Structure. If Bits 0, 1, 2 of NWPC are set to 0, the DUT does not support Namespace Write Protection, and this test is not applicable.
2. Determine if the namespace is Write Protected or perform a Set Feature operation for the Namespace Write Protection Config Feature (FID=84h) to set a particular namespace to a write protected state.
3. Configure the host to perform a Format command specifying a format other than that already on the namespace.

Observable Results:
1. Verify that Format command is aborted with a status code of 'Namespace is Write Protected' (20h).

Case 18: Format with an I/O Command in Progress (OF-FYI, M)

Test Procedure:
1. Check the OACS field of the Identify Controller Data structure to determine if the DUT supports the Format NVM command. If the command is not supported this test is not applicable.
2. Configure the NVMe Host to issue a Format NVM command specifying the current LBAF and Namespace Identifier.
3. Configure the NVMe Host to issue a Flush command to the same namespace as in the previous step 2.
4. Prior to the controller posting the completion queue entry to the I/O completion queue, configure the host to submit a Format NVM command to the same Namespace and LBAF as specified in steps 2 and 3.

Observable Results:
1. Verify that the Format NVM command in step 2 completed successfully.
2. Verify that the status field of the completion queue entry for the Format NVM command in step 4 has the status of 'Command Sequence Error' (0Ch).

Possible Problems: It is possible for the controller to wait and process the I/O command before sending the Format NVM command. The Format NVM command "may" be aborted, and if it is aborted it shall return with the status code of 'Command Sequence Error' (0Ch). In the case that the Format NVM command is not aborted, verify that the Format NVM command in step 4 completed successfully.

Case 19: I/O Command, with a Format Command in Progress (OF-FYI, M)

Test Procedure:
1. Check the OACS field of the Identify Controller Data structure to determine if the DUT supports the Format NVM command. If the command is not supported this test is not applicable.
2. Configure the NVMe Host to issue a Flush command specifying the current Namespace Identifier.
3. Configure the NVMe Host to issue a Format NVM command to the same namespace as in the previous step, specifying the current LBAF.
4. Prior to the controller posting the completion queue entry to the I/O completion queue, configure the host to submit another Flush command to the same Namespace specified in step 3.

Observable Results:
1. Verify that the Flush command in step 2 completed successfully.
2. Verify that the status field of the completion queue entry for the Flush command in step 4 has the status of 'Format in Progress' (84h).

© 2026 UNH–IOL

Possible Problems: It is possible for the controller to choose to not abort an I/O command while a format is in progress. If the controller does this, then the status 'Format in Progress' (84h) will not be returned. If this is the case, verify the I/O command completes successfully.

Case 20: Formatting Namespaces to non-zero STS when in a different format & LBAFEE = 1 (OF-FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Features command to FID 16h. If LBAFEE is set to 0h, this test is not applicable.
2. Configure the NVME Host to issue a NVM Command Set I/O Command Set Specific Identify Namespace (CSI 00h), and record a LBA Format that has a non-zero Storage Tag Size (STS).
3. Configure the NVMe Host to issue a Format NVM command to a namespace with a LBA Format that has a STS of 0h.
4. Configure the NVMe Host to issue a Format NVM command to the same namespace used in step 3 and which specifies the LBAF recorded in step 2.

Observable Results:

1. Verify that all commands expect the Format NVM command in step 4 complete with status 'Success'.
2. Verify that the Format NVM command aborts with a status code of 'Invalid Namespace or Format' (0Bh).

Possible Problems: Host may not indicate support in the Host Behavior Support Data Structure, and if that is the case, then this test is not applicable.

© 2026 UNH–IOL

Test 1.7 – Asynchronous Events (OF, M)

Purpose: To verify that an NVMe Controller can properly report asynchronous events to the host.

References:

Old Ref: [1] NVMe Specification 5.2
NVM Express Base specification 2.0a : 5.2
NVM Express NVM Command Set Specification 1.0a : 4

Resource Requirements:

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

Last Modification: November 20, 2018

Discussion: Asynchronous events are used to notify host software of status, error, and health information as these events occur. To enable asynchronous events to be reported by the controller, host software needs to submit one or more Asynchronous Event Request commands to the controller. The controller specifies an event to the host by completing an Asynchronous Event Request command. Host software should expect that the controller may not execute the command immediately; the command should be completed when there is an event to be reported.

Test Setup: See Appendix A.

Case 1: Asynchronous Event Request Command (M)

The Asynchronous Event Request command is submitted by host software to enable the reporting of asynchronous events from the controller. This command has no timeout. The controller posts a completion queue entry for this command when there is an asynchronous event to be reported to the host. If Asynchronous Event Request commands are outstanding when the controller is reset, the commands are aborted.

Host software may submit multiple Asynchronous Event Request commands to reduce event reporting latency. The total number of simultaneously outstanding Asynchronous Event Request commands is limited by the Asynchronous Event Request Limit specified in the Identify Controller data structure.

If the controller needs to report an event and there are no outstanding Asynchronous Event Request commands, the controller should send a single notification of that Asynchronous Event Type when an Asynchronous Event Request command is received. If a Get Log Page command clears the event prior to receiving the Asynchronous Event Request command or if a power off condition occurs, then a notification is not sent.

The following event types are defined in the NVMe Specification:

- Error Event
- SMART / Health Event
- Notice Events
- I/O Command Set Specification (NVM Command Set) Events:
    - Reservation Log Page Available Event
- Vendor Specific Events

All command specific fields are reserved.

Test Procedure:

1. For each event type described above:
    a. Configure the NVMe Host to issue an Asynchronous Event Request command to the NVMe Controller.
    b. Configure the NVMe Host to cause conditions for generating the event for the NVMe Controller as described in the NVMe Specification.

Observable Results:

© 2026 UNH–IOL

1. Verify that the NVMe Controller does not post a completion queue entry to the Admin Completion Queue for the Asynchronous Event Request command until after the event is generated.
2. Verify that the completion queue entry for the Asynchronous Event Request command is properly formatted and contains information appropriate for the event in Dword 0 as described in the NVMe Specification.
3. Verify that all received responses have all Reserved fields set to 0h.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.7.1 and expect the same observable results.

1. Error Event

Observable Result:

4. Verify that the completion queue entry for the Error event contains a Log Page Identifier of 01h, an Asynchronous Event Type of 000b, and the proper Error Status associated with the generated event.

2. SMART / Health Event

Observable Result:

4. Verify that the completion queue entry for the SMART / Health Status event contains a Log Page Identifier of 02h, an Asynchronous Event Type of 001b, and the proper SMART / Health Status associated with the generated event.

3. Notice Events

Observable Result:

4. Verify that the completion queue entry for the Notice event contains an Asynchronous Event Type of 010b and the proper Log Page Identifier and Asynchronous Event Information for the generated event.

4. Reservation Log Page Available Event

Case 2: Outstanding Commands Aborted after Reset (OF, M)

If Asynchronous Event Request commands are outstanding when the controller is reset, the commands are aborted.

Test Procedure:

1. Configure the NVMe Host to issue an Asynchronous Event Request command to the NVMe Controller.
2. Configure the NVMe Host to initiate a Controller Level Reset for the NVMe Controller.

Observable Results:

1. Verify that after the Controller Level Reset is initiated, if a completion queue entry is posted by the NVMe Controller to the Admin Completion Queue and the status for the command is 07h Command Abort Requested. If no completion queue entry is posted, then this test is not applicable.
2. Verify that all received responses have all Reserved fields set to 0.

Case 3: Clearing Events (M)

If the controller needs to report an event and there are no outstanding Asynchronous Event Request commands, the controller should send a single notification of that Asynchronous Event Type when an Asynchronous Event Request command is received. If a Get Log Page command clears the event prior to receiving the Asynchronous Event Request command or if a power off condition occurs, then a notification is not sent.

When the controller posts a completion queue entry for an outstanding Asynchronous Event Request command and thus reports an asynchronous event, subsequent events of that event type are automatically masked by the controller until the host clears that event. An event is cleared by reading the log page associated with that event using the Get Log Page command.

Test Procedure:

1. For each event type associated with a log page:

© 2026 UNH–IOL

a. Configure the NVMe Host to cause conditions for generating the event for the NVMe Controller as described in the NVMe Specification.
b. Configure the NVMe Host to issue a Get Log Page command to the NVMe Controller for the log page associated with the event, with the Retain Asynchronous Event bit cleared to '0', in order to clear that event.
c. Configure the NVMe Host to issue an Asynchronous Event Request command to the NVMe Controller.
d. After a timeout period of 2 seconds, configure the NVMe Host to clear the outstanding Asynchronous Event Request command by initiating a Controller Level Reset for the NVMe Controller.

Observable Results:

1. Verify that after the completion of each Get Log Page command, the controller posts a completion queue entry to the associated Completion Queue indicating the status for the command.
2. Verify that the NVMe Controller does not post a completion queue entry to the Admin Completion Queue for the Asynchronous Event Request command until after the Controller Level Reset.
3. Verify that each of the Asynchronous Events are cleared due to the Get Log with RAE=0 for each event type
4. Verify that all received responses have all Reserved fields set to 0.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.7.3 and expect the same observable results.

1. **Error Event**
2. **SMART / Health Event**
3. **Notice Events**
4. **Reservation Log Page Available Event**

Case 4: Masking Events (OF, M)

The Asynchronous Event Configuration feature controls the events that trigger an asynchronous event notification to the host. This Feature may be used to disable reporting events in the case of a persistent condition. The attributes are indicated in Command Dword 11.

If a Get Features command is submitted for this Feature, the attributes specified in for Command Dword 11 of the Set Features command are returned in Dword 0 of the completion queue entry for that command.

Test Procedure:

1. For each event which may be disabled with the Asynchronous Event Configuration feature:
    a. Configure the NVMe Host to issue a Set Features command to the NVMe Controller with feature identifier 0Bh (Asynchronous Event Configuration) and formatted to disable the event.
    b. Configure the NVMe Host to issue an Asynchronous Event Request command to the NVMe Controller.
    c. Configure the NVMe Host to cause conditions for generating the event for the NVMe Controller as described in the NVMe Specification.
    d. After a timeout period of 2 seconds, configure the NVMe Host to clear the outstanding Asynchronous Event Request command by initiating a Controller Level Reset for the NVMe Controller.

Observable Results:

1. Verify that after the completion of each Set Features command, the controller posts a completion queue entry to the associated Completion Queue indicating the status for the command.
2. Verify that the NVMe Controller does not post a completion queue entry to the Admin Completion Queue for the Asynchronous Event Request command until after the Controller Level Reset and that the status indicates that the command was aborted and contains no information about the event in Dword 0.
3. Verify that all received responses have all Reserved fields set to 0.

© 2026 UNH–IOL

Possible Problems: Some events are optional as reported by the Optional Asynchronous Events Supported (OAES) field of the Identify Controller data structure or by other means and are therefore only tested if those features are supported.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.7.4 and expect the same observable results.

1. **SMART / Health Event**
2. **Notice Events**

Case 5: Asynchronous Events not Cleared (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to cause conditions for generating the event for the NVMe Controller as described in the NVMe Specification, such as a Namespace Attribute Changed.
2. Configure the NVMe Host to send a Get Log Page for that event, such as Changed Namespace List (04h), with an invalid field and the retain Asynchronous Event field in Dword0 cleared to '0'.
3. Configure the NVMe Host to issue an Asynchronous Event Request command to the NVMe Controller for the generated event, such as Namespace Attribute Changed.

Observable Results:

1. Verify that the Asynchronous Event is created.
2. Verify that the Get Log Page returns a status code other than success and that the generated Asynchronous Event is still present.
3. Verify that after the Asynchronous Event Request command the generated event is not cleared.

Possible Problems: The Changed Namespace List Get Log Page is an optional command.

Case 6: Asynchronous Event Sanitize Operation Entered Media Verification State (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bit 3 of SANICAP, VERS, is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h) with Retain Asynchronous Event (RAE) set to 0b to clear any previous Sanitize Operation Entered Media Verification State.
3. Configure the NVMe Host to issue an Asynchronous Event Request command to the NVMe Controller.
4. Configure the NVMe Host to issue a Sanitize command with Enter Media Verification State (EMVS) set to 1b and the Sanitize Action (SANACT) set to either Block Erase (010b) or Crypto Erase (100b).
5. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h) with Retain Asynchronous Event (RAE) set to 1b. Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that there is a completion queue entry for the Asynchronous Event Request command following the sanitize operation's completion with Log Page Identifier set to 81h, Asynchronous Event Type set to 110b, and Asynchronous Event Information set to 03h indicating 'Sanitize Operation Entered Media Verification State'.

Case 7: Sanitize Operation Completed (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bits 2:0 of SANICAP are cleared to 0h this test is not applicable.

© 2026 UNH–IOL

2. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h) with Retain Asynchronous Event (RAE) set to 0h to clear any previous Sanitize Asynchronous Events.
3. Configure the NVMe Host to issue an Asynchronous Event Request command to the NVMe Controller.
4. Configure the NVMe Host to issue a Sanitize command with a Sanitize Action (SANACT) reported in the SANICAP field.
5. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h) with Retain Asynchronous Event (RAE) set to 1b. Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that there is a completion queue entry for the Asynchronous Event Request command following the sanitize operation's completion with Log Page Identifier set to 81h, Asynchronous Event Type set to 110b, and Asynchronous Event Information indicating status 'Sanitize Operation Completed' (01h).

Sub-Cases: For the following subcase, unless otherwise noted, perform the same test procedure for case 1.7.7 and expect the same observable results.

1. EVNTSP set to NSID of the Sanitization Target

Sub-case Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID 05h). If bit 0 of the Commands Supported and Effects Data Structure for the Sanitize Namespace command (opcode 8Ch) is cleared to 0b then this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize Namespace command with Sanitize Action (SANACT) field set to 100b (Start a Crypto Erase namespace sanitize operation), in place of the Sanitize command.
3. Verify that completion queue entry DWORD 1 is set to the NSID of the sanitization target from the Sanitize Namespace command.

Case 8: Sanitize Operation Unexpected Deallocation (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bits 2:0 of SANICAP are cleared to 0h or No-Deallocate Inhibited (NDI) is cleared to 0b this test is not applicable.
2. Configure the NVMe Host to issue an Get Features command to Sanitize Config (FID 17h). If No-Deallocate Response Mode (NODRM) is disabled (01b), configure the NVMe Host to issue a Set Features command to set NODRM to 1b.
3. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h) with Retain Asynchronous Event (RAE) set to 0 to clear any previous Sanitize Asynchronous Events.
4. Configure the NVMe Host to issue an Asynchronous Event Request command to the NVMe Controller.
5. Configure the NVMe Host to issue a Sanitize command with No-Deallocate After Sanitize (NDAS) set to 1b and a Sanitize Action (SANACT) reported in the SANICAP field.
6. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h) with Retain Asynchronous Event (RAE) set to 1b. Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh
7.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that there is a completion queue entry for the Asynchronous Event Request command following the sanitize operation's completion with Log Page Identifier set to 81h, Asynchronous Event Type set to 110b, and Asynchronous Event Information set to 02h indicating 'Sanitize Operation Completed With Unexpected Deallocation'

© 2026 UNH–IOL

Case 9: Sanitize Clears Events (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. Record the supported Sanitize Capabilities, if SANICAP is cleared to 0h this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h) with Retain Asynchronous Event (RAE) set to 0b to clear any previous Sanitize Asynchronous Events.
3. Configure the NVMe Host to issue a Sanitize command with a Sanitize Action (SANACT) reported in the SANICAP field to generate the Asynchronous Event.
4. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h) with Retain Asynchronous Event (RAE) set to 1b. Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.
5. Configure the NVMe Host to issue a Sanitize command with a Sanitize Action (SANACT) reported in the SANICAP field to clear the Asynchronous Event.
6. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h) with Retain Asynchronous Event (RAE) set to 1b. Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.
7. Configure the NVMe Host to issue two Asynchronous Event Request commands to the NVMe Controller.
8. Wait 15 seconds.
9. Configure the NVMe host to issue a Controller Level Reset.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that the first Asynchronous Event Request command posted a completion queue entry.
3. Verify that the second Asynchronous Event Request command did not post a completion queue entry or posted a completion queue entry with the error status of: ‘Command Abort Requested’ (07h).

Possible Problems: If the device fails to clear the Sanitize Operation Completed event at the start of the second Sanitize command, the two identical outstanding events may be reported as a single response to the Asynchronous Event Request command. This behavior is indistinguishable from the desired response to the above test procedure.

Sub-Cases: For each of the following subcases of test 1.7.9, follow the test procedure of test case 9, except for the modified steps as listed below:

1. Sanitize Operation Completed (AE Information – I/O Command Specific Status, 01h)

Sub-case Procedure:

1. Follow steps as described in 1.7.9.

2. Sanitize Operation Completed with Unexpected Deallocation (AE Information – I/O Command Specific Status, 02h)

Sub-case Procedure:

3. Configure the NVMe Host to issue a Sanitize command with No-Deallocate After Sanitize (NDAS) set to 1b and a Sanitize Action (SANACT) reported in the SANICAP field.

3. Sanitize Operation Entered Media Verification State (AE Information – I/O Command Specific Status, 03h)

Sub-case Procedure:

3. Configure the NVMe Host to issue a Sanitize command with Enter Media Verification State (EMVS) set to 1b and the Sanitize Action (SANACT) set to either Block Erase (010b) or Crypto Erase (100b).

Case 10: Attached and Allocated Namespace Attribute Notice Asynchronous Event (OF-FYI, FYI)

Test Procedure:

© 2026 UNH–IOL

1. Configure the NVMe Host to check bit 3 of the OACS field in the Identify Controller data structure to determine if the DUT supports Namespace Management. If Bit 3 is not set to 1, this test is not applicable.
2. Check the NN value reported by the DUT in the Identify Controller Data Structure. If NN is not greater than 1 then this test is not applicable.
3. Perform a Namespace Management command with action of Create, but do not attach the namespace to the controller.
4. Configure the NVMe host to issue a Set Feature command for FID Asynchronous Event Configuration (0Bh) with bit 8 set to 1 in Dword 11 to enable Attached Namespace Attribute Notices for every controller attached to the namespace that will be deleted in step 6.
5. Configure the NVMe host to issue a Set Feature command for FID Asynchronous Event Configuration (0Bh) with bit 19 set to 1 in Dword 11 to enable Allocated Namespace Attribute Notices for every controller that is not attached to the namespace that will be deleted in step 6.
6. Perform a Namespace Management command with action of Delete, and NSID=FFFFFFFFh.

Observable Results:

1. Verify that each controller that was attached to the namespace issues an Attached Namespace Attribute Changed Asynchronous Event after step 3.
2. Verify that each controller that was not attached to the namespace issues an Allocated Namespace Attribute Changed Asynchronous Event after step 5.

Case 11: Controller Data Queue Tail Pointer Event Triggered (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 1 of TRATTR (Track User Data Changes Support) is cleared to 0, this test is not applicable.
2. Configure the NVMe host to issue a Get Log command with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS69) is cleared to 0h, this test is not applicable.
3. Configure the NVMe Host to issue a Controller Data Queue Command with the Select (SEL) set to Create Controller Data Queue (Management Operation 0h), and the Management Operation Specific (MOS) set to 0h (User Data Migration Queue), a CDQSIZE to 10h, and the Create Queue Specific field set to Controller B.
4. Configure the NVMe Host to issue a Track Send command with the SEL = 0h (Log User Data Changes), LACT = 1h, and CDQID set to the queue created in step 2.
5. Configure the NVMe Host to issue a Set Features command specifying feature identifier 21h (Controller Data Queue) with Head Pointer (HP) set to 1h, the Enable Tail Pointer Trigger (ETPT) bit set to 1h, the Tail Pointer Trigger (TPT) set to 0.
6. Configure the NVMe Host to issue a Track Send command with the SEL = 0h (Log User Data Changes), LACT = 0h, and CDQID set to the queue created in step 2.
7. Configure the NVMe host to issue an Asynchronous Event Request Command with the controller identifier of the migratable controller.
8. Configure the NVMe Host to issue a Get Features command for FID 21h. Read the returned data structure

Observable Results:

1. Verify all commands complete successfully.
2. Verify that in the command DWORD 0 of the Asynchronous Event Command in step 6 indicates a tail pointer event occurred.
3. Verify that the Enable Tail Pointer Trigger (ETPT) returned in DWORD 0 of the Get Features command from step 8 is cleared to 0.

© 2026 UNH–IOL

Test 1.8 – Get Feature Select (M)

Purpose: To verify that an NVMe Controller can properly execute a Get Features command with the Select Field set.

References:

Old Ref: [1] NVMe Specification 5.9.1
NVM Express Base specification 2.0a : 4.2

Resource Requirements:

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

Last Modification: June 9, 2016

Discussion: A Select field set to 000b (i.e., current) returns the current operating attribute value for the Feature Identifier specified.

A Select field set to 001b (i.e., default) returns the default attribute value for the Feature Identifier specified.

A Select field set to 010b (i.e., saved) returns the last saved attribute value for the Feature Identifier specified (i.e., the last Set Features command completed without error, with the Save bit set to '1' for the Feature Identifier specified.)

A Select field set to 011b (i.e., supported capabilities) returns the capabilities supported for this Feature Identifier. The capabilities supported are returned in Dword 0 of the completion entry of the Get Features command.

A completion queue entry is posted to the Admin Completion Queue when the controller has completed returning any attributes associated with the Feature. Depending on the Feature Identifier, Dword 0 of the completion queue entry may contain feature information

Test Setup: See Appendix A.

Case 1: Get Feature Select (M)

Test Procedure:

1. For each of the features described in Table 1 and Table 2 (see Test 1.2):
    a. Configure the NVMe Host to issue a Get Features command indicating a Select Field of 000b.
    b. Configure the NVMe Host to issue a Get Features command indicating a Select Field of 001b.
    c. Configure the NVMe Host to issue a Set Features command, to specify a new value for each FID. Next, configure the NVMe Host to issue a Get Features command indicating a Select Field of 010b.
    d. Configure the NVMe Host to issue a Get Features command indicating a Select Field of 011b.

Observable Results:

1. For Step 1a, verify that the DUT returns the current operating attribute value for the Feature Identifier specified.
2. For Step 1b, verify that the DUT returns the default attribute value for the Feature Identifier specified.
3. For Step 1c, verify that the DUT returns the last saved attribute value for the Feature Identifier Specified. This must match the value assigned in Step 1c.
4. For Step 1d, verify that the DUT returns the capabilities supported for this Feature Identifier. The capabilities supported are returned in Dword 0 of the completion entry of the Get Features command.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.8.1 and expect the same observable results.

1. **FID 01h Arbitration**
2. **FID 02h Power Management**

© 2026 UNH–IOL

3. **FID 03h LBA Range Type**
4. **FID 04h Temperature Threshold**
5. **FID 05h Error Recovery**
6. **FID 06h Volatile Write Cache**
7. **FID 07h Number of Queues**
8. **FID 08h Interrupt Coalescing**
9. **FID 09h Interrupt Vector Configuration**
10. **FID 0Ah Write Atomicity Normal**
11. **FID 0Bh Asynchronous Event Configuration**
12. **FID 0Ch Autonomous Power State Transition**
13. **FID 0Dh Host Memory Buffer**
14. **FID 0Eh Timestamp**
15. **FID 0Fh Keep Alive Timer**
16. **FID 10h Host Controlled Thermal Management**
17. **FID 11h Non-Operational Power State Config**
18. **FID 12h Read Recovery Level Config**
19. **FID 13h Predictable Latency Mode Config**
20. **FID 14h Predictable Latency Mode Window**
21. **FID 15h LBA Status Information Report Interval**
22. **FID 16h Host Behavior Support**
23. **FID 17h Sanitize Config**
24. **FID 18h Endurance Group Event Configuration**
25. **FID 19h I/O Command Set Profile**
26. **FID 1Ah Spinup Control**
27. **FID 7Dh Enhanced Controller Metadata**
28. **FID 7Eh Controller Metadata**
29. **FID 7Fh Namespace Metadata**
30. **FID 80h Software Progress Marker**
31. **FID 81h Host Identifier**
32. **FID 82h Reservation Notification Mask**
33. **FID 83h Reservation Persistence**
34. **FID 84h Namespace Write Protection Config**
35. **FID 24h Power Threshold**
36. **FID 25h Power Measurement**

© 2026 UNH–IOL

Test 1.9 – Feature Saved Across Reset (M)

Purpose: To verify that an NVMe Controller can properly save a Feature value set by a Host across a reset.

References:

Old Ref: [1] NVMe Specification 7.8
NVM Express Base specification 2.0a : 4.2

Resource Requirements:

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

Last Modification: April 2, 2019

Discussion: If bit 4 is set to '1' in the Optional NVM Command Support field of the Identify Controller data structure, then for each Feature, there are three settings: default, saveable, and current.

The saveable value is the value that the Feature has after a power on or reset event. The controller may not support a saveable value for a Feature; this is discovered by using the 'supported capabilities' value in the Select field in Get Features. If the controller does not support a saveable value for a Feature, then the default value is used after a power on or reset event. The current value is the value actively in use by the controller for a Feature after a Set Features command completes.

This test is only applicable to devices which set the Save field (Bit 4) of the ONCS field to 1.

Test Setup: See Appendix A.

Case 1: Feature Saved Across Reset (M)

Test Procedure:

1. Check bit 4 of the ONCS field. If bit 4 is set to 1, proceed to the next step. If not the test cannot be performed.
2. For each of the features described in Table 1 and Table 2 (see Test 1.2):
    a. Determine which features the DUT indicates as both 'Saveable' and 'Changeable'.
    b. Configure the NVMe Host to issue a Get Features command indicating a Select Field of 000b, to obtain the current operating value of the Feature.
    c. Configure the NVMe Host to issue a Set Features command with the Save bit set to 1 (SV=1), to specify a new value for each FID that the DUT indicates as both 'Saveable' and 'Changeable'.
    d. Configure the NVMe Host to issue a Get Features command indicating a Select Field of 000b, to obtain the current operating value of the Feature.
    e. Perform an NVMe Subsystem Reset (NSSR) or another supported reset type.
    f. Configure the NVMe Host to issue a Get Features command indicating a Select Field of 000b, to obtain the current operating value of the Feature.

Observable Results:

1. For each Feature tested, verify that the value reported in Step 2e, matches the value set by the Host in Step 2b. prior to the reset.

Possible Problems: Not all devices will support NSSR, and other reset methods may be necessary. When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 1.9.1 and expect the same observable results.

1. **FID 03h LBA Range Type**
2. **FID 10h Host Controlled Thermal Management**
3. **FID 12h Read Recovery Level Config**

© 2026 UNH–IOL

4. **FID 17h Sanitize Config**
5. **FID 19h I/O Command Set Profile**
6. **FID 1Ah Spinup Control**
7. **FID 1Bh Power Loss Signaling Config**
8. **FID 1Ch Performance Characteristics**
9. **FID 1Dh Flexible Data Placement**
10. **FID 1Eh Flexible Data Placement Events**
11. **FID 1Fh Namespace Admin Label**
12. **FID 78h Embedded Management Controller Address**
13. **FID 79h Host Management Agent Address**
14. **FID 80h Software Progress Marker**
15. **FID 83h Reservation Persistence**
16. **FID 24h Power Threshold**
17. **FID 25h Power Measurement**

© 2026 UNH–IOL

Test 1.10 – Device Self-test Short Operation (OF-FYI, M)

Purpose: To verify that an NVMe Controller can properly perform a Device Self-test Short Operation.

References:

Old Ref: [1] NVMe Specification 5.8
NVM Express Base specification 2.0a : 5.9
NVM Express NVM Command Set Specification 1.0a : 4.1.4.3

Resource Requirements:

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

Last Modification: December 14, 2017

Discussion: The Device Self-test command is used to start a device self-test operation or abort a device self-test operation. The device self-test operation is performed by the controller that the Device Self-test command was submitted to.

Test Setup: See Appendix A.

Case 1: Namespace Test Action = 00000000h, STC=1h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host to send a Device Self-test command with Namespace Test Action set to 00000000h and STC set to 1h (Short device self-test operation).
4. Before the Device Self-test operation finishes, send a Get Log Page command to LID 06h (Device Self-test). Note that since the Device Self Test operation can be performed in the background, a command completion may be received before the Device Self Test operation finishes. Repeat this step until the returned log page indicates that the operation has finished.
5. Report to the user the time elapsed between sending of the Device Self-test command and the operation finishing.

Observable Results:

1. Using the Log Page info returned in response to the first Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 1h when the operation was processing.
2. Verify that the Device Self-test command completes successfully.
3. Using the Log Page info returned in response to the second Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 0h when the operation was completed.
4. Verify that all received responses have all Reserved fields set to 0.

Case 2: Namespace Test Action = 00000001h-FFFFFFFEh, STC=1h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.

© 2026 UNH–IOL

2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to all value between 00000001h and FFFFFFFEh that represent a valid Namespace on the DUT, with STC set to 1h (Short device self-test operation).
4. Before the Device Self-test operation finishes, send a Get Log Page command to LID 06h (Device Self-test). Note that since the Device Self Test operation can be performed in the background, a command completion may be received before the Device Self Test operation finishes. Repeat this step until the returned log page indicates that the operation has finished.
5. Report to the user the time elapsed between sending of the Device Self-test command and the operation finishing.

Observable Results:

1. Using the Log Page info returned in response to the first Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 1h when the operation was processing.
2. Verify that the Device Self-test command completes successfully.
3. Using the Log Page info returned in response to the second Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 0h when the operation was completed.
4. Verify that all received responses have all Reserved fields set to 0.

Case 3: Namespace Test Action = FFFFFFFFh, STC=1h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to FFFFFFFFh, with STC set to 1h (Short device self-test operation).
4. Before the Device Self-test operation finishes, send a Get Log Page command to LID 06h (Device Self-test). Note that since the Device Self Test operation can be performed in the background, a command completion may be received before the Device Self Test operation finishes. Repeat this step until the returned log page indicates that the operation has finished.
5. Report to the user the time elapsed between sending of the Device Self-test command and the operation finishing.

Observable Results:

1. Using the Log Page info returned in response to the first Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 1h when the operation was processing.
2. Verify that the Device Self-test command completes successfully.
3. Using the Log Page info returned in response to the second Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 0h when the operation was completed.
4. Verify that all received responses have all Reserved fields set to 0.

© 2026 UNH–IOL

Case 4: Namespace Test Action = Invalid Namespace, STC=1h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to an Invalid Namespace, with STC set to 1h (Short device self-test operation).
4. Send a Get Log Page command to LID 06h (Device Self-test). Note that since the Device Self Test operation can be performed in the background, a command completion may be received before the Device Self Test operation finishes. Repeat this step until the returned log page indicates that the operation has finished.

Observable Results:

1. Verify that the Device Self-test command returns status Invalid Namespace or Format in Command.
2. Using the Log Page info returned in response to the second Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 0h when the operation was completed.

Case 5: Namespace Test Action = Inactive Namespace, STC=1h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Configure the Testing Station acting as a host to Check Bit 3 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports Namespace Management. If Bit 3 is not set to 1, this test is not applicable.
3. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
4. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to an Inactive Namespace, with STC set to 1h (Short device self-test operation).
5. Send a Get Log Page command to LID 06h (Device Self-test). Note that since the Device Self Test operation can be performed in the background, a command completion may be received before the Device Self Test operation finishes. Repeat this step until the returned log page indicates that the operation has finished.

Observable Results:

1. Verify that the Device Self-test command returns status Invalid Field in Command.
2. Using the Log Page info returned in response to the second Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 0h when the operation was completed.

Case 6: Namespace Test Action = 00000000h, STC=1h and 1h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.

© 2026 UNH–IOL

2. Ensure that no Device Self-test operation is currently in progress by sending an initial Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host to send an initial Device Self-test command with Namespace Test Action set to 00000000h and STC set to 1h (Short device self-test operation). Wait for a Completion to be received with status 'Success'.
4. Determine that the initial Device Self-test command is in progress by sending a second Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 1 (Short device self-test operation in progress). If the Current Device Self-Test Operation field is set to 0, the Device Self-Test Operation is competing too quickly for the test to be performed and this test is not applicable.
5. Before the first Device Self-test operation completes, send a second Device Self-test command with Namespace Test Action set to 00000000h and STC set to 1h (Short device self-test operation). Wait for a Completion to be received.
6. Send a third Get Log Page command to LID 06h (Device Self-test). Repeat until the current Device Self-Test operation field is set to 0 in the Log Page that is returned.

Observable Results:

1. Using the Log Page info returned in response to the second Get Log Page command, which followed the initial Device Self-test command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 1h when the operation was processing.
2. Verify that the completion for the initial Device Self-test command indicates status 'Success' (00h).
3. Verify that the completion for the second attempted Device Self-test command indicates status 'Device Self-test in Progress' (1Dh).
4. Using the Log Page info returned in response to the third or later Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 0h when the operation was completed.
5. Verify that all received responses have all Reserved fields set to 0.

Case 7: Namespace Test Action = 00000001h-FFFFFFFEh, STC=1h and 1h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host to send an initial Device Self-test command with Namespace Test Action set to a value between 00000001h and FFFFFFFEh and STC set to 1h (Short device self-test operation). Wait for a Completion to be received with status 'Success' (00h).
4. Determine that the initial Device Self-test command is in progress by sending a second Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 1 (Short device self-test operation in progress). If the Current Device Self-Test Operation field is set to 0, the Device Self-Test Operation is competing too quickly for the test to be performed and this test is not applicable.
5. Before the first Device Self-test operation completes, send a second Device Self-test command with Namespace Test Action set to a value between 00000001h and FFFFFFFEh (use the same value as in the previous Device Self Test command) and STC set to 1h (Short device self-test operation). Wait for a Completion to be received.
6. Send a third Get Log Page command to LID 06h (Device Self-test). Repeat until the current Device Self-Test operation field is set to 0 in the Log Page that is returned.

© 2026 UNH–IOL

Observable Results:

1. Using the Log Page info returned in response to the second Get Log Page command, which followed the initial Device Self-test command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 1h when the operation was processing.
2. Verify that the completion for the initial Device Self-test command indicates status 'Success' (00h).
3. Verify that the completion for the second attempted Device Self-test command indicates status 'Device Self-test In Progress' (1Dh).
4. Using the Log Page info returned in response to the third or later Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 0h when the operation was completed.
5. Verify that all received responses have all Reserved fields set to 0.

Case 8: Namespace Test Action = FFFFFFFFh, STC=1h, Second DST (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1h, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0h. If the Current Device Self-Test Operation field is not set to 0h, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host to send an initial Device Self-test command with Namespace Test Action set to FFFFFFFFh and STC set to 1h (Short device self-test operation). Wait for a Completion to be received with status 'Success' (00h).
4. Determine that the initial Device Self-test command is in progress by sending a second Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 1h (Short device self-test operation in progress). If the Current Device Self-Test Operation field is set to 0, the Device Self-Test Operation is competing too quickly for the test to be performed and this test is not applicable.
5. Before the first Device Self-test operation completes, send a second Device Self-test command with Namespace Test Action set to FFFFFFFFh and STC set to 1h (Short device self-test operation). Wait for a Completion to be received.
6. Send a third Get Log Page command to LID 06h (Device Self-test). Repeat until the current Device Self-Test operation field is set to 0h in the Log Page that is returned.

Observable Results:

1. Using the Log Page info returned in response to the second Get Log Page command, which followed the initial Device Self-test command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 1h when the operation was processing.
2. Verify that the completion for the initial Device Self-test command indicates status 'Success' (00h).
3. Verify that the completion for the second attempted Device Self-test command indicates status 'Device Self-test In Progress' (1Dh).
4. Using the Log Page info returned in response to the third or later Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 0h when the operation was completed.
5. Verify that all received responses have all Reserved fields set to 0h.

Possible Problems: None known.

© 2026 UNH–IOL

Test 1.11 – Device Self-test Extended Operation (OF-FYI, M)

Purpose: To verify that an NVMe Controller can properly perform a Device Self-test Extended Operation.

References:

Old Ref: [1] NVMe Specification 5.8
NVM Express Base specification 2.0a : 5.9
NVM Express NVM Command Set Specification 1.0a : 4.1.4.3

Resource Requirements:

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

Last Modification: December 14, 2017

Discussion: The Device Self-test command is used to start a device self-test operation or abort a device self-test

Operation. The device self-test operation is performed by the controller that the Device Self-test command was submitted to.

Test Setup: See Appendix A.

Case 1: Namespace Test Action = 00000000h, STC=2h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host to send a Device Self-test command with Namespace Test Action set to 00000000h and STC set to 2h (Extended device self-test operation).
4. Before the Device Self-test operation finishes, send a Get Log Page command to LID 06h (Device Self-test). Note that since the Device Self Test operation can be performed in the background, a command completion may be received before the Device Self Test operation finishes. Repeat this step until the returned log page indicates that the operation has finished.
5. Report to the user the time elapsed between sending of the Device Self-test command and the operation finishing.

Observable Results:

1. Using the Log Page info returned in response to the first Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 2h when the operation was processing.
2. Verify that the Device Self-test command completes successfully.
3. Using the Log Page info returned in response to the second Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 0h when the operation was completed.
4. Verify that all received responses have all Reserved fields set to 0.

Case 2: Namespace Test Action = 00000001h-FFFFFFFEh, STC=2h (OF-FYI, M)

Test Procedure:

© 2026 UNH–IOL

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to all value between 00000001h and FFFFFFFEh that represent a valid Namespace on the DUT, with STC set to 2h (Extended device self-test operation).
4. Before the Device Self-test operation finishes, send a Get Log Page command to LID 06h (Device Self-test). Note that since the Device Self Test operation can be performed in the background, a command completion may be received before the Device Self Test operation finishes. Repeat this step until the returned log page indicates that the operation has finished.
5. Report to the user the time elapsed between sending of the Device Self-test command and the operation finishing.

Observable Results:

1. Using the Log Page info returned in response to the first Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 2h when the operation was processing.
2. Verify that the Device Self-test command completes successfully.
3. Using the Log Page info returned in response to the second Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 0h when the operation was completed.
4. Verify that all received responses have all Reserved fields set to 0.

Case 3: Namespace Test Action = FFFFFFFFh, STC=2h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to FFFFFFFFh, with STC set to 2h (Extended device self-test operation).
4. Before the Device Self-test operation finishes, send a Get Log Page command to LID 06h (Device Self-test). Note that since the Device Self Test operation can be performed in the background, a command completion may be received before the Device Self Test operation finishes. Repeat this step until the returned log page indicates that the operation has finished.
5. Report to the user the time elapsed between sending of the Device Self-test command and the operation finishing.

Observable Results:

1. Using the Log Page info returned in response to the first Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 2h when the operation was processing.
2. Verify that the Device Self-test command completes successfully.
3. Using the Log Page info returned in response to the second Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 0h when the operation was completed.

© 2026 UNH–IOL

4. Verify that all received responses have all Reserved fields set to 0.

Case 4: Namespace Test Action = Invalid Namespace, STC=2h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to an Invalid Namespace, with STC set to 2h (Extended device self-test operation).
4. Send a Get Log Page command to LID 06h (Device Self-test). Note that since the Device Self Test operation can be performed in the background, a command completion may be received before the Device Self Test operation finishes. Repeat this step until the returned log page indicates that the operation has finished.

Observable Results:

1. Verify that the Device Self-test command returns status Invalid Namespace or Format in Command.
2. Using the Log Page info returned in response to the second Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 0h when the operation was completed.

Case 5: Namespace Test Action = Inactive Namespace, STC=2h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Configure the Testing Station acting as a host to Check Bit 3 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports Namespace Management. If Bit 3 is not set to 1, this test is not applicable. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to an Inactive Namespace, with STC set to 2h (Extended device self-test operation).
4. Send a Get Log Page command to LID 06h (Device Self-test). Note that since the Device Self Test operation can be performed in the background, a command completion may be received before the Device Self Test operation finishes. Repeat this step until the returned log page indicates that the operation has finished.

Observable Results:

1. Verify that the Device Self-test command returns status Invalid Field in Command.
2. Using the Log Page info returned in response to the second Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 0h when the operation was completed.

Case 6: Namespace Test Action = 00000000h, STC=2h and 2h (OF-FYI, M)

Test Procedure:

© 2026 UNH–IOL

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host to send an initial Device Self-test command with Namespace Test Action set to 00000000h and STC set to 2h (Extended device self-test operation). Wait for a Completion to be received with status "Success".
4. Determine that the initial Device Self-test command is in progress by sending a second Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 2 (Extended device self-test operation in progress). If the Current Device Self-Test Operation field is set to 0, the Device Self-Test Operation is competing too quickly for the test to be performed and this test is not applicable.
5. Before the first Device Self-test operation completes, send a second Device Self-test command with Namespace Test Action set to 00000000h and STC set to 2h (Extended device self-test operation). Wait for a Completion to be received.
6. Send a third Get Log Page command to LID 06h (Device Self-test). Repeat until the current Device Self-Test operation field is set to 0 in the Log Page that is returned.

Observable Results:

1. Using the Log Page info returned in response to the second Get Log Page command, which followed the initial Device Self-test command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 1h when the operation was processing.
2. Verify that the completion for the initial Device Self-test command indicates status 'Success' (00h).
3. Verify that the completion for the second attempted Device Self-test command indicates status 'Device Self-test In Progress' (1Dh).
4. Using the Log Page info returned in response to the third or later Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 0h when the operation was completed.
5. Verify that all received responses have all Reserved fields set to 0h.

Case 7: Namespace Test Action = 00000001h-FFFFFFFEh, STC=2h and 2h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host to send an initial Device Self-test command with Namespace Test Action set to a value between 00000001h and FFFFFFFEh and STC set to 2h (Extended device self-test operation). Wait for a Completion to be received with status "Success".
4. Determine that the initial Device Self-test command is in progress by sending a second Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 2 (Extended device self-test operation in progress). If the Current Device Self-Test Operation field is set to 0, the Device Self-Test Operation is competing too quickly for the test to be performed and this test is not applicable.
5. Before the first Device Self-test operation completes, send a second Device Self-test command with Namespace Test Action set to a value between 00000001h and FFFFFFFEh (use the same value as in the previous Device Self Test command) and STC set to 2h (Extended device self-test operation). Wait for a Completion to be received.

© 2026 UNH–IOL

6. Send a third Get Log Page command to LID 06h (Device Self-test). Repeat until the current Device Self-Test operation field is set to 0h in the Log Page that is returned.

Observable Results:

1. Using the Log Page info returned in response to the second Get Log Page command, which followed the initial Device Self-test command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 1h when the operation was processing.
2. Verify that the completion for the initial Device Self-test command indicates status 'Success' (00h).
3. Verify that the completion for the second attempted Device Self-test command indicates status 'Device Self-test In Progress' (1Dh).
4. Using the Log Page info returned in response to the third or later Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 0h when the operation was completed.
5. Verify that all received responses have all Reserved fields set to 0h.

Case 8: Namespace Test Action = FFFFFFFFh, STC=2h, Second DST (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host to send an initial Device Self-test command with Namespace Test Action set to FFFFFFFFh and STC set to 2h (Extended device self-test operation). Wait for a Completion to be received with status "Success".
4. Determine that the initial Device Self-test command is in progress by sending a second Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 2 (Extended device self-test operation in progress). If the Current Device Self-Test Operation field is set to 0, the Device Self-Test Operation is competing too quickly for the test to be performed and this test is not applicable.
5. Before the first Device Self-test operation completes, send a second Device Self-test command with Namespace Test Action set to FFFFFFFFh and STC set to 2h (Extended device self-test operation). Wait for a Completion to be received.
6. Send a third Get Log Page command to LID 06h (Device Self-test). Repeat until the current Device Self-Test operation field is set to 0h in the Log Page that is returned.

Observable Results:

1. Using the Log Page info returned in response to the second Get Log Page command, which followed the initial Device Self-test command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 1h when the operation was processing.
2. Verify that the completion for the initial Device Self-test command indicates status 'Success'.
3. Verify that the completion for the second attempted Device Self-test command indicates status 'Device Self-test In Progress' (1Dh).
4. Using the Log Page info returned in response to the third or later Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 0h when the operation was completed.
5. Verify that all received responses have all Reserved fields set to 0h.

Possible Problems: None known.

© 2026 UNH–IOL

Test 1.12 – Abort Device Self-test Short Operation (OF-FYI, M)

Purpose: To verify that an NVMe Controller can properly perform an abort on a Device Self-test Short Operation.

References:

Old Ref: [1] NVMe Specification 5.8, 8.11, TP 4022
NVM Express Base specification 2.0a : 5.9, 8.6

Resource Requirements:

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

Last Modification: May 4, 2020

Discussion: The Device Self-test command is used to start a device self-test operation or abort a device self-test

Operation. The device self-test operation is performed by the controller that the Device Self-test command was submitted to.

Test Setup: See Appendix A.

Case 1: Namespace Test Action = 00000000h, STC=1h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host send a Get Log Page command to LID 06h (Device Self-test).
4. Configure the NVMe Host to send a Device Self-test command with Namespace Test Action set to 00000000h and STC set to 1h (Short device self-test operation).
5. Before the Device Self-test operation completes in the background, send a Device Self-test command with Namespace Test Action set to 00000000h and STC set to Fh (Abort device self-test), to the same controller as in the previous step.
6. Once a command completion is received for the Device Self-test command, send a Get Log Page command to LID 06h (Device Self-test). Repeat until the log Page indicates that the DST operation is no longer in progress.

Observable Results:

1. Verify that the first Device Self-test command, with STC=1h, completes with status 1h, Aborted by a Device Self Test Command.
2. Verify that the second Device Self-test command, with STC=Fh, completes successfully.
3. Compare the Device Self Test Log received in steps 3 and 6 and verify that that the second received Device Self Test Log contains a new log entry in the Newest Self-test Results Data Structure in the Device Self-test Log.
4. Verify that the second received Device Self-test Log has the Current Device Self-test Status field set to 0h.
5. Verify that all received responses have all Reserved fields set to 0.

Case 2: Test Action = 00000001h-FFFFFFFEh, STC=1h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.

© 2026 UNH–IOL

2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host send a Get Log Page command to LID 06h (Device Self-test).
4. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to a value between 00000001h and FFFFFFFEh that represent a valid Namespace on the DUT, with STC set to 1h (Short device self-test operation).
5. Before the Device Self-test operation completes in the background, send a Device Self-test command with Namespace Test Action set the same value as the previous step, and STC set to Fh (Abort device self-test), to the same controller as in the previous test.
6. Once a command completion is received for the Device Self-test command, send a Get Log Page command to LID 06h (Device Self-test). Repeat until the log Page indicates that the DST operation is no longer in progress.

Observable Results:

1. Verify that the first Device Self-test command, with STC=1h, completes with status 1h, Aborted by a Device Self Test Command.
2. Verify that the second Device Self-test command, with STC=Fh, completes successfully.
3. Compare the Device Self Test Log received in steps 3 and 6 and verify that that the second received Device Self Test Log contains a new log entry in the Newest Self-test Results Data Structure in the Device Self-test Log.
4. Verify that the second received Device Self-test Log has the Current Device Self-test Status field set to 0h.
5. Verify that all received responses have all Reserved fields set to 0.

Case 3: Test Action = FFFFFFFFh, STC=1h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host send a Get Log Page command to LID 06h (Device Self-test).
4. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to a value of FFFFFFFFh, with STC set to 1h (Short device self-test operation).
5. Before the Device Self-test operation completes in the background, send a Device Self-test command with Namespace Test Action set the same value as the previous step, and STC set to Fh (Abort device self-test), to the same controller as in the previous test.
6. Once a command completion is received for the Device Self-test command, send a Get Log Page command to LID 06h (Device Self-test). Repeat until the log Page indicates that the DST operation is no longer in progress.

Observable Results:

1. Verify that the first Device Self-test command, with STC=1h, completes with status 1h, Aborted by a Device Self Test Command.
2. Verify that the second Device Self-test command, with STC=Fh, completes successfully.
3. Compare the Device Self Test Log received in steps 3 and 6 and verify that that the second received Device Self Test Log contains a new log entry in the Newest Self-test Results Data Structure in the Device Self-test Log.
4. Verify that the second received Device Self-test Log has the Current Device Self-test Status field set to 0h.
5. Verify that all received responses have all Reserved fields set to 0.

© 2026 UNH–IOL

Case 4: DST with specific NSID Aborted by Format NVM Command with specific NSID (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host send a Get Log Page command to LID 06h (Device Self-test).
4. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to a valid Namespace ID value other than NSID=FFFFFFFFh, with STC set to 1h (Short device self-test operation).
5. Before the Device Self-test operation completes in the background, send a Format NVM command with the same Namespace ID as in the previous step.
6. Once a command completion is received for the Device Self-test command, send a Get Log Page command to LID 06h (Device Self-test). Repeat until the log Page indicates that the DST operation is no longer in progress.

Observable Results:

1. Verify that the first Device Self-test command, with STC=1h, completes with status 4h aborted by a Format NVM command.
2. Verify that the Format NVM command completes successfully.
3. Compare the Device Self Test Log received in steps 3 and 6 and verify that that the second received Device Self Test Log contains a new log entry in the Newest Self-test Results Data Structure in the Device Self-test Log.
4. Verify that all received responses have all Reserved fields set to 0.

Case 5: Aborted by Controller Level Reset (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host send a Get Log Page command to LID 06h (Device Self-test).
4. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to a valid Namespace ID value, with STC set to 1h (Short device self-test operation).
5. Configure the NVMe Host to perform a Controller Level Reset.
6. Once a command completion is received for the Device Self-test command, send a Get Log Page command to LID 06h (Device Self-test). Repeat until the log Page indicates that the DST operation is no longer in progress.

Observable Results:

1. Verify that the first Device Self-test command, with STC=1h, completes with status 2h aborted by a Controller Level Reset.
2. Verify that the Controller Level Reset completes successfully.
3. Compare the Device Self Test Log received in steps 3 and 6 and verify that that the second received Device Self Test Log contains a new log entry in the Newest Self-test Results Data Structure in the Device Self-test Log.
4. Verify that the second received Device Self-test Log has the Current Device Self-test Status field set to 0h.
5. Verify that all received responses have all Reserved fields set to 0.

© 2026 UNH–IOL

Case 6: Aborted by Sanitize Operation (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Check SANICAP field in the Identify Controller Data Structure. If SANICAP is set to 0, this test is not applicable.
3. Check the VER field. This test is only applicable if the DUT claims to support NVMe v1.4 or higher.
4. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
5. Configure the NVMe Host send a Get Log Page command to LID 06h (Device Self-test).
6. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to a valid Namespace ID value, with STC set to 1h (Short device self-test operation).
7. Configure the Testing Station to issue a Sanitize Command for a supported operation type as indicated as supported in the SANICAP field: Overwrite, Block Erase, and Crypto Erase. Repeat entire test procedure for each support operation type.
8. Continually issue a Get Log Page for Sanitize State log until SSTAT is set to 001b.
9. Verify that the Sanitize operation completed by performing a Get Log on the Sanitize Status log page Log Identifier 81h and check the Sanitize progress (SPROG) field.
10. Once a command completion is received for the Device Self-test command, send a Get Log Page command to LID 06h (Device Self-test). Repeat until the log Page indicates that the DST operation is no longer in progress.

Observable Results:

1. Verify that the first Device Self-test command, with STC=1h, completes with status ‘Aborted due to Sanitize’ (09h) indicated in the Self-test Result Data Structure returned in response to the Get Log Page request.

Case 7: DST with specific NSID Aborted by Format NVM Command with NSID = FFFFFFFFh (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host send a Get Log Page command to LID 06h (Device Self-test).
4. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to a valid Namespace ID value other than NSID=FFFFFFFFh, with STC set to 1h (Short device self-test operation).
5. Before the Device Self-test operation completes in the background, send a Format NVM command with NSID=FFFFFFFFh.
6. Once a command completion is received for the Device Self-test command, send a Get Log Page command to LID 06h (Device Self-test). Repeat until the log Page indicates that the DST operation is no longer in progress.

Observable Results:

1. Verify that the first Device Self-test command, with STC=1h, completes with status 4h aborted by Format NVM command.
2. Verify that the Format NVM command completes successfully.
3. Compare the Device Self Test Log received in steps 3 and 6 and verify that that the second received Device Self Test Log contains a new log entry in the Newest Self-test Results Data Structure in the Device Self-test Log.

© 2026 UNH–IOL

4. Verify that all received responses have all Reserved fields set to 0.

Case 8: DST NSID=FFFFFFFFh Aborted by Format NVM Command with NSID = FFFFFFFFh (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host send a Get Log Page command to LID 06h (Device Self-test).
4. Configure the NVMe Host to send Device Self-test commands with NSID=FFFFFFFFh, with STC set to 1h (Short device self-test operation).
5. Before the Device Self-test operation completes in the background, send a Format NVM command with NSID=FFFFFFFFh.
6. Once a command completion is received for the Device Self-test command, send a Get Log Page command to LID 06h (Device Self-test). Repeat until the log Page indicates that the DST operation is no longer in progress.

Observable Results:

1. Verify that the first Device Self-test command, with STC=1h, completes with status 4h aborted by Format NVM command.
2. Verify that the Format NVM command completes successfully.
3. Compare the Device Self Test Log received in steps 3 and 6 and verify that that the second received Device Self Test Log contains a new log entry in the Newest Self-test Results Data Structure in the Device Self-test Log.

Possible Problems: None known.

© 2026 UNH–IOL

Test 1.13 – Abort Device Self-test Extended Operation (OF-FYI, M)

Purpose: To verify that an NVMe Controller can properly perform an abort on a Device Self-test Extended Operation.

References:

Old Ref: [1] NVMe Specification 5.8, 8.11, TP 4022
NVM Express Base specification 2.0a : 5.9, 8.6

Resource Requirements:

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

Last Modification: May 4, 2020

Discussion: The Device Self-test command is used to start a device self-test operation or abort a device self-test

Operation. The device self-test operation is performed by the controller that the Device Self-test command was submitted to.

Test Setup: See Appendix A.

Case 1: Namespace Test Action = 00000000h, STC=2h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host send a Get Log Page command to LID 06h (Device Self-test).
4. Configure the NVMe Host to send a Device Self-test command with Namespace Test Action set to 00000000h and STC set to 2h (Extended device self-test operation).
5. Before the Device Self-test operation completes in the background, send a Device Self-test command with Namespace Test Action set to 00000000h and STC set to Fh (Abort device self-test), to the same controller as in the previous test.
6. Once a command completion is received for the Device Self-test command, send a Get Log Page command to LID 06h (Device Self-test). Repeat until the log Page indicates that the DST operation is no longer in progress.

Observable Results:

1. Verify that the first Device Self-test command, with STC=2h, completes with status 1h Aborted by a Device Self Test command.
2. Verify that the second Device Self-test command, with STC=Fh, completes successfully.
3. Compare the Device Self Test Log received in steps 3 and 6 and verify that that the second received Device Self Test Log contains a new log entry in the Newest Self-test Results Data Structure in the Device Self-test Log.
4. Verify that the second received Device Self-test Log has the Current Device Self-test Status field set to 0h.
5. Verify that all received responses have all Reserved fields set to 0.

Case 2: Test Action = 00000001h-FFFFFFFEh, STC=2h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.

© 2026 UNH–IOL

2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host send a Get Log Page command to LID 06h (Device Self-test).
4. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to a value between 00000001h and FFFFFFFEh that represent a valid Namespace on the DUT, with STC set to 2h (Extended device self-test operation).
5. Before the Device Self-test operation completes in the background, send a Device Self-test command with Namespace Test Action set the same value as the previous step, and STC set to Fh (Abort device self-test), to the same controller as in the previous test.
6. Once a command completion is received for the Device Self-test command, send a Get Log Page command to LID 06h (Device Self-test). Repeat until the log Page indicates that the DST operation is no longer in progress.

Observable Results:

1. Verify that the first Device Self-test command, with STC=2h, completes with status 1h, Aborted by a Device Self Test Command.
2. Verify that the second Device Self-test command, with STC=Fh, completes successfully.
3. Compare the Device Self Test Log received in steps 3 and 6 and verify that that the second received Device Self Test Log contains a new log entry in the Newest Self-test Results Data Structure in the Device Self-test Log.
4. Verify that the second received Device Self-test Log has the Current Device Self-test Status field set to 0h.
5. Verify that all received responses have all Reserved fields set to 0.

Case 3: Test Action = FFFFFFFFh, STC=2h (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host send a Get Log Page command to LID 06h (Device Self-test).
4. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to a value of FFFFFFFFh, with STC set to 2h (Extended device self-test operation).
5. Before the Device Self-test operation completes in the background, send a Device Self-test command with Namespace Test Action set the same value as the previous step, and STC set to Fh (Abort device self-test), to the same controller as in the previous test.
6. Once a command completion is received for the Device Self-test command, send a Get Log Page command to LID 06h (Device Self-test). Repeat until the log Page indicates that the DST operation is no longer in progress.

Observable Results:

1. Verify that the first Device Self-test command, with STC=2h, completes with status 1h, Aborted by a Device Self Test Command.
2. Verify that the second Device Self-test command, with STC=Fh, completes successfully.
3. Compare the Device Self Test Log received in steps 3 and 6 and verify that that the second received Device Self Test Log contains a new log entry in the Newest Self-test Results Data Structure in the Device Self-test Log.
4. Verify that the second received Device Self-test Log has the Current Device Self-test Status field set to 0h.
5. Verify that all received responses have all Reserved fields set to 0.

© 2026 UNH–IOL

Case 4: DST with specific NSID Aborted by Format NVM Command with specific NSID (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host send a Get Log Page command to LID 06h (Device Self-test).
4. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to a valid Namespace ID value other than FFFFFFFFh, with STC set to 2h (Extended device self-test operation).
5. Before the Device Self-test operation completes in the background, send a Format NVM command with the name Namespace ID as in the previous step.
6. Once a command completion is received for the Device Self-test command, send a Get Log Page command to LID 06h (Device Self-test). Repeat until the log Page indicates that the DST operation is no longer in progress.

Observable Results:

1. Verify that the first Device Self-test command, with STC=2h, completes with status 4h, to indicate that the Device Self Test Operation was aborted by a Format NVM command.
2. Verify that the Format NVM command completes successfully.
3. Compare the Device Self Test Log received in steps 3 and 6 and verify that that the second received Device Self Test Log contains a new log entry in the Newest Self-test Results Data Structure in the Device Self-test Log.
4. Verify that all received responses have all Reserved fields set to 0.

Case 5: Aborted by Sanitize Operation (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Check SANICAP field in the Identify Controller Data Structure. If SANICAP is set to 0, this test is not applicable.
3. Check the VER field. This test is only applicable if the DUT claims to support NVMe v1.4 or higher.
4. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
5. Configure the NVMe Host send a Get Log Page command to LID 06h (Device Self-test).
6. Configure the NVMe Host to send Device Self-test commands with Namespace Test Action set to a valid Namespace ID value, with STC set to 2h (Extended device self-test operation).
7. Configure the Testing Station to issue a Sanitize Command for a supported operation type as indicated as supported in the SANICAP field: Overwrite, Block Erase, and Crypto Erase. Repeat entire test procedure for each support operation type.
8. Once a command completion is received for the Device Self-test command, send a Get Log Page command to LID 06h (Device Self-test). Repeat until the log Page indicates that the DST operation is no longer in progress.

Observable Results:

1. Verify that the first Device Self-test command, with STC=2h, completes with status ‘Aborted due to Sanitize’ (9h) indicated in the Self-test Result Data Structure returned in response to the Get Log Page request.

© 2026 UNH–IOL

Case 6: DST with specific NSID Aborted by Format NVM Command with NSID = FFFFFFFFh (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host send a Get Log Page command to LID 06h (Device Self-test).
4. Configure the NVMe Host to send Device Self-test Extended Operation commands with Namespace Test Action set to a valid Namespace ID value other than NSID=FFFFFFFFh, with STC set to 2h (Extended device self-test operation).
5. Before the Device Self-test operation completes in the background, send a Format NVM command with NSID=FFFFFFFFh.
6. Once a command completion is received for the Device Self-test command, send a Get Log Page command to LID 06h (Device Self-test). Repeat until the log Page indicates that the DST operation is no longer in progress.

Observable Results:

1. Verify that the first Device Self-test Extended Operation command, with STC=2h, completes with status 4h, to indicate that the Device Self Test Operation was aborted by a Format NVM command.
2. Verify that the Format NVM command completes successfully.
3. Compare the Device Self Test Log received in steps 3 and 6 and verify that that the second received Device Self Test Log contains a new log entry in the Newest Self-test Results Data Structure in the Device Self-test Log.
4. Verify that all received responses have all Reserved fields set to 0.

Case 7: DST NSID=FFFFFFFFh Aborted by Format NVM Command with NSID = FFFFFFFFh (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If Bit 4 is not set to 1, this test is not applicable.
2. Ensure that no Device Self-test operation is currently in progress by sending Get Log Page with LID=06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0. If the Current Device Self-Test Operation field is not set to 0, wait for the Device Self-Test Operation to complete, and send the Get Log Page with LID=06h again. Repeat until the Current Device Self-Test Operation field is 0.
3. Configure the NVMe Host send a Get Log Page command to LID 06h (Device Self-test).
4. Configure the NVMe Host to send Device Self-test commands with NSID=FFFFFFFFh, with STC set to 2h (Extended device self-test operation).
5. Before the Device Self-test operation completes in the background, send a Format NVM command with NSID=FFFFFFFFh.
6. Once a command completion is received for the Device Self-test command, send a Get Log Page command to LID 06h (Device Self-test). Repeat until the log Page indicates that the DST operation is no longer in progress.

Observable Results:

1. Verify that the first Device Self-test Extended Operation command, with STC=2h, completes with status 4h, to indicate that the Device Self Test Operation was aborted by a Format NVM command.
2. Verify that the Format NVM command completes successfully.
3. Compare the Device Self Test Log received in steps 3 and 6 and verify that that the second received Device Self Test Log contains a new log entry in the Newest Self-test Results Data Structure in the Device Self-test Log.

© 2026 UNH–IOL

Possible Problems: None known.

© 2026 UNH–IOL

Test 1.14 – NVMe-MI Send/Receive (OF-FYI, M)

Purpose: To verify that an NVMe Controller can properly perform an NVMe-MI Send and Receive operation.

References:

Old Ref: [1] NVMe Specification 5.17, 5.18
Old Ref : [2] NVMe-MI Specification 5.5
NVM Express Base specification 2.0a : 5.20, 5.21
NVMe-MI Specification 1.0 : 6.6

Resource Requirements:

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

Last Modification: February 5, 2018

Discussion: The NVMe-MI Send command is used to transfer an NVMe-MI Request Message to the controller.

NVMe-MI Receive command transfers an NVMe-MI Response Message from the controller to the host that corresponds to an NVMe-MI Request Message that was previously submitted to the controller.

Refer to the NVM Express Management Interface (NVMe-MI) specification for the format and servicing of the NVMe-MI Request and Response Messages.

Test Setup: See Appendix A.

Case 1: Request and Response (OF-FYI, M)

Test Procedure:

1. Check Bit 6 of the OACS field in the Identify Controller Data Structure. If Bit 6 is not set to 1, this test is not applicable.
2. Configure the Testing Station to issue an NVMe-MI Send Command containing an Opcode of VPD Write (06h) and an empty data buffer.
3. Configure the Testing Station to issue an NVMe-MI Receive Command with the Opcode of NVM Subsystem Health Status Poll (01h).

Observable Results:

1. Verify that for both the NVMe-MI Send and Receive commands that a command completion queue entry is posted to the Admin Completion Queue with a status code of success, or Invalid Opcode. TP6028 changed VPD Write support to Optional.

Possible Problems: None known.

© 2026 UNH–IOL

Test 1.15 – Directive Receive Identify (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller can properly perform a Directives Receive operation.

References:

Old Ref: [1] NVMe Specification 5.9, 5.10, 9
NVM Express Base specification 2.0a : 5.10, 5.11, 8.7

Resource Requirements:

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

Last Modification: November 26, 2018

Discussion: Directives is a mechanism to enable host and NVM subsystem or controller information exchange. The Directive Receive command is used to transfer data related to a specific Directive Type from the controller to the host. The Directive Send command is used to transfer data related to a specific Directive Type from the host to the controller.

Test Setup: See Appendix A.

Case 1: Valid Receive (OF-FYI, FYI)

Test Procedure:

1. Check Bit 5 of the OACS field in the Identify Controller Data Structure. If Bit 5 is not set to 1, this test is not applicable.
2. Configure the Testing Station to issue a Directive Receive Command with Directive Type set to Identify (00h) and the Directive Operation Value set to Return parameters (01h), and a valid NSID value.

Observable Results:

1. Verify that the controller sent a Return Parameters Data Structure indicating that the identify Directive is support and enabled.

Case 2: Receive with NSID=FFFFFFFFh (OF-FYI, FYI)

Test Procedure:

1. Check Bit 5 of the OACS field in the Identify Controller Data Structure. If Bit 5 is not set to 1, this test is not applicable.
2. Configure the Testing Station to issue a Directive Receive Command with Directive Type set to Identify (00h) and the Directive Operation Value set to Return parameters (01h), and a NSID value of FFFFFFFFh.

Observable Results:

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

Possible Problems: None known.

© 2026 UNH–IOL

Test 1.16 – Directive Send Enable Directive (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller can properly perform a Directives Send operation.

References:

Old Ref: [1] NVMe Specification 5.9, 5.10, 9
NVM Express Base specification 2.0a : 5.10, 5.11, 8.7

Resource Requirements:

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

Last Modification: April 2, 2019

Discussion: Directives is a mechanism to enable host and NVM subsystem or controller information exchange. The Directive Receive command is used to transfer data related to a specific Directive Type from the controller to the host. The Directive Send command is used to transfer data related to a specific Directive Type from the host to the controller.

Test Setup: See Appendix A.

Case 1: Valid Send (OF-FYI, FYI)

Test Procedure:

1. Check Bit 5 of the OACS field in the Identify Controller Data Structure. If Bit 5 is not set to 1, this test is not applicable.
2. Configure the Testing Station to issue a Directive Receive Command with Directive Type set to Identify (00h) and the Directive Operation Value set to Return parameters (01h), and a valid NSID value.
3. Check the Return Parameters Data Structure if the Streams Directive is supported. If the Streams Directive is not supported this test is not applicable.
4. If the Streams Directive is supported, and not enabled, the Testing Station should perform a Directive Send with operation Enable Directive (01h) and DTYPE set to Streams (01h), and ENDIR set to 1.
5. If the Streams Directive is supported, and enabled, the Testing Station should perform a Directive Send with operation Enable Directive (01h) and DTYPE set to Streams (01h), and ENDIR set to 0. Next, the Testing Station should perform a Directive Send with operation Enable Directive (01h) and DTYPE set to Streams (01h), and ENDIR set to 1.

Observable Results:

1. Verify that the controller sent a Return Parameters Data Structure indicating that the identify Directive is support and enabled.
2. Verify that each of the Directives Send commands completes with status “Success”.

Case 2: Send to Enable Identify (OF-FYI, FYI)

Test Procedure:

1. Check Bit 5 of the OACS field in the Identify Controller Data Structure. If Bit 5 is not set to 1, this test is not applicable.
2. Configure the Testing Station to issue a Directive Receive Command with Directive Type set to Identify (00h) and the Directive Operation Value set to Return parameters (01h), and a valid NSID value.
3. Check the Return Parameters Data Structure if the Streams Directive is supported. If the Streams Directive is not supported this test is not applicable.
4. The Testing Station should perform a Directive Send with operation Enable Directive (01h) and DTYPE set to Identify (00h), and ENDIR set to 1.

Observable Results:

© 2026 UNH–IOL

1. Verify that the controller sent a Return Parameters Data Structure indicating that the identify Directive is support and enabled.
2. Verify that each of the Directives Send commands completes with status “Success”.

Case 3: Send to Enable Unsupported Directive (OF-FYI, FYI)

Test Procedure:

1. Check Bit 5 of the OACS field in the Identify Controller Data Structure. If Bit 5 is not set to 1, this test is not applicable.
2. Configure the Testing Station to issue a Directive Receive Command with Directive Type set to Identify (00h) and the Directive Operation Value set to Return parameters (01h), and a valid NSID value.
3. Check the Return Parameters Data Structure if the Streams Directive is supported. If the Streams Directive is supported this test is not applicable.
4. The Testing Station should perform a Directive Send with operation Enable Directive (01h) and DTYPE set to Streams (01h), and ENDIR set to 1.

Observable Results:

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

Case 4: Shared Stream Writes (OF-FYI, FYI)

Test Procedure:

1. Check Bit 5 of the OACS field in the Identify Controller Data Structure. If Bit 5 is not set to 1h, this test is not applicable.
2. Configure the Testing Station to issue a Directive Receive Command with Directive Type set to Identify (00h) and the Directive Operation Value set to Return parameters (01h), and a valid NSID value.
3. Check the Return Parameters Data Structure. If the Streams Directive is supported this test is not applicable.
4. The Testing Station should perform a Directive Send with operation Enable Directive (01h) and DTYPE set to Streams (01h), and ENDIR set to 1h.
5. Issue an Allocate Resources operation to allocate stream resources to the namespace from 2 different Host IDs.
6. Issue multiple write commands to the namespace from 2 different Host IDs using the same stream identifier (Shared Stream Writes). Each Host will write a unique data pattern.
7. Perform a READ operation to the LBAs written in the previous step.

Observable Results:

1. Verify that the Shared Stream Write operations completed successfully and that the expected data pattern was read back.

Case 5: Directive Send Release Resources updates NSSA (OF-FYI, FYI)

Test Procedure:

1. Check Bit 5 of the OACS field in the Identify Controller Data Structure. If Bit 5 is not set to 1, this test is not applicable.
2. Configure the Testing Station to issue a Directive Receive Command with Directive Type set to Identify (00h) and the Directive Operation Value set to Return parameters (01h), and a valid NSID value.
3. Check the Return Parameters Data Structure. If the Streams Directive is not supported this test is not applicable.
4. The Testing Station should perform a Directive Send with operation Enable Directive (01h) and DTYPE set to Streams (01h), and ENDIR set to 1.
5. Perform a Direct Receive Operation of Return Parameters, records the NSSA value.
6. Issue an Allocate Resources operation (03h) to allocate stream resources to the namespace from a single Host IDs with an NSR value of 1.
7. Perform a Direct Receive Operation of Return Parameters, records the NSSA value.
8. Issue an Release Resources operation (02h) to release stream resources from the namespace.
9. Perform a Direct Receive Operation of Return Parameters, records the NSSA value.

© 2026 UNH–IOL

Observable Results:
1. Verify that the NSSA value reported in Step 7 was 1 less than the NSSA value reported in Step 5.
2. Verify that the NSSA value reported in Step 9 was equal to the NSSA value reported in Step 5.

Case 6: Namespace Deletion updates NSSA (OF-FYI, FYI)

Test Procedure:
1. Check Bit 5 of the OACS field in the Identify Controller Data Structure. If Bit 5 is not set to 1, this test is not applicable.
2. Configure the Testing Station to issue a Directive Receive Command with Directive Type set to Identify (00h) and the Directive Operation Value set to Return parameters (01h), and a valid NSID value.
3. Check the Return Parameters Data Structure. If the Streams Directive is not supported this test is not applicable.
4. The Testing Station should perform a Directive Send with operation Enable Directive (01h) and DTYPE set to Streams (01h), and ENDIR set to 1.
5. Perform a Direct Receive Operation of Return Parameters, records the NSSA value.
6. Issue an Allocate Resources operation (03h) to allocate stream resources to the namespace from a single Host IDs with an NSR value of 1.
7. Perform a Direct Receive Operation of Return Parameters, record the NSSA value.
8. Delete the namespace resources were allocated to.
9. Perform a Direct Receive Operation of Return Parameters, record the NSSA value.
10. Recreate the namespace that was deleted in step 8

Observable Results:
1. Verify that the NSSA value reported in Step 7 was 1 less than the NSSA value reported in Step 5.
2. Verify that the NSSA value reported in Step 9 was equal to the NSSA value reported in Step 5.

Case 7: Format NVM to Namespace updates NSSA (OF-FYI, FYI)

Test Procedure:
1. Check Bit 5 of the OACS field in the Identify Controller Data Structure. If Bit 5 is not set to 1, this test is not applicable.
2. Configure the Testing Station to issue a Directive Receive Command with Directive Type set to Identify (00h) and the Directive Operation Value set to Return parameters (01h), and a valid NSID value.
3. Check the Return Parameters Data Structure. If the Streams Directive is supported. If the Streams Directive is supported this test is not applicable.
4. The Testing Station should perform a Directive Send with operation Enable Directive (01h) and DTYPE set to Streams (01h), and ENDIR set to 1.
5. Perform a Direct Receive Operation of Return Parameters, records the NSSA value.
6. Issue an Allocate Resources operation (03h) to allocate stream resources to the namespace from a single Host IDs with an NSR value of 1.
7. Perform a Direct Receive Operation of Return Parameters, records the NSSA value.
8. Perform a Format NVM command on the namespace resources were allocated to.
9. Perform a Direct Receive Operation of Return Parameters, records the NSSA value.

Observable Results:
1. Verify that the NSSA value reported in Step 7 was 1 less than the NSSA value reported in Step 5.
2. Verify that the NSSA value reported in Step 9 was equal to the NSSA value reported in Step 5.

Possible Problems: None known.

Case 8: Send Vendor Specific Directive, not supported (OF-FYI, FYI)

Test Procedure:

© 2026 UNH–IOL

1. Configure the Testing Station to issue an Identify Controller Command and check the value of the OACS bit 5. If the DUT does not support Directives, this test is not applicable.
2. Configure the Testing Station to issue a Directive Receive Command with a Directive Type set to Identify (00h) and the Directive Operation Value set to Return Parameters (01h), and a valid NSID value.
3. Check the Return Parameters Data Structure to determine if the Vendor Specific Directive is supported. If the Vendor Specific Directive is supported this test is not applicable.
4. Configure the Testing Station to perform a Directive Send with operation Enable Directive (01h) and DTYPE set to Vendor Specific (0Fh), and ENDIR set to 1.

Observable Results:

1. Verify that the command in step 4 completes with a status of Invalid Field in Command.
2. Verify that all other commands complete successfully.

Possible Problems: None known.

© 2026 UNH–IOL

Test 1.17 – Sanitize Command (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller can properly perform a Sanitize operation.

References:

Old Ref: [1] NVMe Specification 5.24, 8.15, NVMe v1.3 ECN 005
NVM Express Base specification 2.0a : 5.24, 8.21

Resource Requirements:

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

Last Modification: December 1, 2019

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

Test Setup: See Appendix A.

Case 1: Sanitize Supported (OF-FYI, M)

Test Procedure:

1. Check SANICAP field in the Identify Controller Data Structure. If SANICAP is set to 0, this test is not applicable.
2. Configure the Testing Station to issue a Sanitize Command, for each of the operation types indicated as supported in the SANICAP field: Overwrite, Block Erase, and Crypto Erase. Each Sanitize command should be followed by a Get Log Page command for the Sanitize Status Log Page.

Observable Results:

1. Verify that each of the supported Sanitize operation commands completes with status ‘Success’, and each of the unsupported Sanitize operation commands completes with status ‘Invalid Field in Command’ (02h).
2. Verify that the Sanitize Status Log Page is updated after each Sanitize operation that completes with status ‘Success’.

Case 2: Sanitize Not Supported (OF-FYI, M)

Test Procedure:

1. Check SANICAP field in the Identify Controller Data Structure.
2. Configure the Testing Station to issue a Sanitize Command, repeat for each operation: Overwrite, Block Erase, and Crypto Erase.

Observable Results:

1. Verify that each of the unsupported Sanitize operations completes with status “Invalid Field in Command” and that the corresponding bit in the SANICAP field is set to 0.

Case 3: Sanitize Config FID Not Savable (OF-FYI, FYI)

Test Procedure:

1. Check SANICAP field in the Identify Controller Data Structure. If SANICAP is set to 0, this test is not applicable.
2. Configure the Testing Station to issue a Get Feature Command for FID 17h Sanitize Config with SEL=011b (Supported Capabilities).

© 2026 UNH–IOL

3. Configure the Testing Station to issue a Get Feature Command for FID 17h Sanitize Config with SEL=001b (Default Value)

Observable Results:

1. Verify that if Dword 0 bit 0 of the completion entry of the Get Features command with SEL=011b (i.e., Supported Capabilities) for this Feature Identifier is cleared to '0', then the default value of the NODRM attribute shall be cleared to '0'.

Case 4: Sanitize Config NDI=1, NODRM=1 (OF-FYI, M)

Test Procedure:

1. Check SANICAP field in the Identify Controller Data Structure. If the No Deallocate Inhibited bit (NDI)is set to 0, this test is not applicable.
2. Configure the Testing Station to issue a Set Feature Command for FID 17h Sanitize Config to set NODRM =1.
3. Configure the Testing Station to issue a Get Feature Command for FID 17h Sanitize Config to verify that NODRM =1.
4. Perform a Sanitize Command with the No Deallocate After Sanitize Bit set to 1.
5. Perform a Get Log Page for the Sanitize Status Log Page, LID=81h.

Observable Results:

1. Verify that the Sanitize operation is completed successfully.
2. Verify that bits 2:0 of the Sanitize Status field in the Sanitize Status log page are set to 100b.

Case 5: Sanitize Config NDI=1, NODRM=0 (OF-FYI, M)

Test Procedure:

1. Check SANICAP field in the Identify Controller Data Structure. If the No Deallocate Inhibited bit (NDI)is set to 0h, this test is not applicable.
2. Configure the Testing Station to issue a Set Feature Command for FID 17h Sanitize Config to set NODRM = 0.
3. Configure the Testing Station to issue a Get Feature Command for FID 17h Sanitize Config to verify that NODRM = 0.
4. Perform a Sanitize Command with the No Deallocate After Sanitize Bit set to 1h.
5. Perform a Get Log Page for the Sanitize Status Log Page, LID=81h.

Observable Results:

1. Verify that the Sanitize operation is aborted with status 'Invalid Field in Command' (02h).

Case 6: Sanitize In Progress SPROG (OF-FYI, M)

Test Procedure:

1. Check SANICAP field in the Identify Controller Data Structure. If the SANICAP field set to 0, this test is not applicable.
2. Perform a Sanitize Command, before command completion arrives for the Sanitize Command, issue a Get Log Page for the Sanitize Status Log Page, LID=81h.

Observable Results:

1. Verify that the Sanitize Status Log Page has SSTAT set to 010b, and SPROG is not equal to FFFFh.

Case 7: Sanitize Not In Progress SPROG (OF-FYI, M)

Test Procedure:

1. Check SANICAP field in the Identify Controller Data Structure. If the SANICAP field set to 0, this test is not applicable.

© 2026 UNH–IOL

2. Perform a Sanitize Command, wait for command completion.
3. Perform a Get Log Page for the Sanitize Status Log Page, LID=81h.

Observable Results:

1. Verify that the Sanitize Status Log Page has SSTAT set to 001b, and SPROG=FFFFh.

Case 8: Sanitize Completes Succesfully, then Sanitize with Action = 001b (OF-FYI, M)

Test Procedure:

1. Check SANICAP field in the Identify Controller Data Structure. If the SANICAP field set to 0h, this test is not applicable.
2. Perform a Sanitize Command, wait for command completion. Verify that the Sanitize operation is successful.
3. Perform a Get Log Page for the Sanitize Status Log Page, LID=81h.
4. Perform a Sanitize Command with Sanitize Action set to 001b (Exit Failure Mode), wait for command completion.

Observable Results:

1. Verify that the 2nd Sanitize command completes with status 'Success'.

Case 9: Prohibited Commands Aborted when Sanitize In Progress SPROG (OF-FYI, M)

Test Procedure:

1. Check SANICAP field in the Identify Controller Data Structure. If the SANICAP field set to 0, this test is not applicable.
2. Perform a Sanitize Command, before command completion arrives for the Sanitize Command, issue the following commands:
    a. Compare Command (if supported)
    b. Dataset Management Command (if supported)
    c. Read Command
    d. Reservation Acquire Command (if supported)
    e. Reservation Register Command (if supported)
    f. Reservation Release Command (if supported)
    g. Reservation Report Command (if supported)
    h. Verify Command (if supported)
    i. Write Command
    j. Write Uncorrectable Command (if supported)
    k. Write Zeroes Command (if supported)
    l. Sanitize Namespace Command (if supported)

Observable Results:

1. Verify that each command is aborted with a status of 'Sanitize In Progress' (1Dh).

Case 10: Sanitize on Write Protected Namespace (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. Check the NWPC field. If Namespace Write Protection is not supported, this test is not applicable.
2. Perform a Get Features command for the Namespace Write Protection Config (FID 84h). Verify that Write Protection State is not cleared to 000b. If cleared to 000b, perform a Set Features command to set Namespace Write Protection Config to one of the following statuses:
    a. 'Write Protect' (001b);
    b. 'Write Protect Until Power Cycle' (010b); or
    c. 'Permanent Write Protect' (011b).
3. Perform a Sanitize command.

© 2026 UNH–IOL

Observable Results:

1. Verify that all commands prior to the Sanitize command complete successfully.
2. Verify that the Sanitize command completes with a status code of 'Namespace is Write Protected' (20h).

Sub-Cases: For the following subcase, unless otherwise noted, perform the same test procedure for case 1.17.10 and expect the same observable results.

1. Sanitize Namespace on Write Protected Namespace

Sub-case Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID 05h). If bit 0 of the Commands Supported and Effects Data Structure for the Sanitize Namespace command (opcode 8Ch) is set to 0h then this test is not applicable.
2. Configure the NVMe Host to issue a Set Features command with FID set to 84h (Namespace Write Protection Config) targeting an active namespace to set the protection state to Write Protect (001b).
3. Configure the NVMe Host to issue a Get Features command with FID set to 84h targeting the same namespace, and verify that the Write Protection State field returned is set to 001b.
4. Configure the NVMe Host to issue a Sanitize Namespace command targeting the write-protected namespace with SANACT set to 100b (Crypto Erase), in place of the Sanitize command, and verify that it is aborted by the controller with a status code of 'Namespace is Write Protected' (20h).
5. Configure a Get Log Page command with LID set to 81h (Sanitize Status Log Page) targeting the write-protected namespace, and verify that the SANS field is set to 0h (Idle).

Case 11: Media Verification - Media Verification Supported (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bit 3 of SANICAP, VERS, is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize command with Enter Media Verification State (EMVS) set to 1b and the Sanitize Action (SANACT) set to either Block Erase (010b) or Crypto Erase (100b).
3. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation is complete indicated by Sanitize Progress (SPROG) set to FFFFh.
4. Configure the NVMe Host to issue a Sanitize command with Sanitize Action (SANACT) set to Exit Media Verification State (101b).
5. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation is complete indicated by Sanitize Progress (SPROG) set to FFFFh.

Observable Results:

1. Verify the Sanitize command in step 2 completed with no error, and successfully caused a transition to the Media Verification State.
2. Verify the Sanitize command in step 4 completed with no error, the Sanitize State transitions to the Post-Verification Deallocation state (6h), ends in the Idle State (0h), but does not transition to the Restricted Processing State (1h) or the Unrestricted Processing State (3h).
3. Verify all other commands completed successfully.

Case 12: Media Verification - SPROG=FFFFh, SSI=5b (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bit 3 of SANICAP, VERS is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize command with Enter Media Verification State (EMVS) set to 1b and the Sanitize Action (SANACT) set to either Block Erase (010b) or Crypto Erase (100b).

© 2026 UNH–IOL

3. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation is complete indicated by Sanitize Progress (SPROG) set to FFFFh.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that after completion of the Sanitize operation specifying EMVS=1b, the Sanitize Log Page returns SPROG=FFFFh and SSI=5h.

Case 13: Media Verification - NVM Subsystem Reset Exits Media Verification State (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bit 3 of SANICAP, VERS, is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize command with Enter Media Verification State (EMVS) set to 1b and the Sanitize Action (SANACT) set to either Block Erase (010b) or Crypto Erase (100b).
3. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.
4. Configure the NVMe Host to issue an NVM Subsystem Reset.
5. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h).

Observable Results:

1. Verify all commands completed successfully.
2. Verify that after completion of the Sanitize operation specifying EMVS=1b, the final Sanitize Log Page in step 3, returns SSI set to 5h.
3. Verify that after completion of the Subsystem Reset the Sanitize Log Page from step 5 returns Media Verification Canceled (MVCNCLD) set to 1b and Sanitize State (SANS) set to 0h (Idle).

Case 14: Global Data Erase (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bits 2:0 of SANICAP are cleared to 0h this test is not applicable.
2. Configure the NVM Host to issue a Write command to write a block of FFs to SLBA 0.
3. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h).
4. Configure the NVMe Host to issue a Sanitize command with a Sanitize Action (SANACT) reported in the SANICAP field.
5. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that the Sanitize Log Page from step 3 reported Global Data Erase (GDE) cleared to 0b.
3. Verify that the final Sanitize Log Page in step 5 reports Global Data Erase (GDE) set to 1b.

Case 15: Sanitize Operation Status - Sanitized (SOS=001b) (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bits 2:0 of SANICAP are cleared to 0h this test is not applicable.

© 2026 UNH–IOL

2. Configure the NVMe Host to issue a Sanitize command with a Sanitize Action (SANACT) reported in the SANICAP field.
3. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that directly after the completion of the Sanitize operation the Sanitize Log Page reports Sanitize Operation Status (SOS) set to 001b (Sanitized).

Case 16: Sanitize Operation Status - Sanitizing (SOS=010b) (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bits 2:0 of SANICAP are cleared to 0h this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize command with a Sanitize Action (SANACT) reported in the SANICAP field.
3. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that all Sanitize Log Page commands while SPROG is less than FFFFh report Sanitize Operation Status (SOS) set to 010b (Sanitizing).

Case 17: Sanitize Operation Status - Sanitized Unexpected Deallocate (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bits 2:0 of SANICAP are cleared to 0h or No-Deallocate Inhibited (NDI) is cleared to 0b this test is not applicable.
2. Configure the NVMe Host to issue an Get Features command to Sanitize Config (FID 17h). If No-Deallocate Response Mode (NODRM) is disabled, configure the NVMe Host to issue a Set Features command to set NODRM to 1b.
3. Configure the NVMe Host to issue a Sanitize command with No-Deallocate After Sanitize (NDAS) set to 1b and a Sanitize Action (SANACT) reported in the SANICAP field.
4. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that directly after the completion of the Sanitize operation the Sanitize Log Page reports Sanitize Operation Status (SOS) set to 100b (Sanitized Unexpected Deallocate).

Case 18: Sanitize Command Dword 10 Information (SCDW10) (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bits 2:0 of SANICAP are cleared to 0h this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize command with a Sanitize Action (SANACT) reported in the SANICAP field.

© 2026 UNH–IOL

3. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that all Sanitize Log Page commands during the sanitize operation report Sanitize Command Dword 10 Information (SCDW10) set to the same value as the CDW10 used in the Sanitize command in step 2.

Case 19: Sanitize Operation Status (SOS!=011b) - Failure State=0b (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bit 3 of SANICAP, VERS, is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize command with a Sanitize Action (SANACT) reported in the SANICAP field.
3. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh

Observable Results:

1. Verify that if the Sanitize operation completed successfully, the Sanitize Log Page returns Sanitize Operation Status (SOS) set to a value other than 011b and Failure State (FAILS) is cleared to 0b. If the Sanitize operation failed, verify that SOS is set to 011b and FAILS is not cleared to 0h.
2. Verify all other commands completed successfully.

Case 20: Sanitize Command - VERS=0b and EMVS=1b (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bits 1:0 of SANICAP are cleared to 0h or bit 3 of SANICAP, VERS, is set to 1b, this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize command with Enter Media Verification State (EMVS) set to 1b and the Sanitize Action (SANACT) set to either Block Erase (010b) or Crypto Erase (100b).

Observable Results:

1. Verify the Identify command completed successfully.
2. Verify the Sanitize command was aborted with the status code of ‘Invalid Field in Command’.

Case 21: Media Verification - EMVS=1b and SANACT=011b (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bit 3 of SANICAP (VERS) is set to 0b or bit 2 (OWS) is set to 0b then this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize command with Enter Media Verification State (EMVS) set to 1b and the Sanitize Action (SANACT) set to Overwrite (011b).

Observable Results:

1. Verify the Identify command completed successfully.
2. Verify the Sanitize command was aborted with the status code of ‘Invalid Field in Command’.

Case 22: Media Verification - EMVS=1b and NDAS=1b (OF-FYI, FYI)

Test Procedure:

© 2026 UNH–IOL

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bit 3 of SANICAP, VERS, is set to 0b, this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize command with Enter Media Verification State (EMVS) set to 1b, No-Deallocate After Sanitize (NDAS) set to 1b, and the Sanitize Action (SANACT) set to either Block Erase (010b) or Crypto Erase (100b)

Observable Results:

1. Verify the Identify command completed successfully.
2. Verify the Sanitize command was aborted with the status code of 'Invalid Field in Command'

Case 23: Media Verification - Not in Media Verification State, SANACT=101b (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bit 3 of SANICAP, VERS, is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize command with (SANACT) set to Exit Media Verification State (101b).

Observable Results:

1. Verify the Identify command completed successfully.
2. Verify the Sanitize command was aborted with the status code of 'Invalid Field in Command'

Case 24: Sanitize Persists Across Controller Level Reset (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bits 2:0 of SANICAP are cleared to 0h this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize command with a Sanitize Action (SANACT) reported in the SANICAP field.
3. Configure the NVMe Host to issue a Controller Level Reset.
4. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.

Observable Results:

1. Verify Sanitize command completed successfully despite the Controller Level Reset.
2. Verify all other commands completed successfully.

Case 25: Media Verification - Sanitizing (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bit 3 of SANICAP, VERS, is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize command with Enter Media Verification State (EMVS) set to 1b and the Sanitize Action (SANACT) set to either Block Erase (010b) or Crypto Erase (100b).
3. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation is complete indicated by Sanitize Progress (SPROG) set to FFFFh.
4. Configure the NVMe Host to issue a Sanitize command with Sanitize Action (SANACT) set to Exit Media Verification State (101b).
5. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation is complete indicated by Sanitize Progress (SPROG) set to FFFFh

© 2026 UNH–IOL

Observable Results:

1. Verify all commands completed successfully
2. Verify all Sanitize Status Log Page commands returned Sanitize Operation Status (SOS) set to 010b when Sanitize State (SANS) was set any of the follow states:

Case 26: SANACT field not set to Exit Media Verification (101b) (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bit 3 of SANICAP, VERS, is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize command with Enter Media Verification State (EMVS) set to 1b and the Sanitize Action (SANACT) set to either Block Erase (010b) or Crypto Erase (100b).
3. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.
4. For each supported sanitize action reported in the SANICAP Field other than Exit Media Verification (101b), configure the NVMe Host a Sanitize command.

Observable Results:

1. Verify the Sanitize command in step 2 completed with no error, and successfully caused a transition to the Media Verification State.
2. Verify all Sanitize commands in step 4 were aborted with a status code of ‘Invalid Field in Command’.
3. Verify all other commands completed successfully

Case 27: Media Verification - Deallocates All Media Allocated for User Data (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bit 3 of SANICAP, VERS, is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue multiple Write commands to write 10 blocks of AAs 10 to SLBA 0.
3. Configure the NVMe Host to issue a Sanitize command with Enter Media Verification State (EMVS) set to 1b and the Sanitize Action (SANACT) set to either Block Erase (010b) or Crypto Erase (100b).
4. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.
5. Configure the NVMe Host to issue a Read command to read 10 blocks starting at LBA 0.

Observable Results:

1. Verify all other commands completed successfully.
2. Verify the Sanitize command in step 3 completed with no error, and successfully caused a transition to the Media Verification State.
3. Verify the data patterns read in step 5 do not match the data patterns written in step 2.

Case 28: Media Verification - Read with PRCHK > 0h (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bit 3 of SANICAP, VERS, is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. Check the OACS field to determine if the DUT supports the Format NVM command. If OACS bit 1 is cleared to 0h then this test is not applicable.
3. Configure the NVMe Host to issue an Identify command to obtain the Identify Namespace (CNS 00h) data structure. If DPC is cleared to 0h or no LBAF formats have a non-zero metadata size, this test is not applicable.

© 2026 UNH–IOL

4. Configure the NVMe Host to issue a Format NVM Command to an LBAF with a non-zero metadata size and Protection Information type set to a supported value.
5. Configure the NVMe Host to issue a Sanitize command with Enter Media Verification State (EMVS) set to 1b and the Sanitize Action (SANACT) set to either Block Erase (010b) or Crypto Erase (100b).
6. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.
7. Configure the NVMe Host to issue a Read command with PRCHK > 0h.

Observable Results:

1. Verify the Sanitize command in step 5 completed with no error, and successfully caused a transition to the Media Verification State.
2. Verify the Read command was aborted with a status code of 'Invalid Field in Command' (02h).
3. Verify all other commands completed successfully.

Case 29: Sanitize State Machine (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bit 3 of SANICAP, VERS, is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize command with Enter Media Verification State (EMVS) set to 1b and the Sanitize Action (SANACT) set to either Block Erase (010b) or Crypto Erase (100b).
3. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation is complete indicated by Sanitize Progress (SPROG) set to FFFFh. Record all state transitions indicating a change in the Sanitize State (SANS) value.
4. Configure the NVMe Host to issue a Sanitize command with Sanitize Action (SANACT) set to Exit Media Verification State (101b).
5. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation is complete indicated by Sanitize Progress (SPROG) set to FFFFh. Record all state transitions indicating a change in the Sanitize State (SANS) value.
6. Configure the NVMe Host to issue a Sanitize command with Sanitize Action (SANACT) set to either Block Erase (010b), Crypto Erase (100b), or Override (011b) and Allow Unrestricted Sanitize Exit (AUSE) cleared to 0b.
7. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation is complete indicated by Sanitize Progress (SPROG) set to FFFFh. Record all state transitions indicating a change in the Sanitize State (SANS) value.
8. Configure the NVMe Host to issue a Sanitize command with Sanitize Action (SANACT) set to either Block Erase (010b), Crypto Erase (100b), or Override (011b) and Allow Unrestricted Sanitize Exit (AUSE) set to 1b.
9. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h). Repeat this step periodically until the sanitize operation is complete indicated by Sanitize Progress (SPROG) set to FFFFh. Record all state transitions indicating a change in the Sanitize State (SANS) value.

Observable Results:

1. Verify all commands completed successfully.
2. For all Sanitize State transitions recorded during the sanitize operations in steps 3, 5, and 6, verify that only the following transactions occurred:
    a. Idle state (0h) to Restricted Processing or Unrestricted Processing,
    b. Restricted Processing state (1h) to Idle, Restricted Failure, or Media Verification,
    c. Unrestricted Processing state (3h) to Idle, Unrestricted Failure, or Media Verification,
    d. Media Verification State (5h) to Post-Verification Deallocation, or
    e. Post-Verification Deallocation state (6h) to Idle, Unrestricted Failure, or Restricted Failure

© 2026 UNH–IOL

Case 30: Sanitize Command Aborts While Sanitization Target is a Controller in a Suspended State (FYI)

Test Procedure:

1. Check the SANICAP field in the Identify Controller Data Structure on the migratable controller. If SANICAP is set to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Migration Send command to Controller A specifying Controller B with Select (SEL) set to 0h for the Suspend Management Operation, and DWORD 11 configured with Delete User Data Migration Queue (DUDMQ) set to 0h, and the Suspend Type (STYPE) Set to 1h (Suspend).
3. Configure the NVMe host to issue a Sanitize Command to a Controller A is, repeat for each supported operation: Overwrite, Block Erase, and Crypto Erase.
4. Configure the NVMe Host to issue a Migration Send command with Select (SEL) set to 1h for the Resume Management Operation.

Observable Results:

1. Verify that all Sanitize commands with supported actions in step 4 aborted with a status code of ‘Controller Suspended’ (39h).
2. Verify that all other commands completed with a status code of ‘Success’.

Case 31: Sanitize Execution Behavior with EMVS Bit Cleared (EMVS = 0b) (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID 05h). If bit 0 of the Commands Supported and Effects Data Structure for the Sanitize Namespace command (opcode 8Ch) is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bits 2:0 of SANICAP are cleared to 0h and bit 0 of the Commands Supported and Effects data structure for the Sanitize command (opcode 84h) Sanitize Namespace command (opcode 8Ch) returned in the previous step is cleared to 0h, this test is not applicable.
3. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If the Namespace Write Protection Capabilities (NWPC) field is cleared to 0h, this test is not applicable.
4. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status log page (LID 81h). Verify that the Sanitize State (SANS) field is cleared to 0h (Idle).
5. Issue one or both of the following Sanitize commands are supported, issue the corresponding Sanitize command with Enter Media Verification State (EMVS) cleared to 0b:
    a. If opcode 84h (Sanitize) is supported based on step 1, configure the NVMe Host to issue a Sanitize command (opcode 84h) with a supported SANACT code (e.g., Crypto Erase 100b or Block Erase 010b) and EMVS set to 0b.
    b. If opcode 8Ch (Sanitize Namespace) is supported based on step 1, configure the NVMe Host to issue a Sanitize Namespace command with SANACT set to 100b and EMVS set to 0b.
6. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h) with Retain Asynchronous Event (RAE) set to 1b. Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.
7. Configure the NVMe Host to issue a Sanitize command with SANACT set to 101b (EMVS).

Observable Results:

1. Verify the Sanitize or Sanitize Namespace command in step 5 is completed successfully.
2. Verify that the Sanitize command in step 7 is aborted by the controller with status code ‘Invalid Field in Command’ (02h).

© 2026 UNH–IOL

Case 32: EMVS Bit Ignored When SANACT Does Not Start a Sanitize Namespace Operation (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. Check if bits 2:0 of SANICAP are cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize command with a valid sanitize configuration and set the Enter Media Verification State (EMVS) bit to 1b.
3. Configure the NVMe Host to issue a Sanitize command with the Sanitize Action (SANACT) field set to 001b (Exit Media Verification State) and EMVS set to 1b.
4. Configure the NVMe Host to issue a Sanitize command with SANACT set to 001b (Exit Media Verification State).

Observable Results:

1. Verify the Sanitize command in steps 2 and 3 complete successfully.
2. Verify the Sanitize command in step 4 is aborted by the controller with status code 'Invalid Field in Command' (02h).

Case 33: Sanitize command Option and Rejection Constraints (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. Record the Sanitize Capabilities (SANICAP) field in the Identify Controller data structure. If SANICAP is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize command targeting an attached namespace with the SANACT set to 001b (Exit Failure Mode).
3. Configure the NVMe Host to issue a Sanitize command targeting an attached namespace with the SANACT set to 101b (Exit Media Verification State).
4. Configure the NVMe Host to issue a Sanitize command targeting an attached namespace with the SANACT set to 100b (Crypto Erase) and AUSE set to 1b.

Observable Results:

1. Verify that the Sanitize command in step 2 is aborted by the controller with status code 'Sanitize Failed' (0Ch).
2. Verify that the Sanitize command in step 3 is aborted by the controller with status code 'Invalid Field in Command' (02h).
3. Verify that the Sanitize command in step 4 is aborted by the controller with status code 'Sanitize Namespace Failed' (2Bh).

Case 34: Sanitize Namespace Interruption and Media Verification Canceled Tracking (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects log page (LID 05h). If bit 0 of the Commands Supported and Effects data structure for the Sanitize Namespace command (opcode 8Ch) is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status log page (LID 81h) with Retain Asynchronous Event (RAE) set to 0 to clear any previous events.
3. Configure the NVMe Host to issue a Sanitize Namespace command targeting a specific namespace with the Sanitize Action (SANACT) field set to 100b and the Enter Media Verification State (EMVS) bit set to 1b.
4. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status log page (LID 81h).
5. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status log page (LID 81h) with Retain Asynchronous Event (RAE) set to 1b. Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.
6. Configure the NVMe Host to send a NVM Subsystem Reset and wait for the controller to become ready again.

© 2026 UNH–IOL

7. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status log page (LID 81h).

Observable Results:

1. Verify that the Sanitize command in step 3 completed successfully.
2. Verify that the Media Verification Canceled (MVCNCLD) bit returned in the Get Log Page command in step 7, is set to 1b.

Case 35: Sanitize Command Abort and Restriction in Media Verification State (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If the Verification Support (VERS) bit is cleared to 0b, this test is not applicable.
2. Check bit 4 of the Sanitize Capabilities (SANICAP) field for Namespace Verification Support (NVERS). If bit 4 is cleared to 0b, this test is not applicable.
3. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status log page (LID 81h) with Retain Asynchronous Event (RAE) set to 0b to clear any previous asynchronous events.
4. Configure the NVMe Host to issue an Asynchronous Event Request command to the NVMe Controller.
5. Configure the NVMe Host to issue a Sanitize command with Crypto Erase (100b).
6. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status log page (LID 81h) with Retain Asynchronous Event (RAE) set to 1b. Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.
7. Configure the NVMe Host to issue an Identify command to obtain the Controller list (CNS 13h) data structure. Record the Controller IDs returned.
8. For every active controller discovered in the NVM subsystem, execute the following steps:
    a. Configure the NVMe Host to issue a Sanitize command to the current with the Sanitize Action (SANACT) field set to 010b (Start Block Erase). Record the completion status.
    b. Configure the NVMe Host to issue a Sanitize Namespace command to the current controller, with the SANACT field set to 010b (Start Block Erase). Record the completion status.
9. Configure the NVMe Host to issue a Sanitize command with the SANACT field configured to 101b (Exit Media Verification State) to clear the restricted state.

Observable Results:

1. Verify commands in steps 1, 3, 5, 8, and 9 completely successfully.
2. Verify that the Sanitize commands in steps 8.a and 8.b were aborted with status ‘Invalid Field in Command’ (02h).

© 2026 UNH–IOL

Test 1.18 – Virtualization Management Command (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller properly supports the Virtualization Management command if supported.

References:

Old Ref: [1] NVMe Specification 5.22, 8.5
NVM Express Base specification 2.0a : 5.28, 8.26

Resource Requirements:

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

Last Modification: December 1, 2019

Discussion: The Virtualization Management command is supported by primary controllers that support the Virtualization Enhancements capability. This command is used for Modifying Flexible Resource allocation for the primary controller, Assigning Flexible Resources for secondary controllers, and Setting the Online and Offline state for secondary controllers.

Test Setup: See Appendix A.

Case 1: Valid Virtualization Management Command, RT=000b (OF-FYI, FYI)

Test Procedure:

1. Check OACS field Bit 7 in the Identify Controller Data Structure to determine if the DUT supports the Virtualization Management command. If the DUT does not support the Virtualization Management command, this test is not applicable.
2. Perform an Identify command for CNS=14h and record the Primary Controller Capabilities Structure.
3. Configure the Testing Station to issue a Virtualization Management Command with Resource Type = VQ Resources (000b), ACT=1h (Primary Controller Flexible Allocation), valid CTRLID, valid NR (Number of Controller Resources).

Observable Results:

1. If Bit 0 (VQ Resources) of the CRT field in the Primary Controller Capabilities Structure is set to 1, then verify that the Virtualization Management commands completes with status Success. or Invalid Resource Identifier.
2. If Bit 0 (VQ Resources) of the CRT field in the Primary Controller Capabilities Structure is set to 0, then verify that the Virtualization Management commands completes with status Invalid Resource Identifier.

Case 2: Valid Virtualization Management Command, RT=001b (OF-FYI, FYI)

Test Procedure:

1. Check OACS field Bit 7 in the Identify Controller Data Structure to determine if the DUT supports the Virtualization Management command. If the DUT does not support the Virtualization Management command, this test is not applicable.
2. Perform an Identify command for CSN=14h and record the Primary Controller Capabilities Structure.
3. Configure the Testing Station to issue a Virtualization Management Command with Resource Type = VI Resources (001b), ACT=1h (Primary Controller Flexible Allocation), valid CTRLID, valid NR (Number of Controller Resources).

Observable Results:

1. If Bit 1 (VI Resources) of the CRT field in the Primary Controller Capabilities Structure is set to 1, then verify that the Virtualization Management commands completes with status Success.
2. If Bit 1 (VI Resources) of the CRT field in the Primary Controller Capabilities Structure is set to 0, then verify that the Virtualization Management commands completes with status Invalid Resource Identifier.

© 2026 UNH–IOL

Case 3: Invalid CNTLID (OF-FYI, FYI)

Test Procedure:

1. Check OACS field Bit 7 in the Identify Controller Data Structure to determine if the DUT supports the Virtualization Management command. If the DUT does not support the Virtualization Management command, this test is not applicable.
2. Perform an Identify command for CSN=14h and record the Primary Controller Capabilities Structure.
3. Configure the Testing Station to issue a Virtualization Management Command with Resource Type set to a supported value (based on the contents of the Primary Controller Capabilities Structure), ACT=1h (Primary Controller Flexible Allocation), valid NR (Number of Controller Resource, and an invalid CNTLID.

Observable Results:

1. Verify that the Virtualization Management commands completes with status Invalid Controller Identifier.

Possible Problems: None known.

© 2026 UNH–IOL

Test 1.19 – Security Receive (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller can properly perform a Security Receive operation.

References:

Old Ref: [1] NVMe Specification 5.25
NVM Express Base specification 2.0a : 5.25

Resource Requirements:

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

Last Modification: May 5, 2020

Discussion: The Security Receive command transfers the status and data result of one or more Security Send commands that were previously submitted to the controller.

Test Setup: See Appendix A.

Case 1: SECP = 00h (OF-FYI, M)

Test Procedure:

1. Check Bit 0 of the OACS field in the Identify Controller Data Structure. If Bit 0 is set to 0, this test is not applicable.
2. Configure the Testing Station to issue a Security Receive Command with SECP=00h.

Observable Results:

1. Verify that the controller returned a list of supported security protocols.

Case 2: SECP = Unsupported Value (OF-FYI, M)

Test Procedure:

1. Check Bit 0 of the OACS field in the Identify Controller Data Structure. If Bit 0 is set to 0, this test is not applicable.
2. Configure the Testing Station to issue a Security Receive Command with SECP=00h.
3. Configure the Testing Station to issue a Security Receive Command with SECP= any value not included in the list of supported security protocols returned in response to the previous Security Receive command.

Observable Results:

1. Verify that the second Security Receive command aborted with status Invalid Field in Command.

Possible Problems: None known.

© 2026 UNH–IOL

Test 1.20 – Security Send (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller can properly perform a Security Send operation.

References:

Old Ref: [1] NVMe Specification 5.26
NVM Express Base specification 2.0a : 5.26

Resource Requirements:

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

Last Modification: May 5, 2020

Discussion: The Security Send command is used to transfer security protocol data to the controller. The data structure transferred to the controller as part of this command contains security protocol specific commands to be performed by the controller.

Test Setup: See Appendix A.

Case 1: SECP = Unsupported Value (OF-FYI, M)

Test Procedure:

1. Check Bit 0 of the OACS field in the Identify Controller Data Structure. If Bit 0 is set to 0, this test is not applicable.
2. Configure the Testing Station to issue a Security Receive Command with SECP=00h.
3. Configure the Testing Station to issue a Security Send Command with SECP= any value not included in the list of supported security protocols returned in response to the previous Security Receive command.

Observable Results:

1. Verify that the second Security Send command aborted with status Invalid Field in Command.

Possible Problems: None known.

© 2026 UNH–IOL

Test 1.21 – Command and Feature Lockdown (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller can properly prohibit the execution of commands.

References:

NVM Express Base specification 2.0a : 5.16, 5.19, 8.4

Resource Requirements:

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

Last Modification: May 5, 2020

Discussion: The Lockdown command is used to control the Command and Feature Lockdown capability (refer to section 8.4) which configures the prohibition or allowance of execution of the specified command or Set Features command targeting a specific Feature Identifier.

Test Setup: See Appendix A.

Case 1: Lockdown a not prohibited Opcode or Feature Identifier (OF-FYI, M)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Command and Feature Lockdown Capabilities (bit 10). If the DUT does not support Lockdown feature, then this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page Command specifying a Log Identifier (LID) of 14h.
3. Choose a Feature Identifier or opcode that does not support the Lockdown feature.
4. Configure the NVMe Host to issue a Lockdown Command via Controller Admin Submission Queue on the Feature Identifier or opcode from step 3.

Observable Results:

1. Verify that the return status code from the Lockdown Command is Prohibition of Command Execution Not Supported.

Possible Problems: None known.

Case 2: Lockdown OACS check for all controllers (OF-FYI, M)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Command and Feature Lockdown Capabilities (bit 10) for all controllers in the NVMe Subsystem.

Observable Results:

1. Verify that OACS Command and Feature Lockdown Capabilities (bit 10) are all set to the same value

Possible Problems: None known.

Case 3: Lockdown with incorrect Interface Field (OF-FYI, FYI)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Command and Feature Lockdown Capabilities (bit 10). If the DUT does not support Lockdown feature, then this test is not applicable.
2. If no Management Endpoint is present in the NVM Subsystem then this test is not applicable
3. Configure the NVMe Host to issue a Get Log Page Command specifying a Log Identifier (LID) of 14h.
4. Choose a Feature Identifier or opcode that does support the Lockdown feature.
5. Configure the NVMe Host to issue a Lockdown Command via Controller Admin Submission Queue on the Feature Identifier or opcode from step 4, with the Interface Field set to 01b or 10b.

© 2026 UNH–IOL

Observable Results:

1. Verify that the return status code from the Lockdown Command is Invalid Field in Command.

Possible Problems: None known.

Case 4: Lockdown with incorrect Interface Field and Scope (OF-FYI, M)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Command and Feature Lockdown Capabilities (bit 10). If the DUT does not support Lockdown feature, then this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page Command specifying a Log Identifier (LID) of 14h.
3. Choose a Feature Identifier or opcode that does support the Lockdown feature.
4. Configure the NVMe Host to issue a Lockdown Command via Controller Admin Submission Queue on the Feature Identifier or opcode from step 3, with the Interface Field set to 00b or 01b and Scope field set to 4h.

Observable Results:

1. Verify that the Lockdown Command returned a status code of 'Invalid Field in Command' (02h).

Possible Problems: None known.

Case 5: Lockdown Get Log Page, no Management Endpoint (OF-FYI, FYI)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Command and Feature Lockdown Capabilities (bit 10). If the DUT does not support Lockdown feature, then this test is not applicable.
2. If the NVM Subsystem under test contains a Management Endpoint then this test is not applicable.
3. Configure the NVMe Host to issue a Get Log Page Command specifying a Log Identifier (LID) of 14h, with the Contents field set to 10b.

Observable Results:

1. Verify that the return status code from the Lockdown Command is Invalid Field in Command.

Possible Problems: None known.

Case 6: Lockdown Get Log Page, Length field cleared (OF-FYI, FYI)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Command and Feature Lockdown Capabilities (bit 10). If the DUT does not support Lockdown feature, then this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page Command specifying a Log Identifier (LID) of 14h.

Observable Results:

1. Verify that if the Command and Feature Identifier List field contains no coded values, the Length field is cleared to 0h

Possible Problems: None known.

Case 7: Lockdown Command and Feature Identifier List (OF-FYI, M)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Command and Feature Lockdown Capabilities (bit 10). If the DUT does not support Lockdown feature, then this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page Command specifying a Log Identifier (LID) of 14h, a Contents Selected field set to 00h and Scope Selected field of 0h or 2h.

© 2026 UNH–IOL

Observable Results:

1. Verify that contents of the Command and Feature Identifier List are ordered from lowest numerical order to highest numerical order

Possible Problems: None known.

Case 8: Lockdown Change Aborted (OF-FYI, M)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Command and Feature Lockdown Capabilities (bit 10). If the DUT does not support Lockdown feature, then this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page Command specifying a Log Identifier (LID) of 14h.
3. Choose a Feature Identifier or opcode that does support the Lockdown feature.
4. Configure the NVMe Host to issues a Lockdown Command on the chosen Feature Identifier or opcode.
5. Configure the NVMe Host to perform a change operation on that chosen Feature Identifier or opcode.

Observable Results:

1. Verify that the Lockdown of the chosen Feature Identifier or opcode completed successfully.
2. Verify that the chosen Feature Identifier or opcode does not change and returns a status code of Command Prohibited by Command and Feature Lockdown, in step 5.

Possible Problems: None known.

Case 9: Lockdown persists until power cycle (OF-FYI, M)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Command and Feature Lockdown Capabilities (bit 10). If the DUT does not support Lockdown feature, then this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page Command specifying a Log Identifier (LID) of 14h.
3. Choose a Feature Identifier or opcode that does support the Lockdown feature.
4. Configure the NVMe Host to issues a Lockdown Command on the chosen Feature Identifier or opcode.
5. Configure the NVMe Host to perform a change operation on that chosen Feature Identifier or opcode.
6. Power cycle the DUT.
7. Configure the NVMe Host to perform a change operation on that chosen Feature Identifier or opcode.

Observable Results:

1. Verify that the Lockdown of the chosen Feature Identifier or opcode completed successfully.
2. Verify that the chosen Feature Identifier or opcode does not change and returns a status code of Command Prohibited by Command and Feature Lockdown.
3. Verify that after power cycle of DUT, the Lockdown feature is removed to allow execution of command

Possible Problems: None known.

Case 10: Lockdown persists until next Lockdown Command (OF-FYI, M)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Command and Feature Lockdown Capabilities (bit 10). If the DUT does not support Lockdown feature, then this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page Command specifying a Log Identifier (LID) of 14h.
3. Choose a Feature Identifier or opcode that does support the Lockdown feature.
4. Configure the NVMe Host to issues a Lockdown Command on the chosen Feature Identifier or opcode.
5. Configure the NVMe Host to perform a change operation on that chosen Feature Identifier or opcode.
6. Configure the NVMe Host to disable the Lockdown feature on the chosen Feature Identifier or opcode.
7. Configure the NVMe Host to perform a change operation on that chosen Feature Identifier or opcode.

© 2026 UNH–IOL

Observable Results:

1. Verify that the Lockdown of the chosen Feature Identifier or opcode completed successfully.
2. Verify that the chosen Feature Identifier or opcode does not change and returns a status code of Command Prohibited by Command and Feature Lockdown, in step 5.
3. Verify that after Lockdown feature is disabled, the Lockdown feature is removed to allow execution of command.

Possible Problems: None known.

© 2026 UNH–IOL

Test 1.22 – Capacity Management (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller can configure Endurance Groups and NVM Sets in an NVM subsystem.

References:

NVM Express Base specification 2.0a : 5.3

Resource Requirements:

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

Last Modification: May 5, 2020

Discussion: The Capacity Management command allows host software to select one of a set of configured Endurance Groups and NVM Sets in an NVM subsystem, Fixed Capacity Management, or by specifying the capacity of the Endurance Group or NVM Set to be created, Variable Capacity management.

Test Setup: See Appendix A.

Case 1: Capacity Management command, Invalid Field (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0h, then this test is not applicable.
2. Check Bit 2 (NVM Set) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is set to 0h, then this test is not applicable.
3. Perform Get Log Page for the Supported Capacity Configuration List Log Page (11h). Collect each NVM Set Identifier in the Endurance Group Configuration Descriptor.
4. Perform a Capacity Management command (opcode=20h), with a non-zero Element Identifier that does not correspond to an existing NVM Set Identifier from the Get Log Page, with an Operation of Create NVM Set.

Observable Results:

1. Verify that the Capacity Management command with a status of ‘Invalid Field in Command’ (02h).

Case 2: Media Unit Configuration Selection create Endurance and NVM Set (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0h, then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 18h, store any valid Domain Identifier from the Domain List.
3. Perform Get Log Page for the Media Unit Status Log Page (10h) with a valid Domain Identifier, if the Selected Configuration is not cleared to ‘0h’, then perform a Capacity Management command with the Operation field set to Select Capacity Configuration and the Element Identifier cleared to ‘0h’.
4. Perform Get Log Page for the Supported Capacity Configuration List Log Page (11h) and store away the Capacity Configuration Identifier and associated Endurance group Identifiers and store away the original NVM Set Identifiers.
5. Perform a Capacity Management command, with the Operation field set to Select Capacity Configuration, with the Element Identifier field set to one of the Capacity Configuration Descriptors.
6. Perform Get Log Page for the Media Unit Status Log Page (10h) and store away the Selected Configuration and Endurance group and store away the original NVM Set Identifiers.

Observable Results:

1. Verify that all Get Log Page commands complete successfully

© 2026 UNH–IOL

2. Verify that an Endurance Group was created for each Endurance Group Configuration Descriptor in the Capacity Configuration Descriptor, if there are any
3. Verify that an NVM Set is created for each NVM Set Identifier specified in each Endurance Group configuration, if there are any

Case 3: Media Unit Configuration Selection delete Endurance and NVM Set (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 18h, store any valid Domain Identifier from the Domain List.
3. Perform Get Log Page for the Media Unit Status Log Page (10h) with a valid Domain Identifier.
4. Perform a Capacity Management command with Element Identifier field cleared to '0' and Operations field set to Select Capacity Configuration
5. Perform Get Log Page for the Media Unit Status Log Page (10h) with the valid Domain Identifier
6. Need to get Namespace associated with this Domain Identifier

Observable Results:

1. Verify that all Get Log Page commands complete successfully
2. Verify that all Endurance Groups and NVM Sets in the Domain of the controller are deleted
3. Verify that all namespaces in the domain that contains the controller processing the command are deleted.

Case 4: Media Unit Configuration Selection, No Change (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 18h, store any valid Domain Identifier from the Domain List.
3. Perform a Get Log Page for the Media Unit Status Log Page (10h) with a valid Domain Identifier. If the Selected Configuration is cleared to 0, then perform a Capacity Management command, with the Operation field set to Select Capacity Configuration, and Element Identifier cleared to 0 and record the Selected Configuration. If the Selected Configuration is not cleared to 0, record the Selected Configuration.
4. Perform a Capacity Management command, with the Operation field set to Select Capacity Configuration, with the Element Identifier field set the stored Selected Configuration. Record the Selected Configuration after the command has completed.

Observable Results:

1. Verify that Capacity Management commands completed successfully
2. Verify that the Selected Configurations recorded at the end of steps 3 and 4 are the same

Case 5: Media Unit Configuration Selection, Invalid Field in Command (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Perform Get Log Page for the Media Unit Status Log Page (10h) and store fields
3. Perform a Capacity Management command with Element Identifier field set to an invalid value, such as FFFFFFFFh.
4. Perform Get Log Page for the Media Unit Status Log Page (10h)

Observable Results:

© 2026 UNH–IOL

1. Verify that the controller shall abort the Capacity Management command with a status code of 'Invalid Field in Command'
2. Verify that no changes are made to any Media Unit configuration for all Endurance Groups, NVM Sets, and Channels

Case 6: Create Endurance Group (OF-FYI, FYI)

Test Procedure:

1. Check Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure. If Bit 4 is set to 0, then this test is not applicable.
2. Perform a Capacity Management command with the Operations field set to Create Endurance Group. Set the Capacity Lower and Capacity Upper fields to be the capacity of the Endurance to be created.

Observable Results:

1. Verify that the controller shall returns a non-zero Endurance Group Identifier not assigned to an existing Endurance Group in DWord 0 of the completion queue entry.
2. Verify that if the controller does not have a non-zero unassigned Endurance Group Identifier available, it will abort the command with a status of 'Identifier Unavailable' (2Dh).

Case 7: Create Endurance Group, Insufficient Capacity (OF-FYI, FYI)

Test Procedure:

1. Check Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure. If Bit 4 is set to 0, then this test is not applicable.
2. Perform a Capacity Management command with the Operations field set to Create Endurance Group. Set the Capacity Lower and Capacity Upper fields to be greater than the Max Endurance Group Capacity (MEGCAP), the Unallocated NVM Capacity (UNVMCAP), both in the Identify Controller data structure and greater than the Max Endurance
3. Group Capacity (MEGCAP) field.

Observable Results:

1. Verify that the controller shall abort the command with Insufficient Capacity status and if Error Information log is supported it shall indicate the Command Specific Information the total amount of NVM capacity in bytes of the largest Endurance Group that is able to be created.

Case 8: Delete Endurance Group, Invalid Field in Command (OF-FYI, FYI)

Test Procedure:

1. Check Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure. If Bit 4 is set to 0, then this test is not applicable.
2. Check Bit 13 of the CTRATT field of the Identify Controller Data Structure and log Delete Endurance Group setting, if Bit 13 is set to '0', then this test is not applicable.
3. Perform a Capacity Management command with the Operations field set to Delete Endurance Group. Set the Element Identifier field to a value of 0h.

Observable Results:

1. Verify that the controller shall abort the command with Invalid Field in Command.

Case 9: Create NVM Set (OF-FYI, FYI)

Test Procedure:

1. Check Bit 2 (NVM Set) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is set to 0, then this test is not applicable.

© 2026 UNH–IOL

2. Perform a Capacity Management command with the Operations field set to Create NVM Set. Set the Capacity Lower and Capacity Upper fields to be the capacity of the NVM Set, using a specified Endurance Group.

Observable Results:

1. Verify that the controller created the new NVM Set and shall select a non-zero NVM Set Identifier not assigned to an existing NVM Set in the specified Endurance Group and that the new value is indicated in the new completion queue entry.
2. Verify that if the controller does not have a non-zero unassigned NVM Set Identifier available, it will abort the command with a status of 'Identifier Unavailable' (2Dh).

Case 10: Create NVM Set, Insufficient Capacity (OF-FYI, FYI)

Test Procedure:

1. Check Bit 2 (NVM Set) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is set to 0, then this test is not applicable.
2. Perform a Get Log page on the Endurance Group Information log page.
3. Perform a Capacity Management command with the Operations field set to Create NVM Set Operation. Set the Capacity Lower and Capacity Upper fields to be greater than the Endurance Group Capacity (UEGCAP) in the Endurance Group Information log page.

Observable Results:

1. Verify that the controller shall abort the command with Insufficient Capacity status and if Error Information log is supported it shall indicate the Command Specific Information the total amount of NVM capacity in bytes of the largest NVM Set that is able to be created.

Case 11: Create NVM Set, Element Identifier=0 (OF-FYI, FYI)

Test Procedure:

1. Check Bit 2 (NVM Set) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is set to 0, then this test is not applicable.
2. Perform a Identify Command NVM Set List (CNS 04h)
3. Perform a Capacity Management command with the Operations field set to Create NVM Set Operation. Set the Element Identifier to 0h.
4. Perform a Identify Command NVM Set List (CNS 04h)

Observable Results:

1. Verify that the controller creates the new NVM Set in an existing Endurance Group in an existing domain.

Case 12: Delete NVM Set, Invalid Field in Command (OF-FYI, FYI)

Test Procedure:

1. Check Bit 2 (NVM Set) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is cleared to 0, then this test is not applicable.
2. Check Bit 14 of the CTRATT field of the Identify Controller Data Structure and log support for Delete NVM Set, if Bit 14 is cleared to '0', then this test is not applicable.
3. Perform a Capacity Management command with the Operations field set to Delete NVM Set. Set the Element Identifier field to a value of 0h.

Observable Results:

1. Verify that the controller shall abort the command with Invalid Field in Command.

Case 13: Fixed and/or Variable Capacity Management supported (OF-FYI, FYI)

Test Procedure:

© 2026 UNH–IOL

1. Check the Identify Controller Data Structure to determine what version of the NVMe specification the product under test claims to support. If the product under test does not support NVMe v2.0 or higher, this test is not applicable.
2. Check Bit 11 and 12 (Fixed Capacity Management and Variable Capacity Management) of the CTRATT field of the Identify Controller Data Structure.

Observable Results:

1. Display the controller support for the Fixed or Variable Capacity Management features (bit 11 or 12).

Case 14: Fixed Capacity Management Support (OF-FYI, FYI)

Test Procedure:

1. Check Bit 11 (Fixed Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 11 is set to 0, then this test is not applicable.
2. Check Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure.
3. Perform a Get Log Page on Media Unit Status Log Page
4. Perform a Get Log Page on the Supported Capacity Configuration List log page.
5. Perform Capacity Management command with Operations field set to Select Capacity Configuration, the Element Identifier specifies a Capacity Configuration Descriptor in the Supported Capacity Configuration List page

Observable Results:

1. Verify that the controller has bit 11 (Fixed Capacity Management) of the CTRATT field set to ‘1’
2. Verify that the controller has bit 4 (Endurance Groups) of the CTRATT field set to ‘1’
3. Verify that the Get Log Page on Media Unit Status Log page returned successfully
4. Verify that the Get Log Page on the Supported Capacity Configuration log page returned successfully

Case 15: Variable Capacity Management Support (OF-FYI, FYI)

Test Procedure:

1. Check Bit 12 (Variable Capacity Management) of the CTRATT field of the Identify Controller Data Structure. If Bit 12 is set to 0, then this test is not applicable.
2. Check Bit 4 (Endurance Groups) and Bit 2 (NVM Sets) of the CTRATT field of the Identify Controller Data Structure.
3. Perform Capacity Management command with Operations field set to Create Endurance Group, With the Operations fields set to Create Endurance Group, the Element Identifier specifies a Capacity Configuration Descriptor in the Supported Capacity Configuration List page
4. If NVM Sets are supported, perform a Capacity Management command with the Operations field set to Create NVM Set. Set the Capacity Lower and Capacity Upper fields to be the capacity of the NVM Set, using a specified Endurance Group
5. Perform a Endurance Group Log (09h).
6. If NVM Sets are supported, then perform a NVM Set List (CNS 04h)

Observable Results:

1. Verify that the controller has Bit 12 (Variable Capacity Management) of the CTRATT field set to ‘1’
2. Verify that the controller has Bit 4 (Endurance Groups) and Bit 2 (NVM Sets) of the CTRATT field set to ‘1’
3. Verify that the Capacity Management command to create Endurance Groups returned successfully.
4. Verify that the Capacity Management command to create NVM Sets returned successfully and verify that the NVM Set List contains the new NVMSETID.

Possible Problems: None known.

© 2026 UNH–IOL

Test 1.23 – Cancel Command (FYI)

Purpose: The Cancel Command is used to request an abort for specified commands submitted on the same I/O Submission Queue. This test is not applicable to NVMe over Fabrics implementations.

References:

NVM Express Base specification 2.0 : 7.1

Resource Requirements:

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

Last Modification: May 5, 2020

Discussion: The Cancel Command may apply to a single namespace or multiple namespaces.

Test Setup: See Appendix A.

Case 1: Cancel I/O Command – Single Command Cancel (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID 05h). If bit 0 of the Commands Supported and Effects Data Structure for the Cancel command (Opcode 18h) is set to 0h then this test is not applicable.
2. Configure the NVMe Host to issue two Read commands to read a single LBA. Issue both read commands to the same I/O submission queue.
3. Configure the NVMe Host to issue a Cancel command specifying:
    a. Command Identifier (CID) to be the CID of the first Read command.
    b. Submission Queue Identifier (SQID) to be the I/O queue the Read commands were issued.
    c. The Action Code set to ‘Single Command Cancel’ (00b); and
    d. The NSID field set to FFFFFFFFh.
4. If the first Read command is processed off the submission queue prior to the Cancel command, this test is not applicable.

Observable Results:

1. Verify the Get Log Page command completed successfully.
2. Determine the outcome of the Cancel command action by examining the ‘Commands Eligible for Deferred Abort’ and ‘Commands Aborted’ fields of Dword 0 of the completion queue entry for the Abort command.
    a. If Commands Aborted is set to 1h:
        i. Verify that the Read command targeted by the Cancel command completed with status ‘Command Abort Requested’ (07h).
        ii. Verify that the Completion Queue Entry for the aborted Read command is posted to the I/O Completion Queue before the Completion Queue Entry of the Abort command is posted to the Admin Completion Queue.
        iii. Verify Command Eligible for Deferred Abort is cleared to 0h.
    b. If Command Eligible for Deferred Abort is set to 1h and Read command completed with status ‘Command Abort Requested’ (07h):
        i. Verify the Commands Aborted field is cleared to 0h.
3. Verify the second Read command completed successfully.

Case 2: Cancel I/O Command – Invalid Command ID (FYI)

Test Procedure:

© 2026 UNH–IOL

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID 05h). If bit 0 of the Commands Supported and Effects Data Structure for the Cancel command (Opcode 18h) is cleared to 0 then this test is not applicable.
2. Configure the NVMe Host to issue a Cancel command specifying:
    1. Command Identifier (CID) to be the CID of the Cancel command currently being issued;
    2. Submission Queue Identifier (SQID) set to the QID of any valid I/O queue;
    3. The Action Code set to 'Single Command Cancel' (00b); and
    4. The NSID field set to FFFFFFFFh.

Observable Results:

1. Verify the Get Log Page completed successfully.
2. Verify the Cancel command returns with status: 'Invalid Command ID'.

Case 3: Cancel I/O Command – Incorrect SQID (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID 05h). If bit 0 of the Commands Supported and Effects Data Structure for the Cancel command (Opcode 18h) is cleared to 0h then this test is not applicable.
2. Configure the NVMe Host to issue a Read command to read one LBA.
3. Configure the NVMe Host to issue a Cancel command specifying:
    a. Command Identifier (CID) to be the CID of the Read command.
    b. Submission Queue Identifier (SQID) set to a Submission Queue Identifier that does not match the queue to which the Cancel command is submitted to.
    c. The Action Code set to 'Single Command Cancel' (00b); and
    d. The NSID field set to FFFFFFFFh.

Observable Results:

1. Verify the Cancel command returns with status: 'Invalid Field in Command' (02h).
2. Verify all other commands completed successfully.

Case 4: Cancel I/O Command – Single Command Cancel, Different Namespace (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID 05h). If bit 0 of the Commands Supported and Effects Data Structure for the Cancel command (Opcode 18h) is cleared to 0h then this test is not applicable.
2. Configure the NVMe Host to issue a Read command read to one LBA.
3. Configure the NVMe Host to issue a Cancel command specifying:
    a. Command Identifier (CID) to be the CID of the Read command.
    b. Submission Queue Identifier (SQID) to be the queue the Write command was issued to.
    c. The Action Code set to 'Single Command Cancel' (00b); and
    d. The NSID field set to a NSID that was not used in the Read command.

Observable Results:

1. Verify the Cancel command returns with status: 'Invalid Field in Command' (02h).
2. Verify all other commands completed successfully.

Possible Problems: There may be a single namespace in the NVM subsystem. In this case, the NVMe Host may create and attach a namespace to the controller in order to perform step 3.

Case 5: Cancel I/O Command – Single Command Cancel, Command not Aborted (FYI)

Test Procedure:

© 2026 UNH–IOL

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID 05h). If bit 0 of the Commands Supported and Effects Data Structure for the Cancel command (Opcode 18h) is set to 1h then this test is not applicable.
2. Configure the NVMe Host to issue a Read command read to one LBA.
3. Configure the NVMe Host to issue a Cancel command specifying:
    a. Command Identifier (CID) to be a value that does not correspond to any commands on the I/O submission queue specified.
    b. Submission Queue Identifier (SQID) to be the queue the Read command was issued to.
    c. The Action Code set to 'Single Command Cancel' (00b); and
    d. The NSID field set to FFFFFFFFh.

Observable Results:

1. Verify all commands completed successfully.
2. Verify the Cancel command returns "Commands Eligible for Deferred Abort" and "Commands Aborted" fields of Dword 0 cleared to 0h.

Case 6: Cancel I/O Command – Multiple Command Cancel (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID 05h). If bit 0 of the Commands Supported and Effects Data Structure for the Cancel command (Opcode 18h) is cleared to 0h then this test is not applicable.
2. Configure the NVMe Host to submit 10 Write commands with a known data pattern to write to the I/O Submission Queue.
3. Configure the NVMe Host to issue a Cancel command specifying:
    a. Command Identifier (CID) set to FFFFh;
    b. Submission Queue Identifier (SQID) to be the I/O queue the Write commands were issued;
    c. The Action Code set to 'Multiple Command Cancel' (01b); and
    d. The NSID field set to FFFFFFFFh.
4. Configure the device to process all commands on the submission queue.

Observable Results:

1. Verify the Get Log Page command completed successfully.
2. Verify that any write command which posts a completion queue after the processing of the Cancel command completes with a status of 'Command Abort Requested' (07h).
3. Verify that the sum of 'Commands Aborted' and 'Commands Eligible for Deferred Abort' of Dword 0 of the Cancel command return structure is less than or equal to the number of write commands which were aborted.

Case 7: Cancel I/O Command – Multiple Command Cancel, CID not equal FFFFh (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID 05h). If bit 0 of the Commands Supported and Effects Data Structure for the Cancel command (Opcode 18h) is cleared to 0h then this test is not applicable.
2. Configure the NVMe Host to issue a Read command to read one LBA.
3. Configure the NVMe Host to issue a Cancel command specifying:
    a. Command Identifier (CID) set to the CID of the Read command.
    b. Submission Queue Identifier (SQID) set to the queue the Read command was issued to.
    c. The Action Code set to 'Multiple Command Cancel' (01b); and
    d. The NSID field set to FFFFFFFFh.

Observable Results:

1. Verify the Cancel command returns with status: 'Invalid Field in Command' (02h).
2. Verify all other commands completed successfully.

© 2026 UNH–IOL

Case 8: Cancel I/O Command – Multiple Command Cancel, Different Namespace (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID 05h). If bit 0 of the Commands Supported and Effects Data Structure for the Cancel command (Opcode 18h) is cleared to 0h then this test is not applicable.
2. Configure the NVMe Host to issue 10 Write commands with a known data pattern to write to the device under test.
3. Configure the NVMe Host to issue 10 Write commands with a known data pattern to write to the device under test on a namespace which is different from the namespace written to in step 2.
4. Configure the NVMe Host to issue a Cancel command specifying:
    a. Command Identifier (CID) set to FFFFh;
    b. Submission Queue Identifier (SQID) to be the queue the Write commands were issued to;
    c. The Action Code set to 'Multiple Command Cancel' (01b); and
    d. The NSID field set to the NSID of the namespace written to by the second 10 Write commands issued in step 3.
5. Configure the device to process all commands on the submission queue.

Observable Results:

1. Verify the Get Log Page command completed successfully.
2. Verify that the Write commands from step 2 completed successfully.
3. Verify that any Write command which posts a completion queue after the processing of the Cancel command completes with a status of 'Command Abort Requested' (02h).
4. Verify that the sum of 'Commands Aborted' and 'Commands Eligible for Deferred Abort' of Dword 0 of the Cancel command return structure is less than or equal to the number of write commands which were aborted.

Possible Problems: There may be a single namespace in the NVM subsystem. In this case, the NVMe Host may create and attach a namespace to the controller in order to perform step 3.

© 2026 UNH–IOL

Test 1.24 – Keep Alive Timer (OF-FYI, FYI)

Purpose: To verify that the Keep Alive Timer is implemented as per requirements of ECN 117 KATO Clarifications.

References:

NVM Express Base specification 2.0d

Resource Requirements:

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

Last Modification: January 3, 2024

Discussion: The controller shall treat a Keep Alive Timeout in the same manner as connection loss. If the Keep Alive feature is in use and the timer expires, then the controller shall:

- stop processing commands and set the Controller Fatal Status (CSTS.CFS) bit to '1';
- terminate the NVMe Transport connection; and
- break the host to controller association.

After completing these steps, a controller may accept a Connect command for the Admin Queue from the same or another host in order to form a new association.

Test Setup: See Appendix A.

Case 1: KATO - KAS (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to send an Identify Command specifying CNS value 01h to the controller in order to receive back an Identify Controller Data Structure.
2. Check the KAS value of the returned Identify Controller Data Structure. If the value is 0 then this test is not applicable.
3. Configure the NVMe Host to issue a set feature command to set the desired KATO value
4. Configure the NVMe Host to issue a Keep Alive Command by sending an Admin Command with opcode 18h.

Observable Results:

1. Verify that the Keep Alive command completes with status 'Success'.

Case 2: KATO – Keep Alive Timer Clear to 0h (OF-FYI, FYI)

Test Procedure:

1. This test is only applicable if the transport binding specification requires the use of the Keep Alive Timer Feature
2. Configure the NVMe Host to issue a Get Feature Command with FID=0Fh, Keep Alive Timer.
3. Configure the NVMe Host to issue a Set Feature Command with FID=0Fh and include the Keep Alive Value 0h.
4. Configure the NVMe Host to issue a Get Feature Command with FID=0Fh, Keep Alive Timer

Observable Results:

1. Verify the Keep Alive Timer Value returned in steps 2 and 4 are equal.
2. Verify the Set Feature Command sent in step 3 returns with the Status Code 'Keep Alive Timeout Invalid'.

Case 3: Keep Alive Timer Maximum Keep Alive Timeout (OF-FYI, FYI)

Test Procedure:

1. If the transport binding specification does not have a maximum Keep Alive Timeout Value, then this test is not applicable.

© 2026 UNH–IOL

2. Configure the NVMe Host to issue a Get Feature Command with FID=0Fh, Keep Alive Timer
3. Configure the NVMe Host to issue a Set Feature Command with FID=0Fh and include a Keep Alive Value greater than the maximum Keep Alive Timeout Value specified by the transport binding specification.
4. Configure the NVMe Host to issue a Get Feature Command with FID=0Fh, Keep Alive timer.

Observable Results:

1. The Keep Alive Timer Value returned in steps 2 and 4 are equivalent.
2. Verify the Set Feature Command sent in step 3 returns with the Status Code 'Keep Alive Timeout Invalid'.

Case 4: KATO Command Cleanup Within Quiesce Time (OF-FYI, FYI)

Test Procedure:

1. This test is only applicable if the transport binding specification requires the use of the Keep Alive Timer Feature.
2. Configure the NVMe Host to send an Identify Command specifying CNS value 01h to the controller in order to receive back an Identify Controller Data Structure.
3. Check the KAS value of the returned Identify Controller Data Structure. If the value is 0 then this test is not applicable.
4. Record the Command Quiesce Time (CQT) value of the returned Identify Controller Data Structure. If the value is 0 then this test is not applicable.
5. Configure the NVMe Host to issue a Set Features command with FID=0Fh to set the KATO value to 7530h (30 seconds).
6. Configure the NVMe Host to issue a Keep Alive command by sending an Admin Command with opcode 18h.
7. Wait 30 seconds to exceed the Keep Alive Timeout set in step 5.
8. Configure the NVMe Host to issue a Get Log Page command with LID=01h Error Information
9. Configure the NVMe Host to issue a Get Property command with Offset=1Ch for CSTS
10. Configure the NVMe Host to issue a Keep Alive command by sending an Admin Command with opcode 18h.

Observable Results:

1. Verify that the controller completes the steps below within the CQT time window:
2. Verify that the STATUS field in Error Information Log Entry from Step 8 returns the status code 'Keep Alive Timeout Expired'.
3. Verify that the Controller Fatal Status (CSTS.CFS) bit is set to '1'.
4. Verify that the Keep Alive command in step 10 is not posted to the Admin Completion Queue.

Case 5: Command Based Keep Alive Timer Restart (OF-FYI, FYI)

Test Procedure:

1. This test is only applicable if the transport binding specification requires the use of the Keep Alive Timer Feature.
2. Configure the NVMe Host to send an Identify Command specifying CNS value 01h to the controller to read the Identify Controller Data Structure.
3. Check the KAS value of the returned Identify Controller Data Structure. If the value is 0 then this test is not applicable.
4. Check the TBKAS value of the returned Identify Controller Data Structure. If the value is 1 then this test is not applicable.
5. Configure the NVMe Host to issue a Set Features command with FID=0Fh to set the KATO value to 7530h (30 seconds).
6. Configure the NVMe Host to issue a Get Property command with Offset=14h for CC
7. Configure the NVMe Host to issue a Get Property command with Offset=1Ch for CSTS

Observable Results:

1. Verify the Set Feature Command sent in step 5 returns with the Status Code 'Success'.
2. Verify the following bits are set to the following values to make sure that Keep Alive Timer is active.

© 2026 UNH–IOL

a. CC.EN is set to ‘1’
b. CSTS.RDY is set to ‘1’
c. CC.SHN is cleared to ‘00b’
d. CSTS.SHST is cleared to ‘00b’

© 2026 UNH–IOL

Test 1.25 – Rotational Media (OF-FYI, FYI)

Purpose: To verify that a controller that supports a NVM namespace on rotational media has the correct supported features and utilizes the new features and log pages correctly.

References:

NVM Express Base specification 2.0d

Resource Requirements:

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

Last Modification: January 3, 2024

Discussion: In NVMe 1.4, support for NVMe on rotational media (HDD) was added. Several new requirements, feature identifier, and a new log page were standardized to allow for NVMe operations to be performed on a non-NAND flash memory due to the differences in operational, endurance, and performance characteristics.

Test Setup: See Appendix A

Case 1: Rotational Media Support Requirements (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify Command for CNS 08h (I/O Command Set Independent Identify Namespace Data Structure). If bit 4 of NSFEAT is 0, this test is not applicable.
2. Configure the NVMe Host to issue an Identify Command for CNS 01h (Identify Controller Data Structure, I/O Command Set Independent)
3. Configure the NVMe Host to issue a Get Log Page command for LID 16h.
4. Configure the NVMe Host to issue a Get Log Page command for LID 09h.
5. Configure the NVMe Host to issue a Set Feature command for FID 1Ah with a value of 1.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that bit 4 of CTRATT returned in the data structure in step 2 is set to 1.
3. Verify that the EG Rotational Media bit is set to ‘1’ in the EGFEAT field in the Endurance Group Information log page returned in the Get Log Page command in step 4.

Case 2: Rotational Media Log Page 16h and Endurance Group Log Page Requirements (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify Command for CNS 08h (I/O Command Set Independent Identify Namespace Data Structure). If bit 4 of NSFEAT is 0, this test is not applicable.
2. Configure the NVMe host to issue a Get Log Page command for LID 16h, using the Endurance Group Identifier as the Log Specific Identifier.
3. Read the following fields
    a. Spinup Count
    b. Failed Spinup Count
    c. Load Count
    d. Failed Load Count
4. Transition the controller to a non-operational power state, as indicated by the NOPs field in the Power State Descriptor Data Structure within the – Identify Controller Data Structure, I/O Command Set Independent (CNS 01h)
5. Transition the controller back to the initial power state.

© 2026 UNH–IOL

6. Configure the NVMe host to issue a second Get Log Page command for LID 16h, using the same Endurance Group Identifier as the Log Specific Identifier.
7. Configure the host to issue a Get Log Page command for LID 09h with a valid ENDGID.

Observable Results:

1. Verify that all commands completed successfully.
2. Verify the following data is present in the returned log page.
    a. Bytes 1:0 are equal to the Endurance Group Identifier used in the Get Log Page command in step 2.
    b. Bytes 5:4 are not equal to 0001h.
    c. Spinup Count increased by 1.
    d. Failed Spinup Count did not change.
    e. Load Count increased by 1.
    f. Failed Load Count did not change.
    g. Bytes 511:24 are cleared to zero
3. Verify that bit 0 of the Endurance Group Features (EGFEAT) in the Endurance Group Information Log is set to 1.

Case 3: Rotational Media Log Page 16h During a Sanitize Operation (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify Command for CNS 08h (I/O Command Set Independent Identify Namespace Data Structure). If bit 4 of NSFEAT is 0, this test is not applicable.
2. Check SANICAP field in the Identify Controller Data Structure. If SANICAP is set to 0, this test is not applicable.
3. Configure the Testing Station to issue a Sanitize Command, for each of the operation types indicated as supported in the SANICAP field: Overwrite, Block Erase, and Crypto Erase. Each Sanitize command should be followed by a Get Log Page command for the Rotational Media Log Page.

Observable Results:

1. Verify that each of the supported Sanitize operation commands completes with status ‘Success’, and each of the unsupported Sanitize operation commands completes with status ‘Invalid Field in Command’ (02h).
2. Verify that the following fields in the Rotational Media Log Page all return zero as their value:
    a. Spinup Count
    b. Failed Spinup Count
    c. Load Count
    d. Failed Load Count

Possible Problem: This test procedure requires all three types of Sanitize Operations if they are supported. It is expected the different Sanitize operations will complete in different amounts of time. It is possible that a short Sanitize operation will complete prior to the host being able to check the value of SPROG to verify a Sanitize is in progress to complete the test requirements. If this occurs for all supported Sanitize Operations, then this test is unable to be performed with the standard INTERACT PC Edition test tools.

Case 4: FID 1Ah, Spinup Control on Rotational Media Persistence (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify Command for CNS 08h (I/O Command Set Independent Identify Namespace Data Structure). If bit 4 of NSFEAT is 0, this test is not applicable.
2. Perform a Get Feature specifying FID 1Ah, read bit 0 of Dword 0 and store the value.
3. Perform a Controller Level Reset
4. Perform second Get Feature specifying FID 1Ah.
5. Perform a Set Feature specifying FID 1Ah using a different value than what was returned in the Get Feature command in step 2 (e.g if the value returned in step 2 was 1, set the value to 0. If the value returned in step 2 was 0, set the value to 1)

© 2026 UNH–IOL

6. Perform third Get Feature specifying FID 1Ah.
7. Perform a second Controller Level Reset
8. Perform Fourth Get Feature specifying FID 1Ah.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the Set Feature command in step 6 successfully set FID 1Ah to a new value.
3. Verify that the value of FID 1Ah was the same between the Get Feature commands in step 2 and step 4.
4. Verify that the value of FID 1Ah was the same between the Get Feature commands in step 6 and 8 and is a different value than what was found in observable result 3.

Case 5: Rotational Media Log Page Initial Spinup After Subsystem Reset (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify Command for CNS 08h (I/O Command Set Independent Identify Namespace Data Structure). If bit 4 of NSFEAT is 0, this test is not applicable.
2. Configure the NVMe host to issue a Get Log Page command for LID 16h, using the Endurance Group Identifier as the Log Specific Identifier. Store the value of Spinup Count.
3. Configure the NVMe Host to send a Set Features command specifying FID 1Ah and value equal to 0 to disable the Spinup Control Feature
4. Configure the NVMe Host to send a NVM Subsystem Reset and wait for the controller to become ready again.
5. Configure the NVMe host to issue a Get Log Page command for LID 16h, using the Endurance Group Identifier as the Log Specific Identifier. Store the value of Spinup Count
6. Configure the NVMe Host to send a Set Features command specifying FID 1Ah and value equal to 1 to enable the Spinup Control Feature.
7. Configure the NVMe Host to send a second NVMe Subsystem Reset and wait for the controller to become ready again.
8. Configure the NVMe Host to issue a Get Log Page command for LID 16h, using the Endurance Group Identifier as the Log Specific Identifier. Store the value of Spinup Count
9. Configure the NVMe Host to issue a Set Features command that transitions the controller from a non-operational power state to an operational power state.
10. Perform one more Get Log Page Command for LID 16 with the ENDGID as the log specific identifier and record the value of Spinup Count

Observable Results:

1. Verify all commands completed successfully.
2. Verify that the Spinup Count increased by 1 after the first NVM Subsystem Reset in the log page returned in step 5.
3. Verify that the Spinup Count did not change from the previous value in the log page returned in step 8.
4. Verify that the Spinup Count increased by 1 in the log page returned in step 10.

Case 6: Rotational Media FID 1Ah (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify Command for CNS 08h (I/O Command Set Independent Identify Namespace Data Structure). If bit 4 of NSFEAT is 1, this test is not applicable.
2. Perform a Get Features command specifying FID 1Ah.
3. Perform Set Features command specifying FID 1Ah.

Observable Results:

1. Verify that both the Get and Set Features commands complete with a status of ‘Invalid Field in Command’ (02h).

© 2026 UNH–IOL

Test 1.26 – Device Self-test Host-Initiated Refresh Operation (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller can properly perform an Device Self-test Host-Initiated Refresh Operation.

References:

TP 4058
NVM Express Base specification 2.2 : 8.1.11

Resource Requirements:

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

Last Modification: July 31, 2025

Discussion: The Device Self-test command is used to start a device self-test operation or abort a device self-test

Operation. The device self-test operation is performed by the controller that the Device Self-test command was submitted to. This test group will test the Host Initiated Refresh Operation, combining both the requirements for completing the operation, as well as the conditions that will cause the command to abort.

Test Setup: See Appendix A

Case 1: Device Self-test Command with STC=3h (OF-FYI, FYI)

Test Procedure:

1. Configure the Testing Station acting as a host to check Bit 4 of the OACS field and Bit 1 of the DSTO field in the Identify Controller Data Structure. If Bit 4 of the OACS field or Bit 1 of the DSTO field are not set to 1h, this test is not applicable.
2. Ensure that no device self-test operation is currently in progress by sending Get Log Page with LID 06h to the controller. Check that Current Device Self-Test Operation field in the response is set to 0h. If the Current Device Self-Test Operation field is not set to 0h, wait for the device self-test operation to complete, and send the Get Log Page with LID 06h again. Repeat until the Current Device Self-Test Operation field is 0h.
3. Configure the NVMe Host to send a Device Self-test command with STC set to 3h (Host-Initiated Refresh).
4. Before the device self-test operation finishes, send a Get Log Page command to LID 06h (Device Self-test). Note that since the device self-test operation can be performed in the background, a command completion may be received before the device self-test operation finishes. Repeat this step until the returned log page indicates that the operation has finished.
5. Configure the NVMe Host to send a Device Self-test command with STC set to 3h (Host-Initiated Refresh).
6. Repeat step 4 for the Device Self-test command configured in step 5.

Observable Results:

1. Using the log page info returned in response to the first Get Log Page command, verify that the NVMe Controller set the Current Device Self-test Status field in the Device Self-test Log to 3h when the operation was in progress.
2. Verify that the Device Self-test commands in step 3 and 5 returned a status of ‘Successful Completion’ (00h).
3. Using the Log Page info returned in response to the second Get Log Page command, verify that the Current Device Self-test Status field in the Device Self-test Log is set to 0h when the operation is no longer in progress.

Case 2: Device Self-Test Options SDSO (OF-FYI, M)

Test Procedure:

1. Configure the Testing Station acting as a host to Check Bit 4 of the OACS field in the Identify Controller Data Structure to determine if the DUT supports the Device Self-test command. If set to 1h, this test is not applicable.

© 2026 UNH–IOL

2. Configure the NVMe Host to issue an Identify Command specifying CNS 01h in order to retrieve the Identify Controller Data Structure. Read the value of DSTO, if bit 1 is set to 1h, then this test is not applicable.

Observable Results:

1. Verify that the RHIRI bit and HIRT bit of the DSTO field in the Identify Controller Data Structure are both cleared to 0h.

Case 3: Host-Initiated Operation Aborted (OF-FYI, FYI)

Test Procedure:

1. Configure the Testing Station acting as a host to check Bit 4 of the OACS field and Bit 1 of the DSTO field in the Identify Controller Data Structure. If Bit 4 of the OACS field or Bit 1 of the DSTO field are not set to 1h, this test is not applicable.
2. Ensure that no device self-test operation is currently in progress by sending Get Log Page with LID 06h to the DUT. Check that Current Device Self-Test Operation field in the response is set to 0h. If the Current Device Self-Test Operation field is not set to 0h, wait for the device self-test operation to complete, and send the Get Log Page with LID 06h again. Repeat until the Current Device Self-Test Operation field is 0h.
3. Configure the NVMe Host and send a Get Log Page command to LID 06h (Device Self-test).
4. Configure the NVMe Host to send Device Self-test command with STC set to 3h (Host-Initiated Refresh).
5. Configure the NVMe Host to perform a Controller Level Reset.
6. Once a command completion is received for the Device Self-test command, send a Get Log Page command to LID 06h (Device Self-test). Repeat until the log page indicates that the Device Self-test Operation is no longer in progress.

Observable Results:

1. Verify that the first Device Self-test command, with STC=3h, completes with status 2h aborted by a Controller Level Reset.
2. Verify that the Controller Level Reset completes successfully.
3. Compare the Device Self-test Log received in steps 3 and 6 and verify that the second received Device Self-test Log contains a new log entry in the Newest Self-test Result Data Structure in the Device Self-test Log.
4. Verify that the second received Device Self-test Log has the Current Device Self-test Status field set to 0h.

Sub-Cases: For each of the following subcases of test 1.26, follow the test procedure of test case 3 except for the modified steps as listed below:

1. Controller Level Reset

Sub-case Procedure:

1. Follow steps as described in test case 1.26.3.

2. Sanitize Operation

Sub-case Procedure:

1. Replace step 5 from the test case with the following steps:
2. Check SANICAP field in the Identify Controller Data Structure. If SANICAP is set to 0h, this test is not applicable.
3. Configure the Testing Station to issue a Sanitize Command for a supported operation type as indicated as supported in the SANICAP field: Overwrite, Block Erase, and Crypto Erase. Repeat test procedure for each supported Operation type.
4. Continually issue a Get Log Page for Sanitize State Log until SSTAT is set to 001b.
5. Verify that the Sanitize Operation completed by performing a Get Log on the Sanitize Status log page Log Identifier 81h and check the Sanitize Progress (SPROG) field.

Sub-case Specific Observable Results:

1. Replace observable result 1 and 2 from the main test case with the following:

© 2026 UNH–IOL

2. Verify that the first Device Self-test command, with STC=3h, completed with status ‘Operation aborted due to a Sanitize Operation’ (9h) indicated in the Self-test Result Data Structure returned in response to the Get Log Page request.
3. Verify that the Sanitize completes successfully.

3. Format Command with NSID=FFFFFFFh

Sub-case Procedure:

5. Before the Device Self-test Operation completes in the background, send a Format NVM Command with NSID=FFFFFFFFh.

Sub-case Specific Observable Results:

1. Verify that the first Device Self-test command, with STC=3h, completes with status ‘aborted due to the processing of a Format NVM command’ (4h).
2. Verify that the Format NVM command completes successfully.

4. Device Self-test Operation STC=Fh

Sub-case Procedure:

5. Before the Device Self-test Operation completes in the background, send a Format NVM Command with NSID=FFFFFFFFh.

Sub-case Specific Observable Results:

1. Verify that the first Device Self-test command, with STC=3h, completes with status ‘Aborted by a Device Self-test command’ (1h).
2. Verify that the second Device Self-test command, with STC=Fh, completes successfully.

© 2026 UNH–IOL

Group 2: NVM Command Set

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

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

© 2026 UNH–IOL

Test 2.1 – Compare Command (OF-FYI, M)

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

References:

Old Ref: [1] NVMe Specification 6.6, 8.3.1.4, 8.3.1.4, NVMe v1.3 ECN 001
NVM Express NVM Command Set Specification 1.0a : 2.1.3.1, 3.2.1, 5.2.2.4

Resource Requirements:

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

Last Modification: April 8, 2019

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 (OF-FYI, M)

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 (OF-FYI, M)

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 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).

© 2026 UNH–IOL

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

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 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 (OF-FYI, M)

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 (OF-FYI, M)

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 (OF-FYI, M)

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.

© 2026 UNH–IOL

Observable Results:

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

Case 7: PRCHK Non-zero (OF-FYI, 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. 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 1h, and PRCHK Bit 00 set to 1h, to the controller in order to write a known sequence to the LBA 0000h. If the Namespace is formatted with PI Type 3 and the controller returns Invalid Protection Information, skip to step 5.
4. Configure the NVMe Host to issue a Compare command to the controller with PRACT set to 0, and PRCHK Bit 00 set to 1h, 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 1h, and again with PRCHK Bit 02 set to 1h. Do not skip on PI Type 3.

Observable Results:

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

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 2.1.7 and expect the same observable results.

1. **Separate Buffer**
2. **Extended Buffer**

Case 8: NSID=FFFFFFFFh (OF-FYI, M)

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.

Possible Problems: None known.

© 2026 UNH–IOL

Test 2.2 – Dataset Management Command (OF-FYI, M)

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

References:

Old Ref: [1] NVMe Specification 6.7
NVM Express NVM Command Set Specification 1.0a : 3.2.3

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.

The command uses Command Dword 10 and Command Dword 11 fields. 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, the SGL Entry 1 field is used. All other command specific fields are reserved.

The data that the Dataset Management command provides is a list of ranges with context attributes. Each range consists of a starting LBA, a length of logical blocks that the range consists of and the context attributes to be applied to that range. The definition of the Dataset Management command Range field is specified in Table 5. The maximum case of 256 ranges is shown.

Table 5 – Dataset Management - Range Definition

Range	Byte	Field
Range 0	03:00	Context Attributes
	07:04	Length in logical blocks
	15:08	Starting LBA
Range 1	19:16	Context Attributes
	23:20	Length in logical blocks
	31:24	Starting LBA
…		
Range 255	4083:4080	Context Attributes
	4087:4084	Length in logical blocks
	4095:4088	Starting LBA

Test Setup: See Appendix A.

Case 1: Basic Operation (OF, M)

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.

© 2026 UNH–IOL

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 (OF, M)

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.
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 3: NR Value is Maximum (OF-FYI, M)

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'.

© 2026 UNH–IOL

Observable Results:

1. Verify that the command completes successfully.

Case 4: Correct Range Deallocated (OF, M)

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 (OF, M)

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 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:

© 2026 UNH–IOL

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 (OF-FYI, M)

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 (OF-FYI, 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 ‘Abort Command Limit Exceeded’ (03h).

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

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 ‘Abort Command Limit Exceeded’ (03h).

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

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.

© 2026 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 does not return status ‘Abort Command Limit Exceeded’ (03h).

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

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 ‘Abort Command Limit Exceeded’ (03h).

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

Test Procedure:

1. Check the ONCS Bit 2 of the Identify Controller Data structure. If ONCS Bit 2 is set to 1h, 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 ‘Abort Command Limit Exceeded’ (03h).

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

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 total number of logical blocks in the command exceeds the DMSL value.

Observable Results:

1. Verify that the Dataset Management Command returns status ‘Abort Command Limit Exceeded’ (03h).

Possible Problems: None known.

© 2026 UNH–IOL

Test 2.3 – Read Command (OF-FYI, M)

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

References:

Old Ref: [1] NVMe Specification 6.9, 9.4, NVMe v1.3 ECN 001
NVM Express NVM Command Set Specification 1.0a : 3.2.4

Resource Requirements:

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

Last Modification: April 29, 2019

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 (OF, M)

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.
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 (OF, M)

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 (OF, M)

Test Procedure:

© 2026 UNH–IOL

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 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 (OF, M)

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 (OF, M)

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 (OF, M)

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 (OF, M)

Test Procedure:

© 2026 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 Read command to the controller to the LBA 0000h.
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 (OF, M)

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.
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 Read, LR=1, FUA=0 (OF, M)

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.
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 Read, LR=1, FUA=1 (OF, M)

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.
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.

© 2026 UNH–IOL

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

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 in order to write a known data pattern to one LBA on the NVMe Device.
3. 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.

Case 12: Type 1 Protection, Invalid ILBRT (OF-FYI, FYI)

Test Procedure:

1. Configure the Testing Station to format a namespace with Type 1 protection.
2. Configure the NVMe Host to issue a Write command to the controller with bit 0 of the PRCHK field set to 1 and an invalid ILBRT field. A random value for ILBRT can be used to ensure that it does not match the least significant four bytes of the SLBA field.

Observable Results:

1. Verify that the Write command is aborted with status Invalid Protection Information.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 2.3.12 and expect the same observable results.

1. **Separate Buffer**
2. **Extended Buffer**

Case 13: Type 1 Protection, Invalid EILBRT (OF-FYI, FYI)

Test Procedure:

1. Configure the Testing Station to format a namespace with Type 1 protection.
2. Perform a Write command to SLBA.
3. Configure the NVMe Host to issue a Read command to the controller with bit 0 of the PRCHK field set to 1h and an invalid EILBRT field. A random value for EILBRT can be used to ensure that it does not match the least significant four bytes of the SLBA field.

Observable Results:

1. Verify that the READ command is aborted with status Invalid Protection Information.

Possible Problems: None Known.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 2.3.13 and expect the same observable results.

1. **Separate Buffer**
2. **Extended Buffer**

© 2026 UNH–IOL

Test 2.4 – Write Command (OF-FYI, M)

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

References:

Old Ref: [1] NVMe Specification 6.14, 9.4
NVM Express NVM Command Set Specification 1.0a : 3.2.6

Resource Requirements:

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

Last Modification: November 14, 2017

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 (OF, M)

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 (OF, M)

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 (OF, M)

Test Procedure:

© 2026 UNH–IOL

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 (OF, M)

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:

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 (OF, M)

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 (OF, M)

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 (OF, M)

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.

© 2026 UNH–IOL

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 (OF, M)

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 (M)

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 (OF, M)

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.

© 2026 UNH–IOL

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

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.
3. 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.

Observable Results:

1. Verify that the Write command did not complete successfully and the known data pattern was not read back in the subsequent READ operation.

Possible Problems: None Known.

© 2026 UNH–IOL

Test 2.5 – Write Uncorrectable Command (OF, M)

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

References:

Old Ref: [1] NVMe Specification 6.15
NVM Express NVM Command Set Specification 1.0a : 3.2.7

Resource Requirements:

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

Last Modification: August 29, 2017

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 (OF, M)

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.
4. Configure the NVMe Host to issue a Write command for the LBA on which the Write Uncorrectable command was performed in order to clear the invalid logical block status.
5. Repeat step 3.

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 first Read command indicates *Unrecovered Read Error* status.
3. Verify that the completion queue entry returned for the second Read command indicates Success status.

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

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:

© 2026 UNH–IOL

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.

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

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 (OF, M)

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 (OF, M)

Test Procedure:

1. Check the WUSL fields. If this field is set to a non-zero value this test is not applicable.
2. Check that the ONCS field bit 1 is set to 1 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.

© 2026 UNH–IOL

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.
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 (OF-FYI, M)

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.
4. Configure the NVMe Host to issue a Read command for the LBA on which the Write Uncorrectable command was performed.
5. Configure the NVMe Host to issue a Write command for the LBA on which the Write Uncorrectable command was performed in order to clear the invalid logical block status.
6. Repeat step 3.

Observable Results:

1. Verify that the Write Uncorrectable command does not complete successfully.
2. Verify that the completion queue entry returned for both Read commands indicate Success status.

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

Test Procedure:

1. Check the ONCS bit 1, if set to 1 continue, if set to zero, then check the WUSL field. If WUSL is set to 0h this test 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 command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status of ‘Invalid Field in Command’ (02h) for the Write Uncorrectable command.

Possible Problems: Case 5 is updated per ECN 003 of NVMe Specification v1.2.1. This ECN alters behavior expected in earlier versions.

© 2026 UNH–IOL

Test 2.6 – Flush Command (OF, M)

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

References:

Old Ref: [1] NVMe Specification 6.8, TP 4035
NVM Express Base Specification 2.0a : 7.1

Resource Requirements:

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

Last Modification: April 24, 2019

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 (OF, M)

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 (OF, M)

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 (OF, M)

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.

© 2026 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.

Case 4: Flush when VWC Not Present (OF-FYI, M)

Test Procedure:

1. Check the VWC field of the Identify Controller Data Structure. If Bit 0 of the VWC field is set to 1h, this test is not applicable.
2. Perform a Get Log Page for the Sanitize Status Log Page, LID=81h and verify that the Sanitize Status Log Page has SSTAT set to 001b, and SPROG=FFFFh, to confirm that a Sanitize operation is not in progress.
3. Configure the NVMe Host to issue a Flush command to the NVMe Controller.

Observable Results:

1. Verify that the Flush command completes with status 'Success'.

Possible Problems: None known.

Case 5: FDP Identify Controller, VWCP=1 & VCE=0, Persistent Reset (OF-FYI, M)

Test Procedure:

1. Perform an Identify command for CNS 01h, Identify Controller. Record the bit 0 of the VWC field. If Volatile Write Cache Present (Bit 0) is cleared to 0h, this test is not applicable.
2. Perform a Get-Feature command for FID 06h. If Write Cache Enable (WCE) is set to 1h, then perform a Set-Feature command for FID 06h to clear WCE to 0h.
3. Perform a Write command to the controller in order to write a known data pattern to one LBA.
4. Perform a Read command to the controller in order to read from the same LBA which was written to in step 3.
5. Perform a controller level Reset.
6. Perform a Read command to the controller in order to read from the same LBA which was written to in step 3.

Observable Results:

1. Verify all commands executed successfully.
2. Ensure that the data written in step 3 was persistent after the Controller Reset.

Case 6: Flush With Volatile Write Cache and Power Cycle (M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command for CNS 01h, Identify Controller. Record the MDTS field converted to a number of bytes (MDTS Bytes) and VWC fields. If bit 0 of the VWC field (Volatile Write Cache Present) is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command for CNS 0h, Identify Namespace. Record the NSZE field and the LBADS field for the current format.
3. Configure the NVMe Host to issue a Get Feature command for FID 06h. If the Returned Volatile Write Cache Enable field is cleared to 0h, Configure the NVMe Host to issue a Set Feature command for FID 06h to set the WCE field to 1h.
4. Configure the NVMe Host to issue a Get Log command for LID 02h, Smart/Health Information Log, and record the value of the Power Cycles field.
5. Configure the NVMe Host to issue a Get Feature command for FID 07h and record the returned Number of I/O Submission Queues Allocated (NCQA).
6. Configure the NVMe host to issue NCQA Create I/O Submission Queue and Create I/O completion queue commands to create the maximum number of I/O Submission and Completion Queues.
7. Configure the NVMe host to issue 1 Write command on each I/O Submission Queue specifying:
    a. Incrementing SLBA offsets.

© 2026 UNH–IOL

   b. NLB equal to NSZE / NCQA rounded down to the nearest integer. If NSZE / NCQA is equivalent to a number of bytes greater than MDTS Bytes, instead specify a NLB value of MDTS Bytes / LBADS rounded down to the nearest integer.
   c. A DPTR specifying NLB * LBADS bytes of AAh.
8. Configure the NVMe Host to issue a Flush command.
9. Unceremoniously power cycle the device.
10. Configure the NVMe Host to issue a Get Log command for LID 02h, Smart/Health Information Log, and record the value of the Power Cycles field.
11. Configure the NVMe Host to issue a Read command to read from the target buffer of each successful Write command issued in step 6.

Observable Results:

1. Verify the Flush command in step 7 completed successfully.
2. Verify the Get Log command in steps 4 and 10 completed successfully, and that the value for the Power Cycle field incremented by 1 in the second log.
3. Record the number of Write commands in step 6 which completed successfully.
4. Verify each Read command in step 9 completed successfully, and record the total number of bytes set to a value of AAh in their returned data buffer.
5. Verify that the number of bytes set to AAh is greater than or equal to the number of bytes written by the Write commands that completed successfully in step 6.

Possible Problems: External equipment is required to perform the power cycle in this test without powering down the host.

© 2026 UNH–IOL

Test 2.7 – Write Zeroes Command (OF, M)

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

References:

Old Ref: [1] NVMe Specification 6.16
NVM Express NVM Command Set Specification 1.0a : 3.2.8

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 (OF, M)

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 command to the controller specifying a particular LBA on the NVMe Device. The written data should be either all 1's, or an alternating pattern of 1's and 0's.
3. Configure the NVMe Host to issue a Write Zeroes command to the controller specifying the same LBA on the NVMe Device which was previously written to.
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 2: SLBA Out of Range, NLB Valid (OF, M)

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.

© 2026 UNH–IOL

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

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 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 (OF, M)

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 (OF, M)

Test Procedure:

1. Check the DMRL, DMRSL, and DMSL fields. If these fields are set to non-zero values 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 Write command to the controller specifying a particular LBA on the NVMe Device. The written data should be either all 1’s, or an alternating pattern of 1’s and 0’s.
5. Configure the NVMe Host to issue a Read command to the controller specifying the same LBA for which the previous Write command was performed.
6. Configure the NVMe Host to issue a Write Zeroes command the same LBA on the NVMe Device as was written in Step 1. NLB should be greater than MDTS.
7. Configure the NVMe Host to issue Read commands to the controller specifying the same LBAs for which the Write Zeroes command was performed. Ensure that the NVMe Host issues Read commands to all LBAs affected by the Write Zeroes 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 for the command which should be ‘Successful’.
2. Verify that the data returned by the controller for the Read command after the Write command returns the data written in Step 1.

© 2026 UNH–IOL

3. 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 (OF, M)

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 command to the controller specifying a particular LBA on the NVMe Device. The written data should be either all 1's, or an alternating pattern of 1's and 0's.
3. Configure the NVMe Host to issue a Write Zeroes command to the controller specifying the same LBA on the NVMe Device which was previously written to.
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 7: SLBA In Range, NLB Valid, LR=1, FUA=0 (OF, M)

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 command to the controller specifying a particular LBA on the NVMe Device. The written data should be either all 1's, or an alternating pattern of 1's and 0's.
3. Configure the NVMe Host to issue a Write Zeroes command to the controller specifying the same LBA on the NVMe Device which was previously written to.
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 (OF, M)

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 command to the controller specifying a particular LBA on the NVMe Device. The written data should be either all 1's, or an alternating pattern of 1's and 0's.
3. Configure the NVMe Host to issue a Write Zeroes command to the controller specifying the same LBA on the NVMe Device which was previously written to.
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.

© 2026 UNH–IOL

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 (OF-FYI, M)

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 Write command to the controller specifying the namespace with end to end data protection on the NVMe Device. The written data should be either all 1's, or an alternating pattern of 1's and 0's.
5. Configure the NVMe Host to issue a Write Zeroes command to the controller specifying the same LBA on the NVMe Device which was previously written to, but the PRCHK value within the PRINFO field is non-zero.
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.
2. Verify that the data returned by the controller for the Read command after the Write Zeroes command returns the data pattern written in step 4.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 2.7.9 and expect the same observable results.

1. **Separate Buffer**
2. **Extended Buffer**

Case 10: NSID=FFFFFFFFh, SLBA In Range, NLB Valid, LR=0, FUA=0 (OF-FYI, M)

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 Write command to the controller specifying a particular LBA on the NVMe Device. The written data should be either all 1's, or an alternating pattern of 1's and 0's.
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 written to.
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.

© 2026 UNH–IOL

2. Verify that the data returned by the controller for the Read command after the Write Zeroes command returns the same data as was previously written.

Case 11: NLB greater than WZSL and Non-MDTS Command Size Limits Supported (OF-FYI, M)

Test Procedure:

1. Check the DMRL, DMRSL, and DMSL fields. If these fields are 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'.

Possible Problems: Case 5 is updated per ECN 003 of NVMe Specification v1.2.1. This ECN alters behavior expected in earlier versions.

Case 12: Namespace Zeroes Supported Deallocates Non-Specified LBAs (OF-FYI, FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data Structure. If the Write Zeroes Command is not supported or the Namespace Zeroes Support bit is cleared to 0, then this test is not applicable.
2. Check the DLFEAT value in the Identify Namespace data structure, if set to a value other than 001b, this test is not applicable.
3. Configure the NVMe Host to write a known data pattern to the first LBA on the namespace.
4. Send a Read command to the same LBA that was specified in the previous step.
5. Configure the NVMe Host to issue a Write Zeroes command with the Namespace Zeroes (NSZ) and Deallocate bits set to 1.
6. Send a Read command to the same LBA that was specified in step 3.

Observable Results:

1. Verify that all commands completed successfully.
2. Verify that the Read command in step 4 returned the data pattern written in step 3.
3. Verify that the LBA that was the target of the Write Zeroes command in step 5 contains no user data and is cleared to 0 (i.e. deallocated).

Case 13: WZDSL Greater than WZSL (OF-FYI, FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data Structure. Record the value of bit 3.
2. Configure the NVMe Host to issue an Identify command with CNS 06h and CSI 00h to get the NVM Command Set I/O Command Set Specific Identity Controller Data Structure. If the value of WZSL is cleared to 0 and bit 3 of ONCS is cleared to 0, this test is not applicable. Also if the value of Write Zeroes With Deallocate Size Limit (WZDSL) is cleared to 0h, then this test is not applicable, otherwise, record the value.
3. Configure the NVMe Host to issue a Write Zeroes command with the Deallocate bit set to 1, and NLB greater than WZDSL.
4. Configure the NVMe Host to issue a Write Zeroes command with the Deallocate bit cleared to 0, and NLB greater than WZSL.

Observable Results:

© 2026 UNH–IOL

1. Verify that after the completion of the Write Zeroes commands, 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' in each command.
2. Verify all other commands completed successfully.

Case 14: Namespace Zeroes Requires Deallocate (OF-FYI, FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data Structure. Record the value of NSZ and ONCS bit 3.
2. Configure the NVMe Host to issue an Identify command with CNS 06h and CSI 00h to get the NVM Command Set I/O Command Set Specific Identity Controller Data Structure. If the value of WZSL is cleared to 0 and bit 3 of ONCS is cleared to 0 and NSZ is cleared to 0, this test is not applicable.
3. Configure the NVMe Host to issue a Write Zeroes command with the Namespace Zeroes (NSZ) bit set to 1, and the Deallocate bit cleared to '0'.

Observable Results:

1. Verify that after the completion of the Write Zeroes 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 15: Write Zeroes with Namespace Zeroes Ignores SLBA (OF-FYI, FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data Structure. Record the value of NSZ and ONCS bit 3. If NSZ is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command with CNS 06h and CSI 00h to get the NVM Command Set I/O Command Set Specific Identity Controller Data Structure. If the value of WZSL is cleared to 0 and bit 3 of ONCS is cleared to 0, this test is not applicable.
3. Configure the NVMe Host to write a known data pattern to the first LBA on the namespace.
4. Configure the NVMe Host to issue a Write Zeroes command with the Namespace Zeroes (NSZ) bit set to 1, and the Deallocate bit set to '1', and the SLBA field set to the address of a Logical Block that is not the first Logical Block in the target namespace (i.e. configure the command to send a Write Zeroes command with SBLA set to the address of the 2nd LBA in the namespace.).
5. Configure the NVMe Host to issue a Read command to target the first LBA which was written to in step 3

Observable Results:

1. Verify that after the completion of the Write Zeroes command, the controller posts a completion queue entry to the associated I/O Completion Queue indicating the status for the command of 'Successful Completion'.
2. Verify that all Logical Blocks in the namespace have been cleared to 0.

Case 16: Write Zeroes with Namespace Zeroes Ignores SLBA (OF-FYI, FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data Structure. If the MAXWZD is cleared to '0h', then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command with CNS 06h and CSI 00h to get the NVM Command Set I/O Command Set Specific Identity Controller Data Structure. Record the values of WZSL and WZDSL.

Observable Results:

1. Verify that WZSL is a non-zero value.
2. Verify that WZDSL is greater than WZSL.

© 2026 UNH–IOL

Case 17: Write Zeroes Command with Deallocate Data Size > Write Zeroes Command Data Size with no Deallocate (OF-FYI, FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data Structure. If the MAXWZD is cleared to ‘0h’, then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command with CNS 06h and CSI 00h to get the NVM Command Set I/O Command Set Specific Identity Controller Data Structure. Record the values of WZSL and WZDSL. If bit 3 of ONCS is cleared to ‘0’ and WZSL has a value 0h, then this test is not applicable. If WZDSL is cleared to ‘0h’, then this test is not applicable.
3. Configure the NVMe Host to issue a Write Zeroes command with NLB set to the value of WZSL, and the Deallocate bit cleared to ‘0h’.
4. Configure the NVMe Host to issue a Write Zeroes command with NLB set to the value of WZDSL, and the Deallocate bit cleared to ‘0h’.
5. Configure the NVMe Host to issue a Write Zeroes command with NLB set to the value of WZDSL and Deallocate bit set to ‘1h’, which is expected to be a larger value than the previous step.

Observable Results:

1. Verify that the Write Zeroes commands in steps 3 and 5 complete successfully.
2. Verify the Write Zeroes command in step 4 completes with a status of ‘Invalid Field in Command’ (02h).

© 2026 UNH–IOL

Test 2.8 – Atomicity Parameters (OF, M)

Purpose: To verify that an NVMe Controller properly sets the Atomicity Parameters.

References:

Old Ref: [1] NVMe Specification 6.4
NVM Express Base Specification 2.0a : 3.4.3
NVM Express NVM Command Set Specification 1.0a : 2.1.4

Resource Requirements:

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

Last Modification: November 28, 2017

Discussion: The NVMe specification sets certain restrictions on the Atomicity Parameters relative to one another, as reported in the Identify Controller and Identify Namespace data structures. This tests seeks to ensure that those restrictions are followed. These restrictions apply to the AWUN, AWUPF, ACWU, NAWUN, NAWUPF, NACWU, NABSN, NABO, NABSPF values. Support for Atomic Writes is indicated in the NSFEAT and NAWUN fields. If the DUT does not support Atomic Writes this test is not applicable and does not need to be performed.

Test Setup: See Appendix A.

Case 1: NVM Command Set Supported (OF, M)

Test Procedure:

1. For each namespace in the NVM subsystem, configure the NVMe Host to issue an Identify command specifying CNS value 00h to the controller in order to receive back an Identify Namespace data structure for the specified namespace.
2. Check that Bit 37 of CAP.CSS is set to 1, otherwise this test is not applicable.
3. Configure the NVMe Host to issue an Identify command specifying CNS value 01h to the controller in order to receive back an Identify Controller data structure.
4. Parse the received data structures for the AWUN, AWUPF, ACWU, NAWUN, NAWUPF, NACWU, NABSN, NABO, NABSPF values.

Observable Results:

1. Determine if NABSN is set to a non-zero value. If not, this test case is not applicable.
2. Verify the following:
    a. AWUPF <= AWUN
    b. NAWUN >= AWUN
    c. NAWUPF >= AWUPF
    d. NAWUPF <= NAWUN
    e. NACWU >= ACWU
    f. NABSN >= NAWUN
    g. NABO <= NABSN
    h. NABO <= NABSPF
    i. NABSPF >= NAWUPF

Case 2: NVM Command Set Not Supported (OF-FYI, FYI)

Test Procedure:

1. Check that Bit 37 of CAP.CSS is set to 0h, otherwise this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 1Ch to the controller in order to receive back an Identify I/O Command Set data structure. If the Zoned Namespace Command Set (CSI 02h)

© 2026 UNH–IOL

is supported, this test is not applicable, since the AWUN, AWUPF, and ACWU fields are also applicable to namespaces associated with the Zoned Namespace Command Set.

3. Configure the NVMe Host to issue an Identify command specifying CNS value 01h to the controller in order to receive back an Identify Controller data structure.
4. Parse the received data structures for the AWUN, AWUPF, ACWU.

Observable Results:

1. Verify that AWUN, AWUPF, and ACWU are set to 0h.

Possible Problems: None known.

© 2026 UNH–IOL

Test 2.9 – AWUN/NAWUN (M)

Purpose: To verify that an NVMe Controller properly uses its Atomicity Parameters.

References:

Old Ref: [1] NVMe Specification 6.4.1
NVM Express NVM Command Set Specification 1.0a : 2.1.4.1

Resource Requirements:

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

Last Modification: August 29, 2017

Discussion: AWUN/NAWUN control the atomicity of command execution in relation to other commands. They impose inter-command serialization of writing of blocks of data to the NVM and prevent blocks of data ending up on the NVM containing partial data from one new command and partial data from one or more other new commands.

If a write command is submitted with size less than or equal to the AWUN/NAWUN value and the write command does not cross an atomic boundary, then the host is guaranteed that the write command is atomic to the NVM with respect to other read or write commands. If a write command is submitted with size greater than the AWUN/NAWUN value or crosses an atomic boundary, then there is no guarantee of command atomicity. AWUN/NAWUN does not have any applicability to write errors caused by power failure or other error conditions.

Test Setup: See Appendix A.

Case 1: Atomic Boundaries Not Supported (NABSN/NABSPF = 0) (M)

Test Procedure:

1. Ensure that both NABSN and NABSPF are set to 0. If not, this test case is not applicable and Case 2 below should be performed.
2. Configure the NVMe Host to issue 2 Write Commands, each with a length equal to 4 logical blocks. (i.e. if the logical block size is 512 bytes, the Write would be 2K in length. If the logical block size is 4K, the length of the Write would be 16K). The first command (Command A) will write Logical Blocks 0-3, with an FFFFh pattern. The second command (Command B) will write to Logical Blocks 1-4, with an AAAAh pattern.
3. Repeat for all supported Namespaces.

Observable Results:

1. Verify that one of the following has occurred, any other outcome is a failure:
    a. LBA 0-3 have an FFFFh pattern and LBA 4 has an AAAAh pattern.
    b. LBA 0 has an FFFFh pattern and LBA 1-4 has an AAAAh pattern.

Case 2: Atomic Boundaries Supported (NABSN != 0) (M)

Test Procedure:

1. Determine if NABSN is set to a non-zero value. If not, this test case is not applicable and Case 1 above should be performed.
2. Configure the NVMe Host to issue 2 Write Commands, each with a length of half of NABSN. The first command (Command A) will write 4x sets of length NABSN, with an FFFFh pattern. The second command (Command B) will write 4x sets of length NABSN with an AAAAh pattern. Command B will be offset from Command A by half of NABSN. In total 5 sets of data with length $^1/_2$ NABSN will be written.
3. Repeat for all supported Namespaces.

Observable Results:

1. Verify that one of the following has occurred, any other outcome is a failure:
    a. The first 4x NABSN of data is a FFFFh pattern and the last 1x NABSN of data is an AAAAh pattern.

© 2026 UNH–IOL

b. The first 1x NABSN of data is a FFFFh pattern and the last 4x NABSN of data is an AAAAh pattern.

Possible Problems: For Case 2 above, NABSPF could be set to a non-zero value indicating support for Atomic Boundaries during a Power Failure. However, test capability has not been developed for testing Atomic Writes during Power Failure, so only the Normal condition is tested.

Case 3: Multiple Atomicity, Atomic Boundaries Supported (NABSN != 0) (FYI)

Test Procedure:

1. Determine if all Namespaces support Multiple Atomicity by performing an Identify on the Namespace Data Structure and verify the Multiple Atomicity Mode (MAM) is set to '1', if not this test is not applicable.
2. Determine if NABSN is set to a non-zero value. If not, this test case is not applicable.
3. Create two namespaces that are contiguous, and both support Multiple Atomicity Mode.
4. Configure the NVMe Host to issue 1 Write Commands with a length of double NABSN, and starting at the end of the first namespace. The write command (Command A) will write 4x sets of length NABSN, with an FFFFh pattern that starts towards the end of the first.
5. Perform a read of the written data across each namespace boundary.

Observable Results:

1. Verify that one of the following has occurred, any other outcome is a failure:
    a. That the write occurred correctly across namespace boundaries and that the data is correct

Possible Problems: NABSPF could be set to a non-zero value indicating support for Atomic Boundaries during a Power Failure. However, test capability has not been developed for testing Atomic Writes during Power Failure, so only the Normal condition is tested.

© 2026 UNH–IOL

Test 2.10 – AWUPF/NAWUPF (IP)

Purpose: To verify that an NVMe Controller properly uses its Atomicity Parameters when a power failure occurs.

References:

Old Ref: [1] NVMe Specification 6.4.2
NVM Express NVM Command Set Specification 1.0a : 2.1.4.2

Resource Requirements:

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

Last Modification: June 13, 2016

Discussion: AWUPF and NAWUPF indicate the behavior of the controller if a power fail or other error condition interrupts a write operation causing a torn write. A torn write is a write operation where only some of the logical blocks that are supposed to be written contiguously are actually stored on the NVM, leaving the target logical blocks in an indeterminate state in which some logical blocks contain original data and some logical blocks contain new data from the write operation.

If a write command is submitted with size less than or equal to the AWUPF/NAWUPF value and the write command does not cross an atomic boundary the controller guarantees that if the command fails due to a power failure or other error condition, then subsequent read commands for the logical blocks associated with the write command shall return one of the following:

- All old data (i.e. original data on the NVM in the LBA range addressed by the interrupted write), or
- All new data (i.e. all data to be written to the NVM by the interrupted write)

If a write command is submitted with size greater than the AWUPF/NAWUPF value or crosses an atomic boundary, then there is no guarantee of the data returned on subsequent reads of the associated logical blocks.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the DUT to use AWUPF/NAWUPF values to be greater than the size of 2 logical blocks.
2. Perform a READ operation to Logical Blocks 0-1.
3. Configure the NVMe Host to issue a Write Command, with a length equal to 2 logical blocks. (i.e. if the logical block size is 512 bytes, the Write would be 1K in length. If the logical block size is 4K, the length of the Write would be 8K). The command (Command A) will write Logical Blocks 0-1, with an AAAAh pattern. Before command completion, cause a Power Failure event.
4. Allow the PCIe link to reset, and for the NVMe Controller to return to the enabled state.
5. Perform a READ operation to Logical Blocks 0-1.

Observable Results:

1. Verify that one of the following has occurred, any other outcome is a failure:
    a. Contents of LBA 0-1 have an AAAAh pattern.
    b. Contents of LBA 0-1 are unchanged.

Possible Problems: Tools for reliably creating a power failure during a WRITE operation are not available. This test will remain an FYI test until such tools are available.

© 2026 UNH–IOL

Test 2.11 – Verify Command (OF-FYI, M)

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

References:

Old Ref: [1] NVMe Specification TP 4030
NVM Express NVM Command Set Specification 1.0a : 3.2.5

Resource Requirements:

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

Last Modification: April 24, 2019

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 (OF-FYI, 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.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 2.11.1 and expect the same observable results.

1. **Separate Buffer**
2. **Extended Buffer**
3. **MS = 0**

Case 2: PRACT = 1 (OF-FYI, 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.
3. Repeat for all supported PRINFO values using the same protection information as the namespace was formatted and PRACT=1.

Observable Results:

1. Verify command completes with status ‘Invalid Field in Command’ (02h) (excluding Verify formatted to PI type 3 and PRCHK=1).

© 2026 UNH–IOL

2. For devices that are version 2.0 or later the Verify has a return status of 'Invalid Protection Information' (81h), For devices earlier than version 2.0 has a return status of 'Invalid Protection Information' (81h) or 'Invalid Field in Command' (02h).

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 2.11.2 and expect the same observable results.

1. **Separate Buffer**
2. **Extended Buffer**
3. **MS = 0**

Sub-case Procedure:

3. The PRINFO field is ignored when MS = 0, and therefore the test is unable to format a namespace to use end-to-end protection information. Verify the controller returns a status of Success instead of 'Invalid Field in Command' (02h).

Case 3: NSID=FFFFFFFFh (OF-FYI, M)

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.
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.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 2.11.3 and expect the same observable results.

1. **Separate Buffer**
2. **Extended Buffer**
3. **MS = 0**

Case 4: Command Size Limits (OF-FYI, FYI)

Test Procedure:

1. Check ONCS bit 7, if this bit is set to 1h 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' (02h).

Possible Problems: None known.

© 2026 UNH–IOL

Test 2.12 – Fused Operations (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller can properly execute Fused Operations.

References:

Old Ref: [1] NVMe Specification 6.2
NVM Express NVM Command Set Specification 1.0a : 3.4.2

Resource Requirements:

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

Last Modification: November 13, 2019

Discussion: The Compare and Write fused operation compares the contents of the logical block(s) specified in the Compare command to the data stored at the indicated LBA range. If the compare is successful, then the LBA range is updated with the data provided in the Write command. If the Compare operation is not successful, then the Write operation is aborted with a status of Command Aborted due to Failed Fused Command and the contents in the LBA range are not modified. If the Write operation is not successful, the Compare operation completion status is unaffected.

Test Setup: See Appendix A.

Case 1: Supported Fused Operation (OF-FYI, M)

Test Procedure:

1. Check the FUSES field of the Identify Controller Data structure to determine if the DUT supports Fused Operations. If fused operations are not supported this test is not applicable.
2. Configure the NVMe Host to issue a Compare command with the FUSE field set to 01b, to indicate the first command in a fused operation followed by a Write command with the FUSE field set to 10b to indicate the second command in a fused operation to the controller.

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 and Write commands complete successfully.

Case 2: Fused Operations Not Supported (OF-FYI, M)

Test Procedure:

1. Check the FUSES field of the Identify Controller Data structure to determine if the DUT supports Fused Operations. If fused operations are supported this test is not applicable.
2. Configure the NVMe Host to issue a Compare command with the FUSE field set to 01b, to indicate the first command in a fused operation followed by a Write command with the FUSE field set to 10b to indicate the second command in a fused operation to the controller.

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 both the Compare and Write commands complete with status ‘Invalid Field in Command’ (02h).

Case 3: Missing Fused Command (OF-FYI, FYI)

Test Procedure:

1. Check the FUSES field of the Identify Controller Data structure to determine if the DUT supports Fused Operations. If fused operations are not supported this test is not applicable.

© 2026 UNH–IOL

2. Verify that the DUT claims support for NVME v1.4 or higher. If not, then this test is not applicable.
3. Configure the NVMe Host to issue a Compare command with the FUSE field set to 01b, to indicate the first command in a fused operation.
4. Configure the NVMe Host to issue Write command with the FUSE field set to 00b, to indicate the second command is not a fused operation. Wait for the command to complete.

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 was aborted with status 'Command Aborted due to Missing Fused Command' (0Ah).

Case 4: LBA Range Mismatch (OF-FYI, M)

Test Procedure:

1. Check the FUSES field of the Identify Controller Data structure to determine if the DUT supports Fused Operations. If fused operations are not supported this test is not applicable.
2. Configure the NVMe Host to issue a Compare command with the FUSE field set to 01b, to indicate the first command in a fused operation followed by a Write command with the FUSE field set to 10b to indicate the second command in a fused operation to the controller. The Write command should indicate a different LBA range than the Compare Command.

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 both the Compare and Write commands complete with status 'Invalid Field in Command' (02h).

Case 5: Unsupported Fused Operation (OF-FYI, M)

Test Procedure:

1. Check the FUSES field of the Identify Controller Data structure to determine if the DUT supports Fused Operations. If fused operations are not supported this test is not applicable.
2. Configure the NVMe Host to issue a Read command with the FUSE field set to 01b, to indicate the first command in a fused operation.

Observable Results:

1. Verify that the Read command is aborted with status 'Invalid Field in Command' (02h).

Possible Problems: None known.

Case 6: Command Aborted due to Failed Fused Command (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. Check ONCS bit 0 for support of the Compare command. If the bit is cleared to 0, then this test is not applicable.
2. Check the Fused Operation Support (FUSES) field bit 0. If the bit is cleared to 0, then this test is not applicable.
3. Configure the NVMe Host to issue a Write command to any logical block with a known data pattern of AAh.
4. Configure the NVMe Host to issue a Fused command of Compare and Write to the same logical block with a different data pattern of 00h for the Compare command

Observable Results:

1. Verify that the Identify command completes successfully.
2. Verify that the Write command from step 3 completes successfully.

© 2026 UNH–IOL

3. Verify that the Write portion of the Fused command returns a status code of ‘Command Aborted due to Failed Fused Command’.

Possible Problems: None known.

© 2026 UNH–IOL

Test 2.13 – Deallocate and Allocate (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller can properly operate depending on the DULBE setting.

References:

Old Ref: [1] NVMe Specification 6.7
NVM Express NVM Command Set Specification 1.0a : 3.2.3.2.1

Resource Requirements:

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

Last Modification: May 11, 2020

Discussion: A logical block that has never been written to, or which has been deallocated using the Dataset Management command, the Write Zeroes command or the Sanitize command is called a deallocated or unwritten logical block.

Test Setup: See Appendix A.

Case 1: Deallocate via Dataset Management then READ, DULBE=0 (OF-FYI, M)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports Dataset Management. If the command is not supported this test is not applicable.
2. Perform an Identify Namespace Data Structure Command and record the value of the DLFEAT field.
3. Perform a Set Feature command for FID 05h, Error Recovery Feature, to set DULBE=0.
4. Perform a Get Feature command for FID 05h, Error Recovery Feature, to ensure that DULBE=0.
5. Configure the NVMe Host to issue a Write command to write a known data pattern to the controller.
6. 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 the previous step.
7. Configure the NVMe Host to issue 2 Read commands to the same LBA range deallocated in the previous step.

Observable Results:

1. Verify that the same data pattern is returned to each Read command.
2. Verify that the Read data returned has all bytes cleared to 0h if bits 2:0 in the DLFEAT field are set to 001b.
3. Verify that the Read data returned has all bytes set to FFh if bits 2:0 in the DLFEAT field are set to 010b.
4. Verify that the Read data returned has 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 2: Deallocate via Dataset Management, DULBE=1 (OF-FYI, FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports Dataset Management. If the command is not supported this test is not applicable.
2. Perform a Set Feature command for FID 05h, Error Recovery Feature, to set DULBE=1.
3. Perform a Get Feature command for FID 05h, Error Recovery Feature, to ensure that DULBE=1.
4. Configure the NVMe Host to issue a Write command to write a known data pattern to the controller.
5. 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 the previous step.
6. Configure the NVMe Host to issue the following commands to the same LBA range deallocated in the previous step.
    a. Read
    b. Verify (if supported) (Check Bit 7 of ONCS field)
    c. Compare (if supported) (Check bit 0 of ONCS field)

© 2026 UNH–IOL

Observable Results:

1. Verify that the Read, Verify (if supported), and Compare (if supported) commands fail with status Deallocated or Unwritten Logical Block, 87h.

Case 3: Allocate via Write (OF-FYI, M)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports Dataset Management. If the command is not supported this test is not applicable.
2. Perform a Set Feature command for FID 05h, Error Recovery Feature, to set DULBE=0.
3. Perform a Get Feature command for FID 05h, Error Recovery Feature, to ensure that DULBE=0.
4. Configure the NVMe Host to issue a Dataset Management command with the Attribute – Deallocate (AD) field set to '1'.
5. Perform an Identify Namespace Data Structure Command and record the value of the NUSE field.
6. Configure the NVMe Host to a Write command to the same LBA range deallocated in the previous steps.
7. Perform an Identify Namespace Data Structure Command and record the value of the NUSE field.

Observable Results:

1. Verify that the value reported in the NUSE field in the second Identify Namespace Data Structure Command is greater than the value reported in the first Identify Namespace Data Structure Command.

Case 4: Allocate via Write Uncorrectable (OF-FYI, FYI)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports Dataset Management. If the command is not supported this test is not applicable.
2. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports Write Uncorrectable. If the command is not supported this test is not applicable.
3. Perform a Set Feature command for FID 05h, Error Recovery Feature, to set DULBE=0.
4. Perform a Get Feature command for FID 05h, Error Recovery Feature, to ensure that DULBE=0.
5. 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 the previous step.
6. Perform an Identify Namespace Data Structure Command and record the value of the NUSE field.
7. Configure the NVMe Host to issue a Write Uncorrectable command to the same LBA range deallocated in the previous steps.
8. Perform an Identify Namespace Data Structure Command and record the value of the NUSE field.

Observable Results:

1. Verify that the value reported in the NUSE field in the second Identify Namespace Data Structure Command is greater than the value reported in the first Identify Namespace Data Structure Command.

Case 5: Allocate via Write Zeroes (OF-FYI, M)

Test Procedure:

1. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports Dataset Management. If the command is not supported this test is not applicable.
2. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports Write Zeroes. If the command is not supported this test is not applicable.
3. Perform a Set Feature command for FID 05h, Error Recovery Feature, to set DULBE=0.
4. Perform a Get Feature command for FID 05h, Error Recovery Feature, to ensure that DULBE=0.
5. 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 the previous step.
6. Perform an Identify Namespace Data Structure Command and record the value of the NUSE field.

© 2026 UNH–IOL

7. Configure the NVMe Host to issue a Write Zeroes command to the same LBA range deallocated in the previous steps, with the Deallocate bit (CDW12.DEAC) set to 0.
8. Perform an Identify Namespace Data Structure Command and record the value of the NUSE field.

Observable Results:

1. Verify that the value reported in the NUSE field in the second Identify Namespace Data Structure Command is greater than the value reported in the first Identify Namespace Data Structure Command.

Case 6: Deallocate via Dataset Management then COPY, DULBE=0 (OF-FYI, 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. Check the ONCS field of the Identify Controller Data structure to determine if the DUT supports Dataset Management. If the command is not supported this test is not applicable.
3. Perform an Identify Namespace Data Structure Command and record the value of the DLFEAT field.
4. Perform a Set Feature command for FID 05h, Error Recovery Feature, to set DULBE=0.
5. Perform a Get Feature command for FID 05h, Error Recovery Feature, to ensure that DULBE=0.
6. Configure the NVMe Host to issue a Write command to write a known data pattern to the controller.
7. 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 the previous step.
8. Configure the NVMe Host to issue 2 Copy commands to the same LBA range deallocated in the previous step.
9. Perform a Read operation to the LBA range that the Copy command wrote to.

Observable Results:

1. Verify that the correct data pattern is written to the LBA range that the Copy command wrote to according to the following parameters
    a. Verify that the data written has all bytes cleared to 0h if bits 2:0 in the DLFEAT field are set to 001b.
    b. Verify that the data written has all bytes set to FFh if bits 2:0 in the DLFEAT field are set to 010b.
    c. Verify that the data written has either all bytes cleared to 0h or all bytes set to FFh if bits 2:0 in the DLFEAT field are set to 000b.

Possible Problems: None known.

© 2026 UNH–IOL

Test 2.14 – Copy (OF-FYI, FYI)

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

References:

Old Ref: [1] NVMe Specification Chapter 6
NVM Express NVM Command Set Specification 1.0a : 3.2.2

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 (OF-FYI, 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.
3. Perform a Copy command with valid fields and for allocated logical blocks.

Observable Results:

1. Verify that the MSSRL and MCL fields in the Identify Namespace Data Structure are set to non-zero values.
2. Verify that an completion queue entry for the Copy command is posted to the associated I/O Completion Queue indicating the status of the command.

Case 2: MSSRL Exceeded (OF-FYI, 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’ (83h).

Case 3: MCL Exceeded (OF-FYI, 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’ (83h).

© 2026 UNH–IOL

Case 4: MSRC Exceeded (OF-FYI, 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' (83h).

Case 5: Descriptor Format Not Supported (OF-FYI, 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 a Copy Command with the Descriptor Format field set to a value of Fh.

Observable Results:

1. Verify that the Copy Command is aborted with a status of 'Invalid Field in Command' (02h).

Case 6: PRACT Mismatch (OF-FYI, 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 a Copy Command with the PRACT bit cleared to '0h' in the PRINFOR field and the PRACT bit set to '1h' in the PRINFOW field.

Observable Results:

1. Verify that the Copy Command is aborted with a status of 'Invalid Field in Command' (02h).

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 2.14.6 and expect the same observable results.

1. **Separate Buffer**
2. **Extended Buffer**

Case 7: Copy to Read Only Media (OF-FYI, 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. Check the NWPC field. If Namespace Write Protection is not supported then this test is not applicable.
2. Perform a Set Feature operation for the Namespace Write Protection Config Feature (FID=84h) to set a particular namespace to a write protected state (i.e. read only).
3. Perform a Copy Command which will attempt to write to the namespace which was set to read only in the previous step.

Observable Results:

1. Verify that the Copy Command is aborted with a status of 'Namespace is Write Protected' (20h).

Possible Problems: None known.

© 2026 UNH–IOL

Case 8: Copy Descriptor Formats 2h and 3h Disabled (OF-FYI, 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. In the same data structure, check if bytes 535:534 indicate support for Copy Descriptor Formats 2h and 3h. If Copy Descriptor Formats 2h and 3h are supported, then this test is not applicable.
2. Configure the host to issue a Get Feature for FID 16h. If FID 16h is not supported, this test is not applicable.
3. Configure the host to issue a Set Feature for FID 16h with a buffer to set Copy Descriptor Format 2h Enable (CDF2E) and Copy Descriptor Format 3h Enable (CDF3E) (bits 2 and 3 of byte 4 respectively) to zero, disabling the two Copy Descriptor Formats. If already set to zero, this step can be skipped.
4. Configure the host to issue Copy command with the Descriptor Format field set to 2h.
5. Configure the host to issue a second Copy command with the Descriptor Format field set to 3h.

Observable Results:

1. Verify that all Admin commands complete successfully.
2. Verify that both Copy commands complete with a status of ‘Invalid Field in Command’ (02h).
3. Verify that both bits 0 and 1 of byte 4 (Copy Descriptor Formats Enable) are cleared to 0h.

Case 9: Copy Descriptor Formats 1h and 3h with Destination NSID Formatted to Use 16b Guard Protection (OF-FYI, 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 host to issue an Identify Namespace command to a single namespace. If this namespace is not formatted to use 16b Guard Protection, issue a Format NVM Command to format the namespace to a LBAF that has 16b Guard Protection. If there is no format that exists on the NVMe device that has 16b Guard Protection, or the Format NVM command is not supported, this test is not applicable.
3. Configure the NVMe Host to issue a Copy Command with the Copy Descriptor Format Field set to 1h and the Destination NSID field set to the namespace that was configured in the previous step.
4. Configure the NVMe Host to issue a Copy Command with the Copy Descriptor Format Field set to 3h and the Destination NSID field set to the namespace that was configured in the previous step

Observable Results:

1. Verify that all Admin commands complete successfully.
2. Verify that both Copy commands complete with a status of ‘Invalid Namespace or Format’ (0Bh).

Case 10: Copy Descriptor Formats 0h and 2h with Destination NSID Formatted to Use 32b Guard Protection (OF-FYI, 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 host to issue an Identify Namespace command to a single namespace. If this namespace is not formatted to use 32b Guard Protection, issue a Format NVM Command to format the namespace to a LBAF that has 32b Guard Protection. If there is no format that exists on the NVMe device that has 32b Guard Protection, or the Format NVM command is not supported, this test is not applicable.
3. Configure the NVMe Host to issue a Copy Command with the Copy Descriptor Format Field set to 0h and the Destination NSID field set to the namespace that was configured in the previous step.
4. Configure the NVMe Host to issue a Copy Command with the Copy Descriptor Format Field set to 2h and the Destination NSID field set to the namespace that was configured in the previous step.

Observable Results:

© 2026 UNH–IOL

1. Verify that all Admin commands complete successfully.
2. Verify that both Copy commands complete with a status of 'Invalid Namespace or Format' (0Bh).

Case 11: Copy Descriptor Formats 0h and 2h with Destination NSID Formatted to Use 64b Guard Protection (OF-FYI, 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 host to issue an Identify Namespace command to a single namespace. If this namespace is not formatted to use 64b Guard Protection, issue a Format NVM Command to format the namespace to a LBAF that has 64b Guard Protection. If there is no format that exists on the NVMe device that has 64b Guard Protection, or the Format NVM command is not supported, this test is not applicable.
3. Configure the NVMe Host to issue a Copy Command with the Copy Descriptor Format Field set to 0h and the Destination NSID field set to the namespace that was configured in the previous step.
4. Configure the NVMe Host to issue a Copy Command with the Copy Descriptor Format Field set to 2h and the Destination NSID field set to the namespace that was configured in the previous step.

Observable Results:

1. Verify that all Admin commands complete successfully.
2. Verify that both Copy commands complete with a status of 'Invalid Namespace or Format' (0Bh).

Case 12: Copy Command with Unusable Source Namespace Identifiers (OF-FYI, 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. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
3. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create) and valid values for the attached data structure to the NVMe Controller Under Test in order to create a new namespace. Two namespaces are required for this test. Both namespaces are to be formatted without Protection Information. Do not attach the second namespace to the controller.
4. Configure the NVMe host to issue a Copy command with the Copy Descriptor Format set to 2h, the Source Namespace Identifier set to 0h, Number of Logical Blocks cleared to 0, and Fast Copy Only cleared to 0. The Destination NSID is to be set to only NSID that is attached to the controller.
5. Perform the same command in step 4 two more times, once with the Source NSID field set to an invalid Namespace Identifier (one that does not exist), and third Copy command with the NSID field set to the value FFFFFFFFh.
6. Configure the NVMe host to issue a Copy command with the Copy Descriptor Format set to 2h, the Source Namespace Identifier set to the NSID of the unattached NSID, Number of Logical Blocks cleared to 0, and Fast Copy Only cleared to 0. The Destination NSID is to be set to only NSID that is attached to the controller.

Observable Results:

1. Verify that all Admin commands complete successfully.
2. Verify that all Copy commands in step 4 and 5 complete with a status of 'Invalid Namespace or Format' (0Bh).
3. Verify that the Copy command in step 6 completes with 'Invalid Field in Command' (02h).

Case 13: Copy Command Copy Descriptor Format 2h and 3h with Overlapping I/O Range (OF-FYI, 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.

© 2026 UNH–IOL

2. Configure the host to issue a Get Feature for FID 16h. If FID 16h is not supported, this test is not applicable.
3. Configure the host to issue a Set Feature for FID 16h with a buffer to set Copy Descriptor Format 2h Enable (CDF2E) and Copy Descriptor Format 3h Enable (CDF3E) (bits 2 and 3 of byte 4 respectively) to 1, enabling the two Copy Descriptor Formats. If these Copy Descriptor Formats are not supported, then this test is not applicable. Support for the Copy Descriptor formats can be found in the Identify Controller I/O Command Set Independent Data Structure Copy Descriptor Formats Supported field.
4. Configure the NVMe host to issue a Copy command with the Copy Descriptor Format set to 2h, SLBA set to 0h, Starting Destination LBA (SDLBA) set to 1h and NLB set to 3 in the Source Range Entry.
5. Perform the same Copy command in step 2, but set the Copy Descriptor Format field to 3h.

Observable Results:

1. Verify that all Admin commands complete successfully.
2. Verify both Copy commands complete with the status of 'Overlapping I/O Range' (87h).

Case 14: NVM Copy Single Atomicity Support Requirements (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify Controller Command for the NVM Command Set Specific Data Structure (CNS 06h, CSI 0h). If the VER field (23:20) does not set the MJR field to 1h, this test is not applicable.
2. Configure the NVMe Host to issue an Identify Controller Command (CNS=01h) to the DUT to receive the Identify Controller Data Structure, I/O Command Set Independent. Check the ONCS field Bit 8 to determine if the DUT supports the Copy Command. If the Copy Command is not supported this test is not applicable. If bits 3 and 2 of the Copy Descriptor Formats field (byte 543) are cleared to zero, this test is not applicable. If the VER field is reported to be less than 2.0, then this test is not applicable.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the NVM Copy Single Atomicity (NVMCSA) field in ONCS is set to 1.

Case 15: NVM Copy Single Atomicity Verification (OF-FYI, 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 NVMe Host to issue a Write command to the controller in order to write a known data pattern to two consecutive LBAs on the NVMe Device.
3. Configure the NVMe Host to issue a Read Command to the controller sent to the adjacent two LBAs (i.e. if the Write command is sent to LBA 0-1, Read LBA 2-3 in this command)
4. Configure the NVMe Host to issue a Copy command to copy the known data pattern from LBAs 0-1 to LBAs 2-3
5. Read the LBAs copied to in step 4.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the NVM Copy Single Atomicity (NVMCSA) field in ONCS is set to 1.
3. Verify that the Copy command issued in step 4 was completed with a single write command and that the data is present in step 5.

Case 16: Cross Namespace Copy without Protection Information (OF-FYI, 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.

© 2026 UNH–IOL

2. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
3. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create) and valid values for the attached data structure to the NVMe Controller Under Test to create a new namespace. Two namespaces are required for this test. Both namespaces are to be formatted without Protection Information and are the same LBAF. Attach the second namespace to the controller.
4. Configure the NVMe Host to issue a Write command to the controller to write a known data pattern to two consecutive LBAs on the first namespace (LBA0 and LBA1).
5. Configure the NVMe host to issue a Copy command with the Copy Descriptor Format set to 2h, the Source Namespace Identifier set to 1h, Starting LBA set to 0, Number of Logical Blocks set to, and Fast Copy Only cleared to 0. The NSID specified in the command is to be the second namespace that was created in step 3, and the SLBA field in CDW10 is to be set to 0.
6. Configure the NVMe Host to read the first two LBAs on the second namespace.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that all Copy commands complete successfully.
3. Verify that the same data that was written to the first namespace in step 4 is present on the second namespace that was created in step 3.

Possible Problems: It may be necessary to delete the original namespace if it uses the entire capacity of the NVM, in order to create a second namespace.

Case 17: Cross Namespace Copy with PRACT Mismatched (OF-FYI, 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. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
3. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create) and valid values for the attached data structure to the NVMe Controller Under Test to create a new namespace. Two namespaces are required for this test.
4. Configure the NVMe Host to issue a Copy command specifying the second namespace created in step 3, with the PRACT bit cleared to ‘0h’ in the PRINFOR field and the PRACT bit set to ‘1h’ in the PRINFOW field, the Source NSID field set to namespace 1, and the Starting Destination LBA (SDLBA) field in the Copy command set to 0h.

Observable Results:

1. Verify that the Identify and Namespace Management commands complete successfully.
2. Verify that the Copy Command is aborted with a status of ‘Invalid Field in Command’ (02h).

Possible Problems: It may be necessary to delete the original namespace if it uses the entire capacity of the NVM, in order to create a second namespace.

Case 18: Cross Namespace Copy with LBADs Mismatched (OF-FYI, 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. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
3. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create) and valid values for the attached data structure to the NVMe Controller Under Test to create a new namespace. Two namespaces are required for this test.

© 2026 UNH–IOL

4. Attach and format the new namespace to a LBAF that has a different Logical Block Data Size than the first namespace. If the Format NVM command is not supported, or if the FNA bit in the Identify Controller Data Structure, I/O Command Set Independent is set to 1, then this test is not applicable.
5. Configure the NVMe Host to issue a Write command to the controller to write a known data pattern to two consecutive LBAs on the first namespace (LBA0 and LBA1).
6. Configure the NVMe host to issue a Copy command with the Copy Descriptor Format set to 3h, the Source Namespace Identifier set to 1h, Starting LBA set to 0, Number of Logical Blocks set to 1, and Fast Copy Only cleared to 0. The NSID specified in the command is to be the second namespace that was created in step 3, and the SLBA field in CDW10 is to be set to 0.
7. Configure the NVMe Host to issue a Read Command to read the LBAs that were targeted by the Copy command on the second namespace.

Observable Results:

1. Verify that the Identify and Namespace Management commands complete successfully.
2. Verify that the Copy Command is aborted with a status of 'Incompatible Namespace or Format' (85h).
3. Verify that the data was not copied from one namespace to the other.

Possible Problems: It may be necessary to delete the original namespace if it uses the entire capacity of the NVM, in order to create a second namespace.

Case 19: Cross Namespace Copy with DPS Mismatched (OF-FYI, 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. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
3. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create) and valid values for the attached data structure to the NVMe Controller Under Test to create a new namespace. Two namespaces are required for this test.
4. Attach and format the new namespace to a LBAF that has a different DPS value than the first namespace. If the Format NVM command is not supported, or if the FNA bit in the Identify Controller Data Structure, I/O Command Set Independent is set to 1h, then this test is not applicable.
5. Configure the NVMe Host to issue a Write command to the controller to write a known data pattern to two consecutive LBAs on the first namespace (LBA0 and LBA1).
6. Configure the NVMe host to issue a Copy command with the Copy Descriptor Format set to 3h, the Source Namespace Identifier set to 1h, Starting LBA set to 0h, Number of Logical Blocks set to 1h, and Fast Copy Only cleared to 0h. The NSID specified in the command is to be the second namespace that was created in step 3, and the SLBA field in CDW10 is to be set to 0h.
7. Configure the NVMe Host to issue a Read Command to read the LBAs that were targeted by the Copy command on the second namespace.

Observable Results:

1. Verify that the Identify and Namespace Management commands complete successfully.
2. Verify that the Copy Command is aborted with a status of 'Incompatible Namespace or Format' (85h).
3. Verify that the data was not copied from one namespace to the other.

Possible Problems: It may be necessary to delete the original namespace if it uses the entire capacity of the NVM, in order to create a second namespace.

Case 20: Cross Namespace Copy with PIFA Mismatched (OF-FYI, 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.

© 2026 UNH–IOL

2. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
3. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create) and valid values for the attached data structure to the NVMe Controller Under Test to create a new namespace. Two namespaces are required for this test.
4. Attach and format the new namespace to a LBAF that has a different PIFA (i.e. the Storage Tag Masking Level Attribute value is different) value than the first namespace. If the Format NVM command is not supported, or if the FNA bit in the Identify Controller Data Structure, I/O Command Set Independent is set to 1, then this test is not applicable.
5. Configure the NVMe Host to issue a Write command to the controller to write a known data pattern to two consecutive LBAs on the first namespace (LBA0 and LBA1).
6. Configure the NVMe host to issue a Copy command with the Copy Descriptor Format set to 3h, the Source Namespace Identifier set to 1h, Starting LBA set to 0h, Number of Logical Blocks set to 1h, and Fast Copy Only cleared to 0. The NSID specified in the command is to be the second namespace that was created in step 3, and the SLBA field in CDW10 is to be set to 0h.
7. Configure the NVMe Host to issue a Read Command to read the LBAs that were targeted by the Copy command on the second namespace.

Observable Results:

1. Verify that the Identify and Namespace Management commands complete successfully.
2. Verify that the Copy Command is aborted with a status of ‘Incompatible Namespace or Format’ (85h).
3. Verify that the data was not copied from one namespace to the other.

Possible Problems: It may be necessary to delete the original namespace if it uses the entire capacity of the NVM, in order to create a second namespace.

Case 21: Cross Namespace Copy with Logical Block Storage Tag Mask Mismatched (OF-FYI, 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. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
3. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create) and valid values for the attached data structure to the NVMe Controller Under Test to create a new namespace. Two namespaces are required for this test.
4. Attach and format the new namespace to a LBAF that has a different LBSTM value than the first namespace. If the Format NVM command is not supported, or if the FNA bit in the Identify Controller Data Structure, I/O Command Set Independent is set to 1h, then this test is not applicable.
5. Configure the NVMe Host to issue a Write command to the controller to write a known data pattern to two consecutive LBAs on the first namespace (LBA0 and LBA1).
6. Configure the NVMe host to issue a Copy command with the Copy Descriptor Format set to 3h, the Source Namespace Identifier set to 1h, Starting LBA set to 0h, Number of Logical Blocks set to 1h, and Fast Copy Only cleared to 0. The NSID specified in the command is to be the second namespace that was created in step 3, and the SLBA field in CDW10 is to be set to 0h.
7. Configure the NVMe Host to issue a Read Command to read the LBAs that were targeted by the Copy command on the second namespace.

Observable Results:

1. Verify that the Identify and Namespace Management commands complete successfully.
2. Verify that the Copy Command is aborted with a status of ‘Incompatible Namespace or Format’ (85h).
3. Verify that the data was not copied from one namespace to the other.

© 2026 UNH–IOL

Possible Problems: It may be necessary to delete the original namespace if it uses the entire capacity of the NVM, in order to create a second namespace.

Case 22: Cross Namespace Copy, One Namespace without Protection Information to a Namespace with Protection Information (OF-FYI, 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. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
3. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create) and valid values for the attached data structure to the NVMe Controller Under Test to create a new namespace. Two namespaces are required for this test.
4. Format the first namespace to an LBAF without Protection Information. Format the second namespace to a LBAF with Protection Information. If the Format NVM command is not supported, or if the FNA bit in the Identify Controller Data Structure, I/O Command Set Independent is set to 1h, then this test is not applicable.
5. Configure the NVMe Host to issue a Write command to the controller to write a known data pattern to two consecutive LBAs on the first namespace (LBA0 and LBA1).
6. Configure the NVMe host to issue a Copy command with the Copy Descriptor Format set to 3h, the Source Namespace Identifier set to 1h, Starting LBA set to 0h, Number of Logical Blocks set to 1h, and Fast Copy Only cleared to 0. The NSID specified in the command is to be the second namespace that was created in step 3, and the SLBA field in CDW10 is to be set to 0h.
7. Configure the NVMe Host to issue a Read Command to read the LBAs that were targeted by the Copy command on the second namespace.

Observable Results:

1. Verify that the Identify and Namespace Management commands complete successfully.
2. Verify that the Copy Command is aborted with a status of ‘Incompatible Namespace or Format’ (85h).
3. Verify that the data was not copied from one namespace to the other.

Possible Problems: It may be necessary to delete the original namespace if it uses the entire capacity of the NVM, in order to create a second namespace.

Case 23: Cross Namespace Copy, One Namespace with Protection Information to a Namespace Without Protection Information (OF-FYI, 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. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
3. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create) and valid values for the attached data structure to the NVMe Controller Under Test to create a new namespace. Two namespaces are required for this test.
4. Format the first namespace to an LBAF with Protection Information. Format the second namespace to a LBAF with no Protection Information. If the Format NVM command is not supported, or if the FNA bit in the Identify Controller Data Structure, I/O Command Set Independent is set to 1h, then this test is not applicable.
5. Configure the NVMe Host to issue a Write command to the controller to write a known data pattern to two consecutive LBAs on the first namespace (LBA0 and LBA1).
6. Configure the NVMe host to issue a Copy command with the Copy Descriptor Format set to 3h, the Source Namespace Identifier set to 1h, Starting LBA set to 0h, Number of Logical Blocks set to 1h, and Fast Copy Only cleared to 0h. The NSID specified in the command is to be the second namespace that was created in step 3, the SLBA field in CDW10 set to 0h, and PRINFOR.PRACT bit is cleared to ‘0h’.

© 2026 UNH–IOL

7. Configure the NVMe Host to issue a Read Command to read the LBAs that were targeted by the Copy command on the second namespace.

Observable Results:
1. Verify that the Identify and Namespace Management commands complete successfully.
2. Verify that the Copy Command is aborted with a status of 'Incompatible Namespace or Format' (85h).
3. Verify that the data was not copied from one namespace to the other.

Possible Problems: It may be necessary to delete the original namespace if it uses the entire capacity of the NVM, in order to create a second namespace.

Case 24: Cross Namespace Copy, Metadata Size on Destination Namespace is Larger than the PI Size of the Namespace (OF-FYI, 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. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
3. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create) and valid values for the attached data structure to the NVMe Controller Under Test to create a new namespace. Two namespaces are required for this test.
4. Format the first namespace to an LBAF with Protection Information. Format the second namespace to a LBAF with no Protection Information. If the Format NVM command is not supported, or if the FNA bit in the Identify Controller Data Structure, I/O Command Set Independent is set to 1, then this test is not applicable.
    a. Select an LBAF on the first namespace (the one formatted Protection Information) that has a metadata size that is larger than the PI size.
5. Configure the NVMe Host to issue a Write command to the controller to write a known data pattern to two consecutive LBAs on the first namespace (LBA0 and LBA1).
6. Configure the NVMe host to issue a Copy command with the Copy Descriptor Format set to 3h, the Source Namespace Identifier set to 1h, Starting LBA set to 0h, Number of Logical Blocks set to, and Fast Copy Only cleared to 0h. The NSID specified in the command is to be the second namespace that was created in step 3, the SLBA field in CDW10 set to 0h, and PRINFOR.PRACT bit is set to '1h'.
7. Configure the NVMe Host to issue a Read Command to read the LBAs that were targeted by the Copy command on the second namespace.

Observable Results:
1. Verify that the Identify and Namespace Management commands complete successfully.
2. Verify that the Copy command is aborted with a status of 'Incompatible Namespace or Format' (85h).
3. Verify that the data was not copied from one namespace to the other.

Possible Problems: It may be necessary to delete the original namespace if it uses the entire capacity of the NVM, in order to create a second namespace.

Case 25: Cross Namespace Copy, No Reachability Association (OF-FYI, FYI)

Test Procedure:
1. Issue an Identify command with CNS=01h to retrieve the Identify Controller Data Structure. Check ONCS Bit 8 and the RRSUP bit to determine if the Controller and NVM Subsystem support the Copy command and Reachability Reporting. Check OACS bit 3 for support of the Namespace Management command. If the Copy command, Reachability Reporting, or Namespace Management are not supported, then this test is not applicable.
2. Check bits 2 or 3 of the Copy Descriptor Formats Supported field to ensure that Cross-Namespace Copy is supported. If both bits 2 and 3 are cleared to 0, then this test is not applicable.

© 2026 UNH–IOL

3. Issue a Set Features command to the Host Behavior Support Feature (FID=16h) to enable all supported Copy Descriptor formats from the previous step.
4. Issue a Namespace Management command with SEL=0h (Create) to create a new unattached namespace and record its NSID.
5. For all supported Copy Descriptor Formats:
    a. Issue a single valid Copy command with
        i. The Associated Cross-Namespace Copy Descriptor Format (2h or 3h),
        ii. A Source Range Entry containing the NSID of the namespace created in the previous step,
        iii. Any attached namespace as the destination namespace.

Observable Results:

1. For each Copy command issued, verify it is aborted with a status of ‘Namespace Not Reachable’ (8Ah).
2. Verify all other commands completed successfully.

© 2026 UNH–IOL

Group 3: NVM Features

Overview:
This section describes a method for performing conformance verification for NVMe products implementing NVM Features.

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).

© 2026 UNH–IOL

Test 3.1 – Metadata Handling (OF-FYI, M)

Purpose: To verify that an NVMe Controller properly handles metadata.

References:

Old Ref: [1] NVMe Specification 8.2
NVM Express NVM Command Set Specification 1.0a : 5.8.3

Resource Requirements:

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

Last Modification: November 28, 2017

Discussion: Metadata is additional data allocated on a per logical block basis. There is no requirement for how the host makes use of the metadata area. One of the most common usages for metadata is to convey end–to–end protection information. The metadata may be transferred by the controller to or from the host in one of two ways. The mechanism used is selected when the namespace is formatted.

One of the transfer mechanisms shall be selected for each namespace when it is formatted; transferring a portion of metadata with one mechanism and a portion with the other mechanism is not supported.

Test Setup: See Appendix A.

Case 1: Extended LBA (OF-FYI, M)

The first mechanism for transferring the metadata is as a contiguous part of the logical block that it is associated with. The metadata is transferred at the end of the associated logical block, forming an extended logical block. In this case, both the logical block data and logical block metadata are pointed to by the PRP1 and PRP2 pointers (or SGL Entry 1 if SGLs are used).

Test Procedure:

1. Check the Metadata Size (MS) field in the Identify LBA Format Data Structure, which is within the Identify Namespace Data Structure. If MS is set to 0, then Metadata is not supported and this test is not applicable.
2. Configure the NVMe Host to issue a Write command to the controller specifying an LBA in a namespace which has been formatted to allow metadata via extended LBAs with known data patterns for both the data and metadata.
3. Configure the NVMe Host to issue a Read command to the controller specifying the LBA which was previously written to.

Observable Results:

1. Verify that the data and metadata returned by the controller exactly match the data and metadata patterns which were written.

Case 2: Separate Buffer (OF-FYI, M)

The second mechanism for transferring the metadata is as a separate buffer of data. In this case, the metadata is pointed to with the Metadata Pointer, while the logical block data is pointed to by the Data Pointer. When a command uses PRPs for the metadata in the command, the metadata is required to be physically contiguous. When a command uses SGLs for the metadata in the command, the metadata is not required to be physically contiguous.

Test Procedure:

1. Check the Metadata Size (MS) field in the Identify LBA Format Data Structure, which is within the Identify Namespace Data Structure. If MS is set to 0, then Metadata is not supported and this test is not applicable.

© 2026 UNH–IOL

2. Configure the NVMe Host to issue a Write command to the controller specifying an LBA in a namespace which has been formatted to allow metadata via a separate buffer with known data patterns for both the data and metadata.
3. Configure the NVMe Host to issue a Read command to the controller specifying the LBA which was previously written to.

Observable Results:

1. Verify that the data and metadata returned by the controller exactly match the data and metadata patterns which were written.

Possible Problems: Support for each metadata transfer mechanism is indicated on a per namespace basis according to the FLBAS field Bit 4 of the Identify Namespace data structure. If there is no active namespaces which supports a specific transfer mechanism then tests utilizing that mechanism cannot be performed. If the NVMe Device targets NVMe Specification revision 1.2 or higher and the NVMe Controller Under Test supports the Namespace Management command, then the NVMe Host may create a new namespace which supports a specific transfer mechanism in order to perform procedure steps which require that transfer mechanism.

© 2026 UNH–IOL

Test 3.2 – End–to–end Data Protection (M)

Purpose: To verify that an NVMe Controller can properly handle end–to–end data protection.

References:

Old Ref: [1] NVMe Specification 8.3
NVM Express NVM Command Set Specification 1.0a : 5.2

Resource Requirements:

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

Last Modification: March 2, 2016

Discussion: To provide robust data protection from the application to the NVM media and back to the application itself, end–to–end data protection may be used. If this optional mechanism is enabled, then additional protection information (e.g. CRC) is added to the LBA that may be evaluated by the controller and/or host software to determine the integrity of the logical block.

This additional protection information, if present, is either the first eight bytes of metadata or the last eight bytes of metadata, based on the format of the namespace. For metadata formats with more than eight bytes, if the protection information is contained within the first eight bytes of metadata, then the CRC does not cover any metadata bytes. For metadata formats with more than eight bytes, if the protection information is contained within the last eight bytes of metadata, then the CRC covers all metadata bytes up to but excluding these last eight bytes.

Test Setup: See Appendix A.

Case 1: Write Command Processing (M)

Protection information processing may occur as a side effect of a Write command. If the namespace was formatted with protection information and the PRACT bit is cleared to ‘0’, then logical block data and metadata containing protection information are transferred from the host to NVM. As the logical block and metadata pass through the controller, the protection information is checked. If a protection information check error is detected, the command completes with the status code of the error detected (i.e., End–to–end Guard Check, End–to–end Application Tag Check or End–to end Reference Tag Check). If the namespace was formatted with protection information and the PRACT bit is set to ‘1’, then logical block data is transferred from the host to the controller. The controller inserts protection information and the logical block data and metadata containing protection information are written to NVM.

Test Procedure:

1. Check the DPC field of the Identify Namespace Data Structure to determine if the DUT supports End to End Data Protection Capabilities. If DPC set to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Write command to the controller with the PRACT bit cleared to ‘0’ and specifying an LBA in a namespace which has been formatted with protection information and also specifying properly formatted protection information with a known data pattern.
3. Configure the NVMe Host to issue a Write command to the controller with the PRACT bit cleared to ‘0’ and specifying an LBA in a namespace which has been formatted with protection information and also specifying improperly formatted protection information.
4. Configure the NVMe Host to issue a Write command to the controller with the PRACT bit set to ‘1’ and specifying an LBA in a namespace which has been formatted with protection information and also specifying a known data pattern.

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. For steps 2 and 4, verify that the completion queue entry for each command indicates Success status.

© 2026 UNH–IOL

3. For step 3, verify that the completion queue entry for the command indicates an appropriate status code (i.e., End–to–end Guard Check, End–to–end Application Tag Check or End–to end Reference Tag Check depending on how the protection information was improperly formatted).

Possible Problems: For devices that are formatted to Protection Information Type = 3 & PRCHK=1 for 2.0 and beyond it is possible to receive back a status code of Invalid Protection Information. For 1.4 devices that support Type=3 & PRCHK=1, it is possible to have Invalid Protection Information or Invalid Field in Command.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 3.2.1 and expect the same observable results.

1. **Separate Buffer**
2. **Extended Buffer**

Case 2: Read Command Processing (M)

Protection information processing may occur as a side effect of a Read command. The processing parallels Write command processing with the exception that logical block data flows in the opposite direction. When the PRACT bit is cleared to '0' and the namespace was formatted with protection information, logical block data and metadata are transferred from NVM to the host and checked by the controller. When the PRACT bit is set to '1' and the namespace was formatted with protection information, logical block data and metadata are transferred from the NVM to the controller. The controller checks the protection information and then removes it from the metadata before passing the LBA to the host. If the namespace format contains metadata beyond the protection information, then the protection information is not stripped regardless of the state of the PRACT bit (i.e., the metadata field remains the same size in the host as that in NVM).

Test Procedure:

1. Check the DPC field of the Identify Namespace Data Structure to determine if the DUT supports End to End Data Protection Capabilities. If DPC set to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Read command to the controller with the PRACT bit cleared to '0' and specifying the LBA which was written to in Case 1 step 2.
3. Configure the NVMe Host to issue a Read command to the controller with the PRACT bit cleared to '0' and specifying the LBA which was written to in Case 1 step 4.
4. Configure the NVMe Host to issue a Read command to the controller with the PRACT bit set to '1' and specifying the LBA which was written to in Case 1 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. For step 2, verify that data returned by the controller exactly matches the data and metadata which was written.
3. For step 3, verify that the data returned by the controller exactly matches the data which was written and contains additional metadata containing properly formatted and correct protection information.
4. For step 4, verify that the data returned by the controller contains no protection information.

Possible Problems: Case 2 requires reading the data which was written in Case 1. Therefore these test cases must be performed in succession or the data must be rewritten as part of the Case 2 test procedure implementation.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 3.2.2 and expect the same observable results.

1. **Separate Buffer**
2. **Extended Buffer**

Case 3: DPC=0 PRACT ignored by Controller (M)

Test Procedure:

© 2026 UNH–IOL

1. Check the DPC field of the Identify Namespace Data Structure. If DUT supports End to End Data protection this test is not applicable.
2. 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 PRCHK=0, and with PRACT = 1.
3. 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 2.

Observable Results:

1. Verify that the known data pattern was read correctly from the NVMe Device exactly as it was written

Case 4: DPC=0 PRCHK ignored by Controller (M)

Test Procedure:

1. Check the DPC field of the Identify Namespace Data Structure. If DUT supports End to End Data protection this test is not applicable.
2. 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 PRCHK=4h, PRACT = 0 and a metadata with an invalid crc.
3. 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 2.

Observable Results:

1. Verify that the known data pattern was read correctly from the NVMe Device exactly as it was written

Case 5: DPC= 0 & STS=0 Storage Tag ignored by Controller (M)

Test Procedure:

1. Check the DPC field of the Identify Namespace Data Structure. If DUT supports End to End Data protection this test is not applicable.
2. Check the Storage Tag Size field in the Extended LBA Format data structure. If the Storage Tag Size is nonzero this test is not applicable.
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 to the controller in order to read from the same LBA which was written to in step 3 with storage tag check enabled and an invalid ELBST

Observable Results:

1. Verify that the known data pattern was read correctly from the NVMe Device exactly as it was written

Case 6: Metadata size=0 MSET ignored by Controller (M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to the controller in order to check the LBA Format data structure verify a supported format has MS set equal 0 and at least one supported format has a non-zero MS, otherwise this test is not applicable.
2. Configure the NVMe Host to issue a Format NVM Command with an LBA Format that has a non-zero MS.
3. Configure the host to issue a Format NVM Command with an LBA Format that has MS=0 and the MSET=1.
4. Configure the NVMe Host to issue an Identify Namespace command.

Observable Results:

1. Verify after step 2 that the namespace's current format reported, as in use, supports metadata
2. Verify after step3 that the namespaces current format reported, as in use, doesn't support metadata
3. Verify that the MS field in the second Identify for the namespace for this test is cleared to 0 and FLBAS bit 4 is cleared to 0.

© 2026 UNH–IOL

Case 7: Qualified Protection Information Format Support (FYI)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
2. Retrieve a list of all active NSIDs using the Identify command with CNS 02h
3. For each namespace:
    a. Configure the NVMe Host to issue an Identify command using CNS 00h to receive the Identify Namespace data structure
    b. Perform an Identify command with CNS 05h and CSI of 00h to retrieve the NVM command set specific Identify Namespace Data Structure
    c. Check bit 3 of the Protection Information Capabilities field (QPIFS). If set to 0h, non-unique LBAFs for this namespace are not applicable
    d. Check the value of the Storage Tag Masking Level Attribute field (STMLA). If set to any value except for 010b, non-unique LBAFs for this namespace are not applicable.
    e. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create), LBSTM cleared to all 0s, and valid values for the attached data structure (returned in previous step) to the NVMe Controller Under Test in order to create a new namespace
    f. For each unique format index (if applicable):
        i. Perform an Identify command with CNS 0Ah, CSI 00h and CNS Specific Identifier of the format index to retrieve the NVM command set specific Identify Namespace Data Structure.
        ii. Check bit 3 of the Protection Information Capabilities field (QPIFS). If set to 0h, this format index is not applicable
        iii. Check the value of the Storage Tag Masking Level Attribute field (STMLA). If set to any value except for 010b, this format index is not applicable.
        iv. If the unique format index has PIF other than 00b and STS value bigger than 0:
            i. Check the ELBAS field in Identify Controller Data Structure to determine if the DUT supports Extended LBA Format Supported bit set to ‘1h’. If the DUT does not support ELBAS, then this format index is not applicable.
            ii. Configure NVMe Host to issue Get Feature command with FID 16h to retrieve Host Behavior Support Data Structure. If LBA Format Extension Enable is cleared to 0h, configure NVMe Host to issue Set Feature command with FID 16h to enable LBA Format Extension Enable.
        v. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create), LBSTM cleared to all 0s, and valid values for the attached data structure (returned in previous step) to the NVMe Controller Under Test in order to create a new namespace.

Observable Results:

1. Verify that the Namespace Management commands do not complete successfully.
2. Verify that the new namespace created successfully.

Case 8: Qualified protection Information Format Supported, STMLA=001b (FYI)

Test Procedure:

1. Retrieve a list of all active NSIDs using the Identify command with CNS 02hRetrieve a list of all active NSIDs using the Identify command with CNS 02h
2. For each namespace:
    a. Perform an Identify command with CNS 05h and CSI of 00h to retrieve the NVM Command Set specific Identify Namespace Data Structure.
    b. Check bit 3 of the Protection Information Capabilities field (QPIFS). If set to 0, ignore this namespace for this test
    c. Check the value of the STMLA field. If not set to a value of 001b (Byte Granularity Masking), ignore this namespace for this test.

© 2026 UNH–IOL

d. Check the value of the LBSTM field

Observable Results:

1. Verify that all bits contained in each byte in the Logical Block Storage Tag Mask fields are the same value. Each byte can either be FFh or 0h.

Case 9: 16b Guard Protection Information Storage Tag Mask Requirement (FYI)

Test Procedure:

1. Retrieve a list of all active NSIDs using the Identify command with CNS 02h
2. For each namespace:
    a. Configure the NVMe Host to issue an Identify command using CNS 00h to receive the Identify Namespace data structure
    b. Check the NULBAF field of the Identify Namespace data structure. If cleared to 0, this test is not applicable
    c. For each unique format index:
        i. Perform an Identify command with CNS 0Ah, CSI 00h and CNS Specific Identifier of the format index to retrieve the NVM Command Set specific Identify Namespace Data Structure.
        ii. Check bit 3 of the Protection Information Capabilities field (QPIFS). If set to 0, this format index is not applicable.
        iii. Check if the Protection Information Format (PIF) in the Extended LBA Format Data Structure. If set to a value other than 11b, this format index is not applicable.
        iv. Check the value of the Storage Tag Masking Level Attribute field (STMLA). If set to a value other than 010b, this format index is not applicable.
        v. Check the value of the 16BPISTM bit.

Observable Results:

1. Verify that bit 16BPISTM is set to 1 for each applicable unique format index.

Case 10: Storage Tag Masking Level Attributes LBSTM Formatting (FYI)

Test Procedure:

1. Retrieve a list of all active NSIDs using the Identify command with CNS 02h
2. For each namespace:
    a. Perform an Identify command with CNS 05h and CSI of 00h to retrieve the NVM Command Set specific Identify Namespace Data Structure.
    b. Check bit 3 of the Protection Information Capabilities field (QPIFS). If set to 0, ignore this namespace for this test
    c. Read the value of STMLA.
    d. Read the value of LBSTM

Observable Results:

1. If STMLA is set to 000b (Bit Granularity Masking), the Logical Block Storage Tag Mask fields are any combination of ’1’s and ’0’s
2. If STMLA is set to 001b (Byte Granularity Masking), the Logical Block Storage Tag Mask fields are all the same value on a per byte basis (i.e. byte 0 is all ’1’s, byte 1 is all ’0’s, etc.)
3. If STMLA is set to 010b (Masking Not Supported), every bit in the Logical Block Storage Tag Mask fields are set to ‘1’

Case 11: STMLA Value Check, QPIFS=0 (M)

Test Procedure:

1. Retrieve a list of all active NSIDs using the Identify command with CNS 02h
2. For each namespace:

© 2026 UNH–IOL

a. Configure the NVMe Host to issue an Identify command using CNS 00h to receive the Identify Namespace data structure
b. Check bit 3 of the Protection Information Capabilities field (QPIFS). If set to 1, non-unique LBAFs are not applicable, otherwise, read the value of STMLA
c. For each unique format index (if applicable):
    i. Perform an Identify command with CNS 0Ah, CSI 00h and CNS Specific Identifier of each reported format index to retrieve the NVM Command Set specific Identify Namespace Data Structure.
    ii. Check bit 3 of the Protection Information Capabilities field (QPIFS). If set to 1, this format index is not applicable, otherwise, read the value of STMLA.

Observable Results:
1. Verify that STMLA is all cleared to zero for each applicable format index.

Case 12: STMLA Value Check, PIF not equal to 11b (M)

Test Procedure:
1. Retrieve a list of all active NSIDs using the Identify command with CNS 02h
2. For each namespace:
    a. Configure the NVMe Host to issue an Identify command using CNS 00h to receive the Identify Namespace data structure
    b. Check the NULBAF field of the Identify Namespace data structure. If cleared to 0, this namespace is not applicable
    c. For each unique format index (if applicable):
        i. Perform an Identify command with CNS 0Ah, CSI 00h and CNS Specific Identifier of each reported format index to retrieve the NVM Command Set specific Identify Namespace Data Structure.
        ii. Read the value of Protection Information Format (PIF). If PIF is set to a value of 11b, this format index is not applicable, otherwise, read the value of STMLA.

Observable Results:
1. For each unique format index, verify that STMLA is cleared to zero

Possible Problems: Devices with no unique format indices and other format indices with PIF set to 11b will skip this test.

Case 13: Qualified Protection Information Not Supported (M)

Test Procedure:
1. Retrieve a list of all active NSIDs using the Identify command with CNS 02h
2. For each namespace:
    a. Configure the NVMe Host to issue an Identify command using CNS 00h to receive the Identify Namespace data structure
    b. Check bit 3 of the Protection Information Capabilities field (QPIFS). If set to 1, non-unique format indices are not applicable.
    c. For each non-unique format index:
        i. Read the value of Protection Information Format in the associated Extended LBA Format Data Structure
    d. For each unique format index (if applicable):
        i. Perform an Identify command with CNS 0Ah, CSI 00h and CNS Specific Identifier of each reported format index to retrieve the NVM Command Set specific Identify Namespace Data Structure.
        ii. Read the value of Protection Information Format for the current format index.

© 2026 UNH–IOL

Observable Results:

1. For each applicable format index, verify that the value of PIF is not 11b

© 2026 UNH–IOL

Test 3.3 – Power Management (OF, M)

Purpose: To verify that an NVMe Controller can properly handle power management.

References:

Old Ref: [1] NVMe Specification 8.4
NVM Express Base Set Specification 1.0a : 8.15

Resource Requirements:

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

Last Modification: June 24, 2015

Discussion: The power management capability allows the host to manage NVM subsystem power statically or dynamically. Static power management consists of the host determining the maximum power that may be allocated to an NVM subsystem and setting the NVM Express power state to one that consumes this amount of power or less. Dynamic power management consists of the host modifying the NVM Express power state to best satisfy changing power and performance objectives. This power management mechanism is meant to complement and not replace autonomous power management performed by a controller.

The number of power states implemented by a controller is returned in the Number of Power States Supported (NPSS) field in the Identify Controller data structure. A controller shall support at least one power state and may optionally support up to a total of 32 power states. Power states are contiguously numbered starting with zero such that each subsequent power state consumes less than or equal to the maximum power consumed in the previous state. Thus, power state zero indicates the maximum power that the NVM subsystem is capable of consuming.

Associated with each power state is a Power State Descriptor in the Identify Controller data structure. The descriptors for all implemented power states may be viewed as forming a table as shown in Table 6 which contains sample power state values for a controller with seven implemented power states.

Test Setup: See Appendix A.

Case 1: Relative Write Latency (OF, M)

Relative Write Latency (RWL): This field indicates the relative write latency associated with this power state. The value in this field shall be less than the number of supported power states (e.g., if the controller supports 16 power states, then valid values are 0 through 15). A lower value means lower write latency.

Table 6 – Example Power State Descriptor Table

Power State	Maximum Power (MP) (W)	Entry Latency (ENTLAT) (µs)	Exit Latency (EXLAT) (µs)	Relative Read Throughput (RRT)	Relative Read Latency (RRL)	Relative Write Throughput (RWT)	Relative Write Latency (RWL)
0	25	5	5	0	0	0	0
1	18	5	7	0	0	1	0
2	18	5	8	1	0	0	0
3	15	20	15	2	0	2	0
4	10	20	30	1	1	3	0
5	8	20	50	2	2	4	0
6	5	20	5000	4	3	5	1

Test Procedure:

© 2026 UNH–IOL

1. Configure the NVMe Host to issue an Identify command specifying CNS value 01h to the controller in order to receive back an Identify Controller data structure.
2. For each power state supported by the controller as indicated by the NPSS field, read the RWL field of the associated Power State Descriptor.

Observable Results:

1. Verify that the relative write latency values are all less than the number of supported power states.

Case 2: Relative Write Throughput (OF, M)

Relative Write Throughput (RWT): This field indicates the relative write throughput associated with this power state. The value in this field shall be less than the number of supported power states (e.g., if the controller supports 16 power states, then valid values are 0 through 15). A lower value means higher write throughput.

The amount of data an application can write to stable storage on the server over a period of time is a measurement of the write throughput of a distributed file system. Write throughput is therefore an important aspect of performance

Test Procedure:

1. Configure the NVMe Host to issue an Identify command specifying CNS value 01h to the controller in order to receive back an Identify Controller data structure.
2. For each power state supported by the controller as indicated by the NPSS field, read the RWL field of the associated Power State Descriptor.

Observable Results:

1. Verify that the relative write throughput values are all less than the number of supported power states.

Case 3: Relative Read Latency (OF, M)

Relative Read Latency (RRL): This field indicates the relative read latency associated with this power state. The value in this field shall be less than the number of supported power states (e.g., if the controller supports 16 power states, then valid values are 0 through 15). A lower value means lower read latency.

Test Procedure:

1. Configure the NVMe Host to issue an Identify command specifying CNS value 01h to the controller in order to receive back an Identify Controller data structure.
2. For each power state supported by the controller as indicated by the NPSS field, read the RWL field of the associated Power State Descriptor.

Observable Results:

1. Verify that the relative read latency values are all less than the number of supported power states.

Case 4: Relative Read Throughput (OF, M)

Relative Read Throughput (RRT): This field indicates the relative read throughput associated with this power state. The value in this field shall be less than the number of supported power states (e.g., if the controller supports 16 power states, then valid values are 0 through 15). A lower value means higher read throughput.

Test Procedure:

1. Configure the NVMe Host to issue an Identify command specifying CNS value 01h to the controller in order to receive back an Identify Controller data structure.
2. For each power state supported by the controller as indicated by the NPSS field, read the RWL field of the associated Power State Descriptor.

Observable Results:

1. Verify that the relative read throughput values are all less than the number of supported power states.

© 2026 UNH–IOL

Case 5: Power Management Feature (OF, M)

The host may dynamically modify the power state using the Set Features command and determine the current power state using the Get Features command. The host may directly transition between any two supported power states.

The Entry Latency (ENTLAT) field in the power management descriptor indicates the maximum amount of time in microseconds that it takes to enter that power state and the Exit Latency (EXLAT) field indicates the maximum amount of time in microseconds that it takes to exit that state. The maximum amount of time to transition between any two power states is equal to the sum of the old state's exit latency and the new state's entry latency. The host is not required to wait for a previously submitted power state transition to complete before initiating a new transition.

Test Procedure:

1. Configure the NVMe Host to issue a Set Features command for the Power Management Feature for each of the supported Power State values supported by the NVMe Controller to the controller.
2. After each Set Features command completes, configure the NVMe Host to issue a Get Features command for the Power Management Feature to the controller.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
2. For each Get Features command, verify that the controller successfully transitioned to the selected power state.

Possible Problems: None known.

© 2026 UNH–IOL

Test 3.4 – Host Memory Buffer (M)

Purpose: To verify that an NVMe system can properly handle supports host memory buffer.

References:

Old Ref: [1] NVMe Specification 8.9, 5.14.1.13, NVMe v1.3 ECN 004a
NVM Express Base Specification 2.0a : 8.9

Resource Requirements:

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

Last Modification: April 29, 2019

Discussion: The Host Memory Buffer feature allows the controller to utilize an assigned portion of host memory exclusively. The use of the host memory resources is vendor specific. Host software may not be able to provide any or a limited amount of the host memory resources requested by the controller. The controller shall function properly without host memory resources. Refer to section 5.14.1.13 of the NVMe Specification.

During initialization, host software may provide a descriptor list that describes a set of host memory address ranges for exclusive use by the controller. The host memory resources assigned are for the exclusive use of the controller (host software should not modify the ranges) until host software requests that the controller release the ranges and the controller completes the Set Features command. The controller is responsible for initializing the host memory resources. Host software should request that the controller release the assigned ranges prior to a shutdown event, a Runtime D3 event, or any other event that requires host software to reclaim the assigned ranges. After the controller acknowledges that it is no longer using the ranges, host software may reclaim the host memory resources. In the case of Runtime D3, host software should provide the host memory resources to the controller again and inform the controller that the ranges were in use prior to the RTD3 event and have not been modified.

Test Setup: See Appendix A.

Case 1: Proper Structure (M)

Test Procedure:

1. Check the HMPRE field of the Identify Controller data structure to determine if the DUT supports Host Memory Buffer. If the HMPRE field is set to 0, 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.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the associated Admin Completion Queue indicating the status for the command.
2. Verify that the Host Memory Buffer Preferred Size (HMPRE) field of the Identify Controller data structure is larger than or equal to the value indicated by the Host Memory Buffer Minimum Size (HMMIN) field.

Case 2: Configuration (M)

The Host Memory Feature controls the Host Memory Buffer. The attributes are indicated in Command Dword 11, Command Dword 12, Command Dword 13, Command Dword 14, and Command Dword 15.

The Host Memory Buffer feature provides a mechanism for the host to allocate a portion of host memory for the controller to use exclusively. After a successful completion of a Set Features enabling the host memory buffer, the host shall not write to the associated host memory region, buffer size, or descriptor list until the host memory buffer has been disabled.

© 2026 UNH–IOL

After a successful completion of a Set Features command that disables the host memory buffer, the controller shall not access any data in the host memory buffer until the host memory buffer has been enabled. The controller should retrieve any necessary data from the host memory buffer in use before posting the completion queue entry for the Set Features command. Posting of the completion queue entry for the Set Features command acknowledges that it is safe for the host software to modify the host memory buffer contents. Refer to section 8.9 of the NVMe Specification.

Test Procedure:

1. Check the HMPRE field of the Identify Controller data structure to determine if the DUT supports Host Memory Buffer. If the HMPRE field is set to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Set Features command for the Host Memory Buffer feature with the Enable Host Memory (EHM) bit set to '1h' to the NVMe Controller.
3. Configure the NVMe Host to issue a Get Features command for the Host Memory Buffer feature to the NVMe Controller.
4. Configure the NVMe Host to issue a Set Features command for the Host Memory Buffer feature with the Enable Host Memory (EHM) bit cleared to '0h' to the NVMe Controller.
5. Configure the NVMe Host to issue a Get Features command for the Host Memory Buffer feature to the NVMe Controller.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
2. Verify in each case that the value for EHM returned by the controller in response to the Get Feature command matches what was set by the Host in the Set Feature command.

Case 3: Reset Persistent (M)

The host memory resources are not persistent in the controller across a reset event. Host software should provide the previously allocated host memory resources to the controller after the reset completes. If host software is providing previously allocated host memory resources (with the same contents) to the controller, the Memory Return bit is set to '1h' in the Set Features command. The controller shall ensure that there is no data loss or data corruption in the event of a surprise removal while the Host Memory Buffer feature is being utilized.

Test Procedure:

1. Check the HMPRE field of the Identify Controller data structure to determine if the DUT supports Host Memory Buffer. If the HMPRE field is set to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Get Features command for the Host Memory Buffer feature to the NVMe Controller.
3. Configure the NVMe Host to issue a Set Features command for the Host Memory Buffer feature with the Enable Host Memory (EHM) bit set to '1h' and Save (SV) set to 0h, to the NVMe Controller.
4. Configure the NVMe Host to issue a Get Features command for the Host Memory Buffer feature to the NVMe Controller.
5. Configure the NVMe Host to issue a Controller Level Reset to the NVMe Controller.
6. Configure the NVMe Host to issue a Get Features command for the Host Memory Buffer feature to the NVMe Controller.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
2. Verify that the DUT responds to the Get Features command in step 4 with EHM set to 1h.
3. Verify that after the controller reset, the controller responds to the Get Feature command with EHM set to 0h.

Case 4: Enable HMB when Already Enabled (M)

Test Procedure:

© 2026 UNH–IOL

1. Check the HMPRE field of the Identify Controller data structure to determine if the DUT supports Host Memory Buffer. If the HMPRE field is set to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Get Features command for the Host Memory Buffer feature to the NVMe Controller.
3. Configure the NVMe Host to issue a Set Features command for the Host Memory Buffer feature with the Enable Host Memory (EHM) bit set to '1' and Save (SV) set to 0, to the NVMe Controller.
4. Configure the NVMe Host to issue a Get Features command for the Host Memory Buffer feature to the NVMe Controller.
5. Configure the NVMe Host to issue a Set Features command for the Host Memory Buffer feature with the Enable Host Memory (EHM) bit set to '1' and Save (SV) set to 0, to the NVMe Controller.

Observable Results:

1. Verify that the DUT responds to the Set Features command in step 5 with status 'Command Sequence Error' (0Ch).

Case 5: Disable HMB when Already Disabled (M)

Test Procedure:

1. Check the HMPRE field of the Identify Controller data structure to determine if the DUT supports Host Memory Buffer. If the HMPRE field is set to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Get Features command for the Host Memory Buffer feature to the NVMe Controller.
3. Configure the NVMe Host to issue a Set Features command for the Host Memory Buffer feature with the Enable Host Memory (EHM) bit set to '1' and Save (SV) set to 0, to the NVMe Controller.
4. Configure the NVMe Host to issue a Get Features command for the Host Memory Buffer feature to the NVMe Controller.
5. Configure the NVMe Host to issue a Set Features command for the Host Memory Buffer feature with the Enable Host Memory (EHM) bit set to '0' and Save (SV) set to 0, to the NVMe Controller.
6. Configure the NVMe Host to issue a Get Features command for the Host Memory Buffer feature to the NVMe Controller.
7. Configure the NVMe Host to issue a Set Features command for the Host Memory Buffer feature with the Enable Host Memory (EHM) bit set to '0' and Save (SV) set to 0, to the NVMe Controller.

Observable Results:

1. Verify that the DUT responds to the Set Features command in step 7 with status Success.

Possible Problems: None known.

Case 6: HMB Restrict Non-Operational Power State Access Enable (FYI)

Test Procedure:

1. Check the HMPRE field of the Identify Controller data structure to determine if the DUT supports Host Memory Buffer. If the HMPRE field is set to 0 or the HMBR is set to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Set Features command specifying FID 0Dh (Host Memory Buffer) with CDW11 bit 0 Enable Host Memory (EHM) set to 1.
3. Configure the NVMe Host to issue a Get Features command specifying FID 0Dh (Host Memory Buffer). Record the value of the feature.
4. Configure the NVMe Host to issue a Set Features command specifying FID 0Dh (Host Memory Buffer) with CDW11 bit 0 Enable Host Memory (EHM) cleared to 0h.
5. Configure the NVMe Host to issue a Set Features command specifying FID 0Dh (Host Memory Buffer) with CDW11 bits 0 Enable Host Memory (EHM) and 2 Host Memory Non-Operational Access Restriction Enable (HMNARE) set 1.
6. Configure the NVMe Host to issue a second Get Features command specifying FID 0Dh (Host Memory Buffer). Record the value of the feature

© 2026 UNH–IOL

Observable Results:

1. Verify the Set Features command in step 2 and 4 complete with the status ‘Successful Completion’ (00h).
2. Verify that the value of FID 0Dh is reported as 01h in the first Get Features command.
3. Verify that the value of FID 0Dh is reported as 05h in the second Get Features command.

Case 7: HMB Restrict Non-Operational Power State Access Not Supported (FYI)

Test Procedure:

1. Check the HMPRE field of the Identify Controller data structure to determine if the DUT supports Host Memory Buffer. If the HMPRE field is set to 0h or the HMBR is set to 1h, this test is not applicable.
2. Configure the NVMe Host to issue a Set Features command for the Host Memory Buffer feature with the Enable Host Memory (EHM) bit set to ‘1h’ to the NVMe Controller.
3. Configure the NVMe Host to issue a Get Features command specifying FID 0Dh (Host Memory Buffer). Record the value of the feature.
4. Configure the NVMe Host to issue a Set Features command specifying FID 0Dh (Host Memory Buffer) with CDW11 bit 2 Host Memory Non-operational Access Restriction Enable (HMNARE) set to 1h.
5. Configure the NVMe Host to issue a second Get Features command specifying FID 0Dh (Host Memory Buffer). Record the value of the feature.

Observable Results:

1. Verify the Set Features command in step 4 completes with a status of ‘Invalid Field in Command’ (02h).
2. Verify that the value of FID 0Dh is identical between the two Get Features commands.

Case 8: HMB Restrict Non-Operational Power State CQE Verification (FYI)

Test Procedure:

1. Check the HMPRE field of the Identify Controller data structure to determine if the DUT supports Host Memory Buffer. If the HMPRE field is set to 0h or the HMBR is set to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Set Features command for the Host Memory Buffer feature with the Enable Host Memory (EHM) bit set to ‘1h’, and CDW11 bit 2 set to ‘1h’ to the NVMe Controller.
3. Configure the NVMe Host to issue a Get Features command specifying FID 0Dh (Host Memory Buffer). Record the value of the feature.
4. Configure the NVMe Host to issue a Set Features command specifying FID 0Ch (Autonomous Power State Transition) with CDW11 bit 00 cleared to 0h to disable Autonomous Power State Transitions.
5. Configure the NVMe Host to issue a Set Features command specifying FID 02h (Power Management) and a value that indicates a non-operational power state.
6. Configure the NVMe Host to issue a Get Features command specifying FID 0Dh (Host Memory Buffer). Record the value of the feature.

Observable Results:

1. Verify that all commands complete with a status of ‘Successful Completion’ (00h).
2. Verify that the Completion Queue Entry Dword 0 of the first Get Features command reports bit 2 set to 1h, and that bits 1 and 3 are cleared to 0h.
3. Verify that the Completion Queue Entry Dword 0 of the second Get Features command reports bit 2 and 3 are set to 1h, and that bit 1 is cleared to 0h.

© 2026 UNH–IOL

Test 3.5 – Replay Protected Memory Block (IP)

Purpose: To verify that an NVMe system can properly handle replay protected memory blocks.

References:

Old Ref: [1] NVMe Specification 8.10
NVM Express Base Specification 2.0a : 8.18

Resource Requirements:

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

Last Modification: April 9, 2019

Discussion: The Replay Protected Memory Block (RPMB) provides a means for the system to store data to a specific memory area in an authenticated and replay protected manner. This is provided by first programming authentication key information to the controller that is used as a shared secret. The system is not authenticated in this phase, therefore the authentication key programming should be done in a secure environment (e.g., as part of the manufacturing process). The authentication key is utilized to sign the read and write accesses made to the replay protected memory area with a Message Authentication Code (MAC). Use of random number (nonce) generation and a write count register provide additional protection against replay of messages where messages could be recorded and played back later by an attacker.

The controller may support multiple RPMB targets. RPMB targets are not contained within a namespace. Security Send and Security Receive commands for RPMB do not use the namespace ID field; NSID shall be cleared to 0h. Each RPMB target operates independently – there may be requests outstanding to multiple RPMB targets at once (where the requests may be interleaved between RPMB targets). In order to guarantee ordering the host should issue and wait for completion for one Security Send or Security Receive command at a time. Each RPMB target requires individual authentication and key programming. Each RPMB target may have its own unique Authentication Key.

The message types defined in Figure 223 of the NVMe Specification are used by the host to communicate with an RPMB target. Request Message Types are sent from the host to the controller. Response Message Types are sent to the host from the controller.

Test Setup: See Appendix A.

Case 1: RPMB Operations (IP)

The host sends a Request Message Type to the controller to request an operation by the controller or to deliver data to be written into the RPMB memory block. To deliver a Request Message Type, the host uses the Security Send command. If the data to be delivered to the controller is more than reported in Identify Controller data structure, the host sends multiple Security Send commands to transfer the entire data.

The host sends a Response Message Type to the controller to read the result of a previous operation request, to read the Write Counter, or to read data from the RPMB memory block. To deliver a Response Message Type, the host uses the Security Receive command. If the data to be read from the controller is more than reported in Identify Controller data structure, the host sends multiple Security Receive commands to transfer the entire data.

Test Procedure:

1. Check the RPMBS field in the Identify Controller Data Structure to determine if the controller supports RPMB. If RPMB is not supported this test is not applicable.
2. Configure the NVMe Host to issue a Security Send command with the feature ID set to Replay Protected Memory Blocks to the NVMe Controller.
3. Configure the NVMe Host to issue a Security Receive command with the feature ID set to Replay Protected Memory Blocks to the NVMe Controller.

Observable Results:

© 2026 UNH–IOL

1. Verify that after the completion of the command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
2. Verify that the data received is correct, and if the data is larger than reported in the Identify Controller data structure the host sends multiple Security Receive commands to transfer the entire data.

Case 2: Authentication Key Programming (IP)

Authentication Key programming is initiated by a Security Send command to program the Authentication Key to the specified RPMB target, followed by a subsequent Security Send command to request the result, and lastly, the host issues a Security Receive command to retrieve the result.

Test Procedure:

1. Check the RPMBS field in the Identify Controller Data Structure to determine if the controller supports RPMB. If RPMB is not supported this test is not applicable.
2. Configure the NVMe Host to issue a Security Send command for the Replay Protected Memory Blocks Feature to the NVMe Controller, with the RPMB target set to a target to access, and the Mac/Key field set to the key to be programmed, and the Request/Response field set to “0001h”.
3. Configure the NVMe Host to issue a Security Send command for the Replay Protected Memory Blocks Feature to the NVMe Controller, with the RPMB target set to a target to access, and the Request/Response field set to “0005h”.
4. Configure the NVMe Host to issue a Security Receive command for the Replay Protected Memory Blocks Feature to the NVMe Controller.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
2. Verify the Security Receive command is populated correctly by the NVMe Controller.

Case 3: Read Write Counter Value (IP)

The Read Write Counter Value sequence is initiated by a Security Send command to request the Write Counter value, followed by a Security Receive command to retrieve the Write Counter result.

Test Procedure:

1. Check the RPMBS field in the Identify Controller Data Structure to determine if the controller supports RPMB. If RPMB is not supported this test is not applicable.
2. Configure the NVMe Host to issue a Security Send command for the Replay Protected Memory Blocks Feature to the NVMe Controller, with the RPMB target set to a target to access, and the Nonce field set to a specific Nonce generated by the host, and the Request/Response field set to “0002h”.
3. Configure the NVMe Host to issue a Security Receive command for the Replay Protected Memory Blocks Feature to the NVMe Controller.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
2. Verify the Security Receive command is populated correctly by the NVMe Controller.

Case 4: Authenticated Data Write (IP)

The success of programming the data should be checked by the host by reading the result register of the RPMB.

1. Check the RPMBS field in the Identify Controller Data Structure to determine if the controller supports RPMB. If RPMB is not supported this test is not applicable.
2. The host initiates the Authenticated Data Write verification process by issuing a Security Send command with delivery of a RPMB data frame containing the Request Message Type = 0005h.

© 2026 UNH–IOL

3. The controller returns a successful completion of the Security Send command when the verification result is ready for retrieval.
4. The host should then retrieve the verification result by issuing a Security Receive command.
5. The controller returns a successful completion of the Security Receive command and returns the RPMB data frame containing the Response Message Type = 0300h, the incremented counter value, the data address, the MAC and result of the data programming operation.

Test Procedure:

1. Configure the NVMe Host to issue a Security Send command for the Replay Protected Memory Blocks Feature to the NVMe Controller, with the Mac/Key field set to a MAC generated by the host, and the RPMB target set to a target to access, and the Write Counter field set to the current write counter value, and the Address field set to the RPMB address, and the Sector Count field set the to the number of 512B blocks, and the Request/Response field set to "0003h", and the Data field set to data to be written.
2. Configure the NVMe Host to issue a Security Send command for the Replay Protected Memory Blocks Feature to the NVMe Controller, with the RPMB target set to a target to access, and the Request/Response field set to "0005h".
3. Configure the NVMe Host to issue a Security Receive command for the Replay Protected Memory Blocks Feature to the NVMe Controller.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
2. Verify the Security Receive command is populated correctly by the NVMe Controller.

Case 5: Authenticated Data Read (IP)

The Authenticated Data Read sequence is initiated by a Security Send command. The RPMB data frame delivered from the host to the controller includes the Request Message Type = 0004h, Nonce, Address, and the Sector Count.

When the controller receives this RPMB Data Frame, it first checks the Address. If there is an error in the Address then the result is set to 0004h (address failure) and the data read is not valid. When the host receives a successful completion of the Security Send command from the controller, it should send a Security Receive command to the controller to retrieve the data. The controller returns an RPMB Data Frame with Response Message Type (0400h), the Sector Count, a copy of the Nonce received in the request, the Address, the Data, the controller calculated MAC, and the Result. Note: It is the responsibility of the host to verify the MAC returned on an Authenticated Data Read Request.

If the data transfer from the addressed location in the controller fails, the returned Result is 0006h (read failure). If the Address provided in the Security Send command is not valid, then the returned Result is 0004h (address failure). If another error occurs during the read procedure then the returned Result is 0001h (general failure).

Test Procedure:

1. Check the RPMBS field in the Identify Controller Data Structure to determine if the controller supports RPMB. If RPMB is not supported this test is not applicable.
2. Configure the NVMe Host to issue a Security Send command for the Replay Protected Memory Blocks Feature to the NVMe Controller, with the RPMB target set to a target to access, and the Nonce field set to a Nonce generated by the host, and the Address field set to the RPMB address, and the Sector Count set to the number of 512B blocks, and the Request/Response field set to "0004h", and the Data field set to data to be written.
3. Configure the NVMe Host to issue a Security Receive command for the Replay Protected Memory Blocks Feature to the NVMe Controller.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
2. Verify the Security Receive command is populated correctly by the NVMe Controller.

© 2026 UNH–IOL

Possible Problems: This is a Mandatory if Supported test, depending on device feature support according to the Replay Protected Memory Block Support (RPMBS) field of the Identify Controller data structure.

© 2026 UNH–IOL

Test 3.6 – IO Determinism (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller can properly supports predictable latency mode if supported.

References:

Old Ref: [1] NVMe Specification TP 4003c
NVM Express Base Specification 2.0a : 8.16.1

Resource Requirements:

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

Last Modification: April 22, 2019

Discussion: Predictable Latency Mode is used to achieve predictable latency for read and write operations. When configured to operate in this mode using the Predictable Latency Mode Config Feature, the namespaces in an NVM Set provide windows of operation for deterministic operation or non-deterministic operation. Support for predictable latency mode also requires support for associated Features and Log Pages.

Test Setup: See Appendix A.

Case 1: Predictable Latency Mode Supported (OF-FYI, FYI)

Test Procedure:

1. Check Bit 5 of the CTRATT field in the Identify Controller Data Structure. If CTRATT Bit 5 is set to 0, this test is not applicable.
2. Configure the Testing Station to issue a Set Feature for Feature Identifier 13h “Predictable Latency Mode Config” to enable Predictable Latency Mode.
3. Configure the Testing Station to issue a Get Feature for Feature Identifier 13h “Predictable Latency Mode Config”.
4. Configure the Testing Station to issue a Get Feature for Feature Identifier 14h “Predictable Latency Mode Window”.
5. Configure the Testing Station to issue a Get Log Page for Log Identifier 0Ah “Predictable Latency Per NVM Set”.
6. Configure the Testing Station to issue a Get Log Page for Log Identifier 0Bh “Predictable Latency Event Aggregate”.
7. Configure the Testing Station to issue a Get Feature for Feature Identifier 13h “Predictable Latency Mode Config”.

Observable Results:

1. Verify that each of the Get Log and Get Feature operations completed with status ‘Success’.

Case 2: Predictable Latency Mode Not Supported (OF, M)

Test Procedure:

1. Check Bit 5 of the CTRATT field in the Identify Controller Data Structure. If CTRATT Bit 5 is set to 1, this test is not applicable.
2. Configure the Testing Station to issue a Set Feature for Feature Identifier 13h “Predictable Latency Mode Config” to enable Predictable Latency Mode.
3. Configure the Testing Station to issue a Get Feature for Feature Identifier 13h “Predictable Latency Mode Config”.
4. Configure the Testing Station to issue a Get Feature for Feature Identifier 14h “Predictable Latency Mode Window”.
5. Configure the Testing Station to issue a Get Log Page for Log Identifier 0Ah “Predictable Latency Per NVM Set”.

© 2026 UNH–IOL

6. Configure the Testing Station to issue a Get Log Page for Log Identifier 0Bh “Predictable Latency Event Aggregate”.
7. Configure the Testing Station to issue a Get Feature for Feature Identifier 13h “Predictable Latency Mode Config”.

Observable Results:

1. Verify that None of the Get Log and Get Feature operations completed with status ‘Success’.

Case 3: Predictable Latency Mode Not Enabled (OF-FYI, FYI)

Test Procedure:

1. Check Bit 5 of the CTRATT field in the Identify Controller Data Structure. If CTRATT Bit 5 is set to 0, this test is not applicable.
2. Configure the Testing Station to issue a Get Feature for Feature Identifier 13h “Predictable Latency Mode Config”.
3. If Predictable Latency Mode is enabled, configure the Testing Station to issue a Set Feature for Feature Identifier 13h “Predictable Latency Mode Config” to disable Predictable Latency Mode. Otherwise, proceed to the next step.
4. Configure the Testing Station to issue a Get Feature for Feature Identifier 14h “Predictable Latency Mode Window”.

Observable Results:

1. Verify that the Get Feature command for FID 14h returned an error of “Invalid Field in Command”.

Possible Problems: None known.

© 2026 UNH–IOL

Test 3.7 – Namespace Write Protection (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller has properly implemented Namespace Write Protection if supported.

References:

Old Ref: [1] NVMe Specification TP 4005b
NVM Express Base Specification 2.0a : 8.12

Resource Requirements:

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

Last Modification: April 24, 2019

Discussion: Namespace Write Protection is an optional Feature identifier that controls write protection on a per namespace basis. This Feature may be used to prevent modification of the specified namespace. The controller fails commands that attempt to modify the namespace while it is write protected.

Test Setup: See Appendix A.

Case 1: Enable and Disable Write Protection (OF, FYI)

Test Procedure:

1. Check Namespace Write Protection Capabilities (NWPC) in the Identify Controller Data Structure. If Bits 0, 1, 2 of NWPC are set to 0, the DUT does not support Namespace Write Protection, and this test is not applicable.
2. Perform a Get Feature Operation for the Namespace Write Protection Config (Feature ID 84h). Verify that Write Protection State is 000b. If not, perform a Set Feature operation to set Namespace Write Protection Config to 'No Write Protect' 000b. If the protection state of the namespace is set to Write Protect Until Power Cycle (010b) or Permanent Write Protect (011b), then the Set Feature command will fail with an error of Feature Not Changeable and this test is not applicable.
3. Perform a Write operation to write a known data pattern (i.e. AAAAh) to the Namespace.
4. Perform a Set Feature operation to set Namespace Write Protection Config to 'Write Protect' 001b.
5. Perform the following Admin commands to the namespace.
    a. Device Self Test
    b. Get Feature
    c. Get Log
    d. Identify
6. Perform the following NVM commands to the namespace.
    a. Compare
    b. Read
    c. Flush
    d. Verify
7. Perform a Write command with a known data pattern which is different than the data pattern used in Step 3 to the namespace (i.e. BBBBh).
8. Perform a Read command to the namespace.
9. Perform a Set Feature operation to set Namespace Write Protection Config to 'No Write Protect' 000b.
10. Perform a Write command with a known data pattern, different than the data pattern used in Step 3 to the namespace (i.e. CCCCh).
11. Perform a Read command to the namespace.

Observable Results:

1. Verify that each of the commands in step 5 and 6 completed successfully.
2. Verify that the Read operation performed in Step 8, returned the data pattern written in Step 2, and not the data pattern attempted to be written in Step 7.
3. Verify that the Read operation performed in Step 11, returned the data pattern written in Step 10.

© 2026 UNH–IOL

4. Verify that the Write operation attempted in Step 7 completed with status ‘Namespace is Write Protected’ (20h).

Possible Problems: None known.

© 2026 UNH–IOL

Test 3.8 – Persistent Memory Region (IP, IP)

Purpose: To verify that an NVMe Controller has properly implemented Persistent Memory Region if supported.

References:

Old Ref: [1] NVMe Specification 4.8
NVM Express Base Specification 2.0a : 8.14

Resource Requirements:

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

Last Modification: May 21, 2020

Discussion: The Persistent Memory Region (PMR) is an optional region of general purpose PCI Express read/write persistent memory that may be used for a variety of purposes. The controller indicates support for the PMR by setting CAP.PMRS to '1' and indicates whether the controller supports command data and metadata transfers to or from the PMR by setting support flags in the PMRCAP register. When command data and metadata transfers to or from PMR are supported, all data and metadata associated with a particular command shall be either entirely located in the Persistent Memory Region or outside the Persistent Memory Region.

Test Setup: See Appendix A.

Case 1: PMR Persistence Across Reset (IP, IP)

Test Procedure:

1. Check that CAP.PMRS is set to 1. If CAP.PMRS is not set to 1, the DUT does not support Persistent Memory Region and this test is not applicable.
2. The Testing Station acting as a Host should set PMRMSC.CBA to a valid value. If the DUT supports a version of the NVMe specification greater than NVMe v1.4, or if the DUT supports NVMe v1.4 and has implemented NVMe v1.4 ECN 006, then the Host should set PMRMSCL.CBA and PMRMSCU.CBA to valid values.
3. Enable the PMR controller memory space via PMRMSC.CMSE and PMRCTL.EN.
4. Wait 2x the timer specified in PMRCAP.PMRTO
5. Check that the controller indicates that the PMR is ready by clearing PMRSTS.NRDY to 0.
6. Perform a WRITE operation of a non-zero known pattern to the address range specified for the PMR.
7. Perform a Conventional Reset, PCIe Hot Reset.
8. Wait 2x the timer specified in PMRCAP.PMRTO
9. Perform a READ operation to the address range specified for the PMR.
10. Perform a WRITE operation of all 0s' to the address range specified for the PMR.
11. Perform a READ operation to the address range specified for the PMR.

Observable Results:

1. Verify that each READ operation returned the same known data pattern that was previously written to the PMR.

Case 2: PMR Persistence Across Disable and Enable (IP, IP)

Test Procedure:

1. Check that CAP.PMRS is set to 1. If CAP.PMRS is not set to 1, the DUT does not support Persistent Memory Region and this test is not applicable.
2. The Testing Station acting as a Host should set PMRMSC.CBA to a valid value.
3. Enable the PMR controller memory space via PMRMSC.CMSE and PMRCTL.EN.
4. Wait 2x the timer specified in PMRCAP.PMRTO
5. Check that the controller indicates that the PMR is ready by clearing PMRSTS.NRDY to 0.
6. Perform a WRITE operation of a non-zero known pattern to the address range specified for the PMR.
7. Disable the PMR using the PMRCTL.EN bit.

© 2026 UNH–IOL

8. Wait 2x the timer specified in PMRCAP.PMRTO
9. Enable the PMR using the PMRCTL.EN bit.
10. Wait 2x the timer specified in PMRCAP.PMRTO
11. Perform a READ operation to the address range specified for the PMR.
12. Perform a WRITE operation of all 0s' to the address range specified for the PMR.
13. Perform a READ operation to the address range specified for the PMR.

Observable Results:

1. Verify that each READ operation returned the same known data pattern that was previously written to the PMR.

Case 3: PMRMSC L and U CBA Persistence Across Reset (IP, IP)

Test Procedure:

1. Check that CAP.PMRS is set to 1. If CAP.PMRS is not set to 1, the DUT does not support Persistent Memory Region and this test is not applicable.
2. The must DUT support a version of the NVMe specification greater than NVMe v1.4, or the DUT must support NVMe v1.4 and have implemented NVMe v1.4 ECN 006, otherwise this test is not applicable.
3. The Testing Station acting as host should set PMRMSCL.CBA and PMRMSCU.CBA to valid values by writing new values to those registers.
4. The Testing Station should Enable the PMR controller memory space via PMRMSC.CMSE and PMRCTL.EN.
5. Wait 2x the timer specified in PMRCAP.PMRTO
6. Check that the controller indicates that the PMR is ready by clearing PMRSTS.NRDY to 0.
7. Perform a Controller Reset.
8. Read the PMRMSCL and PMRMSCU registers.

Observable Results:

1. Verify that that the PMRMSCU and PMRMSCL values did not change after the Controller Reset.

Possible Problems: None known.

© 2026 UNH–IOL

Test 3.9 – Rebuild Assist via Get LBA Status (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller has properly implemented Get LBA Status functionality if supported.

References:

Old Ref: [1] NVMe Specification 5.27, 8.22
NVM Express NVM Command Set Specification 1.0a : 4.2.1

Resource Requirements:

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

Last Modification: May 4, 2020

Discussion: Potentially Unrecoverable LBAs are LBAs that, when read, may result in the command that caused the media to be read being aborted with Unrecovered Read Error status. The Get LBA Status capability provides the host with the ability to identify Potentially Unrecoverable LBAs. The logical block data is able to be recovered from another location and re-written.

Test Setup: See Appendix A.

Case 1: Get LBA Status (OF-FYI, M)

Test Procedure:

1. Check that Bit 9 of the OACS field in the Identify Controller Data Structure is set to 1, to indicate support for the Get LBA Status capability. If this bit is not set to 1, this test is not applicable.
2. Check that Bit 13 of the OAES field in the Identify Controller Data Structure is set to 1, to indicate support for LBA Status Notification event. If this bit is not set to 1, this test is not applicable.
3. Perform a Get Log Page command for the LBA Status Information log page (0Eh).
4. Perform a Get Feature command for the LBA Status Attributes feature (15h).
5. Perform a Set Feature command for the LBA Status Attributes feature (15h) to set a new value for the LBA Status Information Report Interval (LSIRI).
6. Perform a Get Feature command for the LBA Status Attributes feature (15h).
7. Perform a Get LBA Status command with ATYPE set to 10h.
8. Perform a Get LBA Status command with ATYPE set to 11h.

Observable Results:

1. Verify that the controller indicates support for the Log Page Offset and extended Number of Dwords (i.e. 32 bits rather than 12 bits in the Log Page Attributes field of the Identify Controller data structure.
2. Verify that the Get Log Page operation for the LBA Status Information log page (0Eh) completes successfully.
3. Verify that each Get Feature operation for the LBA Status Attributes feature (15h) completes successfully.
4. Verify that the Get Feature operation following the Set Feature operation for feature 15h indicates the value set by the host.
5. Verify if the Get LBA Status command with ATYPE=0 completes successfully and includes a properly formatted LBA Status Descriptor list and:
    a. the controller returns Untracked LBAs and Tracked LBAs in the range specified in the Get LBA Status command for the namespace specified in the Namespace Identifier (CDW1.NSID);
    b. the controller removes all LBAs in the range specified in the Get LBA Status command, which prior to processing the Get LBA Status command were successfully re-written, from relevant internal data structures (e.g., internal Tracked LBA list).
6. Verify if the Get LBA Status command with ATYPE=0 command did not complete successfully, that it completed with the status of ‘Invalid Field in Command’ (02h).
7. Verify that the Get LBA Status command with ATYPE=1 completes successfully and includes a properly formatted LBA Status Descriptor list and:

© 2026 UNH–IOL

   a. returns Tracked LBAs in the range specified in the Get LBA Status command for the namespace specified in the Namespace Identifier (CDW1.NSID) field;
   b. removes all LBAs in the range specified in the Get LBA Status command, which prior to processing the Get LBA Status command were successfully re-written, from relevant internal data structures (e.g., internal Tracked LBA list);
8. Verify if the Get LBA Status command with ATYPE=1 command did not complete successfully, that it completed with the status of 'Invalid Field in Command' (02h).

Possible Problems: Since the controller identifies which LBAs to track, the polling internal of tracked LBA status is not tested, as there is not a deterministic way to cause LBAs to be tracked.

Case 2: Reporting Allocated LBA Supported (OF-FYI, FYI)

Test Procedure:

1. that Bit 9 of the OACS field in the Identify Controller Data Structure is set to 1, to indicate support for the Get LBA Status capability. If this bit is not set to 1, this test is not applicable.
2. Check that Bit 13 of the OAES field in the Identify Controller Data Structure is set to 1, to indicate support for LBA Status Notification Event. If this bit is not set to 1, this test is not applicable.
3. Configure the NVMe Host to issue an Identify command with CNS 06h (I/O Command Set Specific Identity Controller Data Structure) with CSI 00h for the NVM Command Set. If bit 0 of byte 18 is cleared to zero, this test is not applicable.
4. Configure the NVMe Host to issue an Identify command with CNS 05h (I/O Command Set Specific Identify Namespace) with CSI 00h, and specify a valid namespace and read the Tracked LBA Allocation Granularity value bytes 295:292. If this field is cleared to zero, then the alignment and granularity are not reported, and this test is not applicable.
5. Configure the NVMe Host to issue a Get LBA Status command specifying the same namespace used in step 4, the Range Length set to 0h, and the Action Type field set to 02h.

Observable Results:

1. Verify that the Identify command in step 1 was completed successfully. If it does not complete successfully and the DUT reports NVMe Base version 1.4 or less, this test is not applicable.
2. Verify that the Get LBA Status command in step 5 completes successfully.
3. Verify that the data buffer returned in step 5 has the following information and values.
   a. Bytes 03:00 contain a value greater than 0. If equal to zero, the following fields are not applicable.
   b. Byte 04 contains a value of 2h
   c. Byte 07:05 are cleared to zero.
   d. Starting with byte 08, LBA Status Descriptors are listed.
4. Verify that within the first LBA Status Descriptor:
   a. Byte 07:00 Descriptor Starting LBA (DSLBA) is equal to the first logical block of the LBA range for which this LBA Status Descriptor Entry reports LBA status.
   b. Byte 13 (Status) is set to a value of 010b

Case 3: Descriptors not Returned When no LBAs are Allocated (OF-FYI, 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 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.
5. Configure the NVMe Host to issue a Get LBA Status command specifying the same namespace used in step 4, the Range Length set to 0h, and the Action Type field set to 02h

© 2026 UNH–IOL

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 value of bytes 03:00 in the returned data buffer from step 5 is 0.

Case 4: Get LBA Status - RALBAS=1, ATYPE=2 Reports User Data Changes (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A in order to obtain the I/O Command Set Specific Identify Controller (CNS 06h) Data Structure. If RALBAS is cleared to 0b, then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 1 of TRATTR (Track User Data Changes Support) is cleared to 0h, this test is not applicable.
3. Configure the NVMe host to issue a Get Log command with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS69) is cleared to 0h, this test is not applicable.
4. Configure the NVMe Host to issue a Controller Data Queue command to Controller A with SEL=0h (Create), QT=0h, CNTLID=CNTLID specifying Controller B, CDQSIZE=10h, PC=1b, and a PRP Entry aligned with CC.MPS specifying a portion of host memory to be used as a User Data Migration Queue. Record the value of CDQID from CQE Dword 0.
5. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=1h, and CDQID=CDQID from step 2.
6. Configure the NVMe Host to issue a Write command to any LBA on any namespace associated with Controller B.
7. Configure the NVMe Host to issue a Get LBA Status command to Controller A with ATYPE=2h.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the data buffer returned from the Get LBA Status command reports an LBA Status Descriptor Entry with Status=010b for the LBA written to in step 7.

© 2026 UNH–IOL

Test 3.10 – Improved Performance Parameters (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller operates correctly even when advertised improved performance parameters are not used by the host.

References:

Old Ref: [1] NVMe Specification 5.15, 8.25
NVM Express NVM Command Set Specification 1.0a : 5.8.2

Resource Requirements:

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

Last Modification: May 4, 2020

Discussion: NVMe controllers may require constrained I/O sizes and alignments to achieve the full performance potential. There are a number of optional attributes that the controller uses to indicate these recommendations. If hosts do not follow these constraints, then the controller shall function correctly, but performance may be limited.

Test Setup: See Appendix A.

Case 1: Unaligned Write Command (OF-FYI, FYI)

Test Procedure:

1. Check the Identify Namespace Data Structure NSFEAT bit 4. If this bit is not set to 1, then this test is not applicable.
2. Check the Identify Namespace Data Structure and record the following values:
    a. Namespace Preferred Write Granularity (NPWG)
    b. Namespace Preferred Write Alignment (NPWA)
    c. Namespace Preferred Deallocate Granularity (NPDG)
    d. Namespace Preferred Deallocate Alignment NPDA
3. Perform a Direction Operation for Return Parameters (01h) and record the following values.
    a. Stream Write Size (SWS)
    b. Stream Granularity Size (SGS)
4. Perform a Write command which addresses a number of logical blocks (NLB) which is not a multiple of NPWG and SWS and the SLBA field is not aligned to NPWA.

Observable Results:

1. Verify that Write operation completes successfully.

Case 2: Unaligned Write Uncorrectable Command (OF-FYI, FYI)

Test Procedure:

1. Check the Identify Namespace Data Structure NSFEAT bit 4. If this bit is not set to 1, then this test is not applicable.
2. Check the ONCS field in the Identify Controller Data Structure to determine if the DUT supports the Write Uncorrectable command. If this command is not supported this test is not applicable.
3. Check the Identify Namespace Data Structure and record the following values:
    a. Namespace Preferred Write Granularity (NPWG)
    b. Namespace Preferred Write Alignment (NPWA)
    c. Namespace Preferred Deallocate Granularity (NPDG)
    d. Namespace Preferred Deallocate Alignment NPDA
4. Perform a Direction Operation for Return Parameters (01h) and record the following values.
    a. Stream Write Size (SWS)
    b. Stream Granularity Size (SGS)

© 2026 UNH–IOL

5. Perform a Write Uncorrectable command which addresses a number of logical blocks (NLB) which is not a multiple of NPWG and SWS and the SLBA field is not aligned to NPWA.

Observable Results:

1. Verify that Write Uncorrectable operation completes successfully.

Case 3: Unaligned Write Zeroes Command (OF-FYI, FYI)

Test Procedure:

1. Check the Identify Namespace Data Structure NSFEAT bit 4. If this bit is not set to 1, then this test is not applicable.
2. Check the ONCS field in the Identify Controller Data Structure to determine if the DUT supports the Write Zeroes command. If this command is not supported this test is not applicable.
3. Check the Identify Namespace Data Structure and record the following values:
    a. Namespace Preferred Write Granularity (NPWG)
    b. Namespace Preferred Write Alignment (NPWA)
    c. Namespace Preferred Deallocate Granularity (NPDG)
    d. Namespace Preferred Deallocate Alignment NPDA
4. Perform a Direction Operation for Return Parameters (01h) and record the following values.
    a. Stream Write Size (SWS)
    b. Stream Granularity Size (SGS)
5. Perform a Write Zeroes command which addresses a number of logical blocks (NLB) which is not a multiple of NPWG and SWS and the SLBA field is not aligned to NPWA.

Observable Results:

1. Verify that Write Zeroes operation completes successfully.

Case 4: Unaligned Dataset Management Command (OF-FYI, FYI)

Test Procedure:

1. Check the Identify Namespace Data Structure NSFEAT bit 4. If this bit is not set to 1, then this test is not applicable.
2. Check the ONCS field in the Identify Controller Data Structure to determine if the DUT supports the Dataset Management command. If this command is not supported this test is not applicable.
3. Check the Identify Namespace Data Structure and record the following values:
    a. Namespace Preferred Write Granularity (NPWG)
    b. Namespace Preferred Write Alignment (NPWA)
    c. Namespace Preferred Deallocate Granularity (NPDG)
    d. Namespace Preferred Deallocate Alignment NPDA
4. Perform a Direction Operation for Return Parameters (01h) and record the following values.
    a. Stream Write Size (SWS)
    b. Stream Granularity Size (SGS)
5. Perform a Dataset Management command, with AD bit set to 1, which addresses a number of logical blocks (NLB) which is not a multiple of NPDG, and the start of each range is not aligned to NPDA and SGS logical blocks.

Observable Results:

1. Verify that Dataset Management operation completes successfully.

Possible Problems: None known.

Case 5: Performance Valid Values (OF-FYI, M)

Test Procedure:

© 2026 UNH–IOL

1. Retrieve a list of all active NSIDs using the Identify command with CNS 02h.
2. For each active namespace in the NVM subsystem, configure the NVMe Host to issue an Identify command using CNS 00h to receive the Identify Namespace data structure for each active namespace.
3. Perform an Identify command with CNS 05h and CSI of 00h to retrieve the NVM command set specific Identify Namespace Data Structure.
4. Read and save values for NPRG, NPRA, and NORS.

Observable Results:

1. NPRG, NPRA, and NORS are non-zero values.
2. Verify that NPRG is less than MDTS.
3. Verify that NORS is less than MTDS and is a multiple of NPRG.
4. If OPTRPERF is 0, that all previous fields are set to 0.

Case 6: Optimal Write Performance Value Fields Supported (OF-FYI, M)

Test Procedure:

1. Retrieve a list of all active NSIDs using the Identify command with CNS 02h.
2. For each active namespace in the NVM subsystem, configure the NVMe Host to issue an Identify command using CNS 00h to receive the Identify Namespace data structure for each active namespace.
3. Perform an Identify command with CNS 05h and CSI of 00h to retrieve the NVM command set specific Identify Namespace Data Structure.
4. Read the value of Optional Write Performance (OPTPERF), composed of bits 4 and 5 in the Namespace Features field.

Observable Results:

1. Verify that reserved fields are set to 0, based on the value of the Optional Write Performance (OPTPERF).

Case 7: Optimal Read Performance Ignored, OPTRPERF=1 (OF-FYI, FYI)

Test Procedure:

1. Retrieve a list of all active NSIDs using the Identify command with CNS 02h.
2. For each active namespace in the NVM subsystem, configure the NVMe Host to issue an Identify command using CNS 00h to receive the Identify Namespace data structure for each active namespace.
3. Perform an Identify command with CNS 05h and CSI of 00h to retrieve the NVM command set specific Identify Namespace Data Structure.
4. Check the value of Optional Write Performance (OPTRPERF) bit 7 in NSFEAT, if set to 0 this test is not applicable.
5. Read and save the values for NPRG, NPRA, and NORS.
6. Send an NVM Read command with an NLB that is not a multiple of NPRG and exceeds NORS, and with an SLBA that is not aligned with the NPRA field.

Observable Results:

1. Verify that the NVM Read command completes successfully when using non-optimal performance parameters.

Case 8: Optimal Read Performance, OPTRPERF=0 (OF-FYI, M)

Test Procedure:

1. Retrieve a list of all active NSIDs using the Identify command with CNS 02h.
2. For each active namespace in the NVM subsystem, configure the NVMe Host to issue an Identify command using CNS 00h to receive the Identify Namespace data structure for each active namespace.
3. Read the value of Optional Read Performance (OPTRPERF)bit 7 of NSFEAT, composed of bits 4 and 5 in the Namespace Features field.
4. Perform an Identify command with CNS 05h and CSI of 00h to retrieve the NVM command set specific Identify Namespace Data Structure.

© 2026 UNH–IOL

Observable Results:

1. Verify that all reserved fields (NPRG, NPRA, NORS) are set to 0

© 2026 UNH–IOL

Test 3.11 – Controller Memory Buffer (IP, IP)

Purpose: To verify that an NVMe Controller correctly implements controller memory buffer if supported.

References:

Old Ref: [1] NVMe Specification 4.7
NVM Express Base Specification 2.0a : 8.5

Resource Requirements:

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

Last Modification: May 18, 2020

Discussion: The Controller Memory Buffer (CMB) is a region of general purpose read/write memory on the controller. The controller indicates support for the CMB by setting CAP.CMBS to '1'. The host indicates intent to use the CMB by setting CMBMSC.CRE to '1'. Once this bit is set to '1', the controller indicates the properties of the CMB via the CMBLOC and CMBSZ registers.

Test Setup: See Appendix A.

Case 1: CMB Supported (IP, IP)

Test Procedure:

1. Check CAP.CMBS. If CAP.CMBS=0, then this test is not applicable.
2. The Testing Station acting as a Host should set CMBMSC.CBA to a valid value.
3. The Testing Station acting as a Host should set CMBMSC.CRE to 1h.
4. Read the CMBLOC and CMBSZ registers. Record the value of CMBSZ.LISTS and CMBSZ.SQS. If either of these fields in CMBSZ are cleared to '0h', this test is unapplicable.
5. Perform a WRITE operation of a known data pattern to the CMB using the parameters in CMBLOC and CMBSZ.
6. Perform a READ operation to the CMB using the parameters in CMBLOC and CMBSZ.

Observable Results:

1. Verify that WRITE operation completes successfully.
2. Verify that READ operation completes successfully and that the DUT returns the same data written in the previous WRITE operation.

Possible Problems: None known.

© 2026 UNH–IOL

Test 3.12 – NVM Sets (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller correctly implements NVM Sets if supported.

References:

Old Ref: [1] NVMe Specification 4.9
NVM Express Base Specification 2.0a : 3.2.2

Resource Requirements:

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

Last Modification: December 1, 2019

Discussion: An NVM Set is a collection of NVM that is separate (logically and potentially physically) from NVM in other NVM Sets. One or more namespaces may be created within an NVM Set and those namespaces inherit the attributes of the NVM Set. A namespace is wholly contained within a single NVM Set and shall not span more than one NVM Set.

Test Setup: See Appendix A.

Case 1: NVM Sets Supported (OF-FYI, FYI)

Test Procedure:

1. Check Bit 2 (NVM Sets) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is not set to 1h, then this test is not applicable.
2. Record the value of Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure.
3. Perform an Identify Command to the DUT using CNS 00h (Identify Namespace Data Structure).
4. Perform an Identify Command to the DUT using CNS 04h (NVM Set List).
5. Perform a Namespace Management command with operation Create Namespace using the NVM Set Identifier.

Observable Results:

1. Verify that the DUT returns an NVM Set List with properly formatted NVM Set Attributes Entries in response to the Identify Command to CNS 04h, including the associated Endurance Group.
2. Verify that the DUT indicates the NVM Set Identifier with which the namespace is associate in the response to the Identify Command to CNS 00h.
3. Verify that the DUT indicates support for Endurance Groups by checking Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure.
4. Verify that the Namespace Management command completes successfully.

Case 2: RLL Supported in NVM Set (OF-FYI, FYI)

Test Procedure:

1. Check Bit 2 (NVM Sets) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is not set to 1, then this test is not applicable.
2. Check Bit 3 (Read Recovery Level) of the CTRATT field of the Identify Controller Data Structure. If Bit 3 is not set to 1h, then this test is not applicable.
3. Record the value of Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure.
4. Perform an Identify Command to the DUT using CNS 00h (Identify Namespace Data Structure).
5. Perform an Identify Command to the DUT using CNS 04h (NVM Set List).
6. Perform a Namespace Management command with operation Create Namespace using the NVM Set Identifier.
7. Perform a Set Feature command for the Read Recovery Level Feature for a specific NVM Set Identifier.

Observable Results:

© 2026 UNH–IOL

1. Verify that the DUT returns an NVM Set List with properly formatted NVM Set Attributes Entries in response to the Identify Command to CNS 04h, including the associated Endurance Group.
2. Verify that the DUT indicates the NVM Set Identifier with which the namespace is associate in the response to the Identify Command to CNS 00h.
3. Verify that the DUT indicates support for Endurance Groups by checking Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure.
4. Verify that the Namespace Management command completes successfully.
5. Verify that the Set Feature command for the Read Recovery Level Feature completes successfully.

Case 3: Endurance Group Info Log Matches SMART/Health Log Summary (OF-FYI, M)

Test Procedure:

1. Perform an Identify Controller Data Structure Command (CNS=01h).
2. Check Bit 2 (NVM Sets) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is not set to 1, then this test is not applicable.
3. Check Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure. If Bit 4 is not set to 1, then this test is not applicable.
4. Perform an Identify Namespace Data Structure Command (CNS=00h) for all attached Namespaces. Record the Endurance Group Identifier (ENDGID) for the given Namespaces.
5. Perform Get Log Page for the Endurance Group Info Log (09h). Repeat this for all recorded Endurance Group Identifiers.
6. Perform Get Log Page for the SMART/Health Log (02h) for the Controller.

Observable Results:

1. Verify that any bits set to 1 in the Critical Warning field of any retrieved Endurance Group Info Log Pages are also set to 1 in the Endurance Group Critical Warning Summary field of the SMART/Health Log Page.

Case 4: Endurance Group Identifier = 0 (OF-FYI, M)

Test Procedure:

1. Perform an Identify Controller Data Structure Command (CNS=01h).
2. Check Bit 2 (NVM Sets) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is not set to 1h, then this test is not applicable.
3. Check Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure. If Bit 4 is not set to 1h, then this test is not applicable.
4. Perform an Identify Namespace Data Structure Command (CNS=00h) for all attached Namespaces. Record the Endurance Group Identifier (ENDGID) for the given Namespaces.
5. Perform a Get Log Page for the Endurance Group Info Log (09h) using an Endurance Group Identifier of 0h.

Observable Results:

1. Verify that the Get Log Page command is aborted with status ‘Invalid Field in Command’ (02h).

Case 5: Endurance Group Identifier Does not Exist for Set Feature (OF-FYI, M)

Test Procedure:

1. Perform an Identify Controller Data Structure Command (CNS=01h).
2. Check Bit 2 (NVM Sets) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is not set to 1h, then this test is not applicable.
3. Check Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure. If Bit 4 is not set to 1h, then this test is not applicable.
4. Perform an Identify Namespace Data Structure Command (CNS=00h) for all attached Namespaces. Record the Endurance Group Identifier (ENDGID) for the given Namespaces.
5. Perform a Set Feature command for the Endurance Group Event Configuration (18h) using an Endurance Group Identifier that does not exist.

© 2026 UNH–IOL

Observable Results:

1. Verify that the Set Feature command is aborted with status 'Invalid Field in Command' (02h).

Case 6: Endurance Group Identifier Does not Exist for Get Feature (OF-FYI, M)

Test Procedure:

1. Perform an Identify Controller Data Structure Command (CNS=01h).
2. Check Bit 2 (NVM Sets) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is not set to 1h, then this test is not applicable.
3. Check Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure. If Bit 4 is not set to 1h, then this test is not applicable.
4. Perform an Identify Namespace Data Structure Command (CNS=00h) for all attached Namespaces. Record the ENDGIDMAX for the given Namespaces.
5. Perform a Get Feature command for the Endurance Group Event Configuration (18h) using an Endurance Group Identifier = ENDGIDMAX + 1.

Observable Results:

1. Verify that the Get Feature command is aborted with status 'Invalid Field in Command' (02h).

Case 7: Endurance Group Critical Warnings Configuration with Reserved Field (OF-FYI, M)

Test Procedure:

1. Perform an Identify Controller Data Structure Command (CNS=01h).
2. Check Bit 2 (NVM Sets) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is not set to 1h, then this test is not applicable.
3. Check Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure. If Bit 4 is not set to 1h, then this test is not applicable.
4. Perform an Identify Namespace Data Structure Command (CNS=00h) for all attached Namespaces. Record the Endurance Group Identifier (ENDGID) for the given Namespaces.
5. Perform Set Feature command for the Endurance Group Event Configuration (18h) using a valid Endurance Group Identifier, and Bit 23 set to 1h. Bit 23 should correspond to Bit 7 in the Critical Warning Field of the Endurance Group Log, which is reserved in NVMe specification v1.4.

Observable Results:

1. Verify that the Set Feature command is aborted with status 'Invalid Field in Command' (02h).

Case 8: Endurance Groups not Supported (OF-FYI, M)

Test Procedure:

1. Perform an Identify Controller Data Structure Command (CNS=01h).
2. Check Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure. If Bit 4 is set to 1h, then this test is not applicable.
3. Perform a Set feature Command for the Asynchronous Event Configuration Feature (FID=0Bh) with the Endurance Group Event Aggregate Log Change Notices bit set to 1h.

Observable Results:

1. Verify that the Set Feature command is aborted with status 'Invalid Field in Command' (02h).

Case 9: NVM Sets Supported (OF-FYI, M)

Test Procedure:

1. Check Bit 2 (NVM Sets) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is not set to 1, then this test is not applicable.
2. Perform an Identify Command to the DUT using CNS 04h (NVM Set List).

© 2026 UNH–IOL

3. Perform an Identify Command to the DUT using CNS 00h (Identify Namespace Data Structure) for a Namespace that is associated with an NVM Set.

Observable Results:

1. Verify that the NOWS value returned in the Identify Namespace Data Structure is set to the Optimal Write Size field returned in the NVM Set Attributes Entry for the NVM Set with which this namespace is associated.

Possible Problems: None known

Case 10: Endurance Group Info Log Matches SMART/Health Log Summary (OF-FYI, M)

Test Procedure:

1. Perform an Identify Controller Data Structure Command (CNS=01h).
2. Check Bit 2 (NVM Sets) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is not set to 1, then this test is not applicable.
3. Check Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure. If Bit 4 is not set to 1, then this test is not applicable.
4. Perform an Identify Namespace Data Structure Command (CNS=00h) for all attached Namespaces. Record the Endurance Group Identifier (ENDGID) for the given Namespaces.
5. Perform Get Log Page for the Endurance Group Info Log (09h). Repeat this for all recorded Endurance Group Identifiers.
6. Perform Get Log Page for the SMART/Health Log (02h) for the Controller.

Observable Results:

1. Verify that any bits set to 1 in the Critical Warning field of any retrieved Endurance Group Info Log Pages are also set to 1 in the Endurance Group Critical Warning Summary field of the SMART/Health Log Page.

Case 11: NVM Set List, no zero Entry (OF-FYI, M)

Test Procedure:

1. Check Bit 2 (NVM Sets) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is not set to 1, then this test is not applicable.
2. Check Bit 4 (Endurance Groups) of the CTRATT field of the Identify Controller Data Structure. If Bit 4 is not set to 1, then this test is not applicable.
3. Perform an Identify NVM Set List (CNS=04h).

Observable Results:

1. Verify that none of the NVM Set Attributes Entry data areas are cleared to ‘0’

© 2026 UNH–IOL

Test 3.13 – Read Recovery Level (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller correctly implements NVM Sets if supported.

References:

Old Ref: [1] NVMe Specification 8.16
NVM Express Base Specification 2.0a : 8.17

Resource Requirements:

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

Last Modification: May 4, 2020

Discussion: The Read Recovery Level (RRL) is a NVM Set configurable attribute that balances the completion time for read commands and the amount of error recovery applied to those read commands. The Read Recovery Level applies to an NVM Set with which the Read Recovery Level is associated. A namespace created within an NVM Set inherits the Read Recovery Level of that NVM Set. If NVM Sets are not supported, all namespaces in the NVM subsystem use an identical Read Recovery Level.

Test Setup: See Appendix A.

Case 1: Proper RLL Levels Supported (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Controller Data Structure command.
2. Check Bit 3 (Read Recovery Level) of the CTRATT field of the Identify Controller Data Structure. If Bit 3 is not set to 1, then this test is not applicable.
3. Perform a WRITE operation to a given namespace writing a known Data pattern.
4. Perform a READ operation to the same namespace as was written in the previous step, ensuring that the known pattern was written.
5. Perform a Get Feature operation to FID 12h to query the current Read Recovery Level in the same namespace.
6. Perform a Set Feature operation to FID 12h change the Read Recovery Level to another supported value, in the same namespace, as indicated in the Read Recovery Levels Supported field in the Identify Controller data structure.
7. Perform a Get Feature operation to FID 12h to query the current Read Recovery Level, verifying that the previously supplied value is returned.
8. Perform a READ operation to the same namespace, ensuring that the previously written known pattern is still present.

Observable Results:

1. Verify the DUT indicates support for at least Read Recovery Levels 4 and 15 in the Read Recovery Levels Supported field in the Identify Controller data structure.

Case 2: NVM Sets Not Supported (OF-FYI, FYI)

Test Procedure:

1. Check Bit 3 (Read Recovery Level) of the CTRATT field of the Identify Controller Data Structure. If Bit 3 is not set to 1, then this test is not applicable.
2. Record the RRLS value in the Identify Controller Data Structure.
3. Check Bit 2 (NVM Sets) of the CTRATT field of the Identify Controller Data Structure. If Bit 2 is set to 1 (NVM Sets supported), then this test is not applicable.
4. Perform a Get Feature Command to FID 12h to retrieve the support RRL.

Observable Results:

© 2026 UNH–IOL

1. Verify that the RRL value returned in response to the Get Feature command matched the RRLS values reported in the Identify Controller Data Structure.

Possible Problems: None known.

© 2026 UNH–IOL

Test 3.14 – Asymmetric Namespace Access Reporting (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller correctly implements Asymmetric Namespace Access if supported.

References:

Old Ref: [1] NVMe Specification 5.14.1.12, 8.20
NVM Express Base Specification 2.0a : 8.1, 8.10

Resource Requirements:

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

Last Modification: December 1, 2019

Discussion: Asymmetric Namespace Access (ANA) occurs in environments where namespace access characteristics (e.g., performance or ability to access the media) may vary based on the controller used to access the namespace (e.g., Fabrics) and the internal configuration of the NVM subsystem. Asymmetric Namespace Access Reporting is used to indicate to the host information about those access characteristics.

Test Setup: See Appendix A.

Case 1: ANA Log Page RGO=0 (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Controller Data Structure command. Record Bit 3 of the CMIC field. If Bit 3 is set to 0, this test is not applicable.
2. Perform a Get Log Page for Log Page ID 0Ch with the RGO bit set to 0.

Observable Results:

1. Verify that the Get Log Page command for Log Page ID 0Ch completed successfully.
2. Verify that the returned LPOL and LPOU fields are cleared to 0.
3. Verify that returned ANA Group Descriptors contain the Namespace Identifiers of attached namespaces that are members of the ANA Group described by that ANA Group Descriptor and the Number of NSID Values field set to the number of Namespace Identifier values in that ANA Group Descriptor.

Case 2: ANA Log Page RGO=1 (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Controller Data Structure command. Record Bit 3 of the CMIC field. If Bit 3 is set to 0, this test is not applicable.
2. Perform a Get Log Page for Log Page ID 0Ch with the RGO bit set to 1.

Observable Results:

1. Verify that the Get Log Page command for Log Page ID 0Ch completed successfully, and that Log Page returned included ANA Group Descriptors with the Number of NSID value field in each ANA Group Descriptor cleared to 0h.

Case 3: Single AER for Namespace Attach Change Event (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Controller Data Structure command. Record Bit 3 of the CMIC field. If Bit 3 is set to 0, this test is not applicable.
2. Enable Asynchronous Event Reporting using the Set Feature command for FID 0x0B with bits 11 (Asymmetric Namespace Access Change Notices) and 8 (Namespace Attribute Notices) set to 1.
3. Issue an Asynchronous Event Request command to enable the DUT to report events.

© 2026 UNH–IOL

4. Perform a Namespace Attachment command to attach a new namespace.

Observable Results:

1. Verify that the DUT sent either a Asymmetric Namespace Access Change event notice or a Namespace Attribute changed notice after the Namespace Attach command, but not both.

Case 4: Single AER for Namespace Deletion Change Event (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Controller Data Structure command. Record Bit 3 of the CMIC field. If Bit 3 is set to 0, this test is not applicable.
2. Enable Asynchronous Event Reporting using the Set Feature command for FID 0x0B with bits 11 (Asymmetric Namespace Access Change Notices) and 8 (Namespace Attribute Notices) set to 1.
3. Issue an Asynchronous Event Request command to enable the DUT to report events.
4. Perform a Namespace Delete command to delete a namespace. It may be necessary to create a namespace first depending on the state of the DUT.

Observable Results:

1. Verify that the DUT sent either a Asymmetric Namespace Access Change event notice or a Namespace Attribute changed notice after the Namespace Deletion command, but not both.

Case 5: Single AER for Namespace Detachment Change Event (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Controller Data Structure command. Record Bit 3 of the CMIC field. If Bit 3 is set to 0, this test is not applicable.
2. Enable Asynchronous Event Reporting using the Set Feature command for FID 0x0B with bits 11 (Asymmetric Namespace Access Change Notices) and 8 (Namespace Attribute Notices) set to 1.
3. Issue an Asynchronous Event Request command to enable the DUT to report events.
4. Perform a Namespace Detachment command to detach a namespace. It may be necessary to create and attach namespace first depending on the state of the DUT.

Observable Results:

1. Verify that the DUT sent either a Asymmetric Namespace Access Change event notice or a Namespace Attribute changed notice after the Namespace Detachment command, but not both.

Case 6: No Namespace attached, Get Log Page ANA (OF-FYI, FYI)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management then this test is not applicable.
2. Perform an Identify Controller Data Structure command. Record Bit 3 of the CMIC field. If Bit 3 is set to 0, this test is not applicable.
3. Configure the NVMe Host to issue an Identify command specifying CNS field value 02h and NSID value 01h to the controller in order to receive back a Namespace List containing active namespace IDs attached to the controller.
4. Detach all Namespaces attached to the Controller under test by configure the NVMe Host to issue a Namespace Attachment command, specifying Select field value 1h (Detach).
5. Perform a Get Log Page for Asymmetric Namespace Access Log (0Ch).

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 there are no namespaces attached to the Controller under test after step 4.

© 2026 UNH–IOL

3. Verify that the Number of ANA Group Descriptors is cleared to 0h.

Possible Problems: None known.

Case 7: Asymmetric Namespace Access - IOS=1, OT=1 (OF-FYI, FYI)

Test Procedure:

1. Issue a Get Log Page command with LID 0h to retrieve the Supported Log Pages Log Page. If the Index Offset Supported (IOS) bit is cleared to 0 for LID 0Ch then this test is not applicable.
2. Issue a Get Log page command with LID 0Ch and OT=0 to retrieve the Asymmetric Namespace Access Log Page and record its contents.
3. Issue a Get Log Page command with LID 0Ch, OT=1, and LPOU & LPOL configured to 1 to retrieve ANA Group Descriptor 0.

Observable Results:

1. Verify the Get Log Page commands complete successfully
2. Verify the first 16 bytes returned in the Get Log Page commands from steps 2 and 3 do not match.

© 2026 UNH–IOL

Test 3.15 – Namespace Granularity (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller correctly implements Namespace Granularity if supported.

References:

Old Ref: [1] NVMe Specification 8.12
NVM Express Base Specification 2.0a : 5.17.2.15
NVM Express NVM Command Set Specification 1.0a : 4.1.5.6

Resource Requirements:

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

Last Modification: December 1, 2019

Discussion: The size granularity and the capacity granularity are hints which may be used by the host to minimize the capacity that is allocated for a namespace and that is not able to be addressed by logical block addresses. The granularities are used in specifying values for the Namespace Size (NSZE) and Namespace Capacity (NCAP) fields of the data structure used for the create operation of the Namespace Management command.

Test Setup: See Appendix A.

Case 1: Create Namespace with all capacity Allocated (OF-FYI, M)

Test Procedure:

1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure CTRATT Bit 7 to determine if the product under test supports Namespace Granularity. If the product under test does not support Namespace Granularity (Bit 7 = 1) this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 16h and a valid value for Controller ID in CDW10.CNTID.
3. Perform a Namespace Management command to Create a namespace with the following conditions
    a. The Product of NSZE and the Formatted LBA Size is an integral multiple of the Namespace Size Granularity.
    b. The Product of NCAP and the Formatted LBA Size is an integral multiple of the Namespace Size Granularity.
    c. NSZE is equal to NCAP.
4. Attached the Namespace created in the previous step.
5. Perform at least 1 WRITE and READ operation to the Namespace.
6. Detach and Delete the namespace.

Observable Results:

1. Verify that the Identify command for CNS=16h completes successfully and includes a properly formatted Namespace Granularity List.
2. Verify that the namespace granularity descriptors with an index greater than the value in the Number of Descriptors field are cleared to 0h.
3. Verify that the Namespace Management Create command completes successfully.
4. Verify that the WRITE, READ, Namespace Detach, and Namespace Delete commands complete successfully.

Case 2: Create Namespace with not all capacity Allocated (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure CTRATT Bit 7 to determine if the product under test supports Namespace Granularity. If the product under test does not support Namespace Granularity (Bit 7 = 1) this test is not applicable.

© 2026 UNH–IOL

2. Configure the NVMe Host to issue an Identify command specifying CNS value 16h and a valid value for Controller ID in CDW10.CNTID.
3. Perform a Namespace Management command to Create a namespace with the following conditions
    a. The Product of NSZE and the Formatted LBA Size is not an integral multiple of the Namespace Size Granularity.
    b. The Product of NCAP and the Formatted LBA Size is not an integral multiple of the Namespace Size Granularity.
    c. NSZE is not equal to NCAP.
4. Attached the Namespace created in the previous step.
5. Perform at least 1 WRITE and READ operation to the Namespace.
6. Detach and Delete the namespace.

Observable Results:

1. Verify that the Identify command for CNS=16h completes successfully and includes a properly formatted Namespace Granularity List.
2. Verify that the namespace granularity descriptors with an index greater than the value in the Number of Descriptors field are cleared to 0h.
3. Verify that the Namespace Management Create command completes successfully.
4. Verify that the WRITE, READ, Namespace Detach, and Namespace Delete commands complete successfully.

Possible Problems: None known.

© 2026 UNH–IOL

Test 3.16 – SQ Associations (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller correctly implements SQ Associations if supported.

References:

Old Ref: [1] NVMe Specification 8.23
NVM Express Base Specification 2.0a : 8.3.2

Resource Requirements:

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

Last Modification: December 1, 2019

Discussion: The SQ Associations capability provides hints to the controller as to which specific I/O Queues are associated with a given NVM Set. The controller uses this information to further enhance performance when Predictable Latency Mode is enabled. A controller which supports SQ Associations shall also support NVM Sets and Predictable Latency Mode.

Test Setup: See Appendix A.

Case 1: Associated Features Supported (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure CTRATT Bit 8 to determine if the product under test supports SQ Associations. If the product under test does not support SQ Associations (Bit 8 = 1) this test is not applicable.

Observable Results:

1. Verify that CTRATT Bit 2 (NVM Sets) is set to 1.
2. Verify that CTRATT Bit 5 (Predictable Latency Mode) is set to 1.

Possible Problems: None known.

© 2026 UNH–IOL

Test 3.17 – Transport SGL Data Block Descriptor (OF-FYI)

Purpose: To verify that an NVMe Controller correctly implements SGL Data Block Descriptor if supported.

References:

[1] NVMe Specification 4.2, 4.4, 5.15
NVM Express Base Specification 2.0a : 3.3.3.1, 4.1.2

Resource Requirements:

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

Last Modification: August 1, 2025

Discussion: A controller may support byte or dword alignment and granularity of Data Blocks. If a controller supports only dword alignment and granularity as indicated in the SGL Support field of the Identify Controller data structure, then the values in the Address and Length fields of all Data Block descriptors shall have their lower two bits cleared to 00b. This requirement applies to Data Block descriptors that indicate data and/or metadata memory regions. This test only applies to devices using a Fabrics transport.

Test Setup: See Appendix A.

Case 1: Correct Descriptor Format (OF-FYI)

Test Procedure:

1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure SGLS Bit 21 to determine if the product under test supports the Transport SGL Data Block Descriptor. If the product under test does not support the Transport SGL Data Block Descriptor (Bit 21 = 1) this test is not applicable.
2. Record Bits 01:00 of the SGLS field.
    a. If this field equals 01b. then SGLs are supported, and there are no alignment or granularity requirement for Data Blocks.
    b. If this field equals 10b then SGLs are supported, and there is a dword alignment and granularity requirement for Data Blocks.
3. Perform an NVMe Connect command with a value valid for the transport being used.
    a. RDMA: Sub type values Ah to Eh are reserved, value Fh is used with the Keyed SGL Data Block Descriptor Type (4h).
    b. FC: Ah is the only value.
    c. TCP: 1h and Ah are the only valid values.

Observable Results:

1. Verify that if bits 01:00 of the SGLS field is set to 10b (DUT has dword alignment and granularity requirements), then the lower 2 bits of the Transport SGL Data Block Descriptor Length field are cleared to 00b.
2. Verify that the SGL Descriptor Type Field returned in the Transport SGL Data Block Descriptor is not set to a reserved value.
3. Verify that the connect command completed successfully.

Case 2: Incorrect Descriptor Format (OF-FYI)

Test Procedure:

1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure SGLS Bit 21 to determine if the product under test supports the Transport SGL Data Block Descriptor. If the product under test does not support the Transport SGL Data Block Descriptor (Bit 21 = 1) this test is not applicable.
2. Record Bits 01:00 of the SGLS field.
    a. If this field equals 01b. then SGLs are supported, and there are no alignment or granularity requirement for Data Blocks.

© 2026 UNH–IOL

   b. If this field equals 10b then SGLs are supported, and there is a dword alignment and granularity requirement for Data Blocks.
3. Perform an NVMe Connect command with a value which is invalid for the transport being used.
   a. RDMA: Sub type values Ah to Eh are reserved, value Fh is used with the Keyed SGL Data Block Descriptor Type (4h).
   b. FC: Ah is the only value.
   c. TCP: 1h and Ah are the only valid values.

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

Case 3: SGL Bit Bucket Descriptor Type (OF-FYI)

Test Procedure:
1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure SGLS Bit 16 to determine if the product under test supports the SGL Bit Bucket Descriptor. If the product under test does not support the SGL Bit Bucket Descriptor (Bit 16 = 1) this test is not applicable.
2. Perform an NVMe Connect command with the following values:
   a. SGL Descriptor Type = 1h (SGL Bit Bucket Descriptor)
   b. SGL Descriptor SubType = 1h (Offset)

Observable Results:
1. Verify that the command is aborted with status SGL Descriptor Type Invalid.

Case 4: Keyed SGL Data Block Descriptor Type (OF-FYI)

Test Procedure:
1. Perform an Identify Command to CNS=01h, then check the Identify Controller Data Structure SGLS Bit 02 to determine if the product under test supports the Keyed SGL Data Block Descriptor. If the product under test does not support the Keyed SGL SGL Data Block Descriptor (Bit 02 = 1) this test is not applicable.
2. Perform an NVMe Connect command with the following values:
   a. SGL Descriptor Type = 4h (Keyed SGL Data Block Descriptor)
   b. SGL Descriptor SubType = 1h (Offset)

Observable Results:
1. Verify that the command is aborted with status SGL Descriptor Type Invalid.

Possible Problems: This test is only described for NVMe-oF products.

Case 5: Data Transfer Size Exceeds Total Metadata SGL Length (IP)

Test Procedure:
1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address of the DUT and is able to send packets to the DUT.
2. Configure the Testing Station to issue an Identify Controller command and verify the SGL Support (SGLS) field is not set to 00b. If this field is set to 00b then this test is not applicable.
3. Verify the MPTR SGL Descriptor Support (MSDS) bit is set to 1 in the Identify Controller data structure. If it is not then this test is not applicable.
4. Configure the Testing Station to issue an Identify Namespace command and verify that the Metadata Size (MS) field is not set to 0. If metadata is not supported then this test is not applicable.
5. Verify the Metadata Transferred as Separate Buffer Support (MTSBS) field is set to 1, if it is not supported then this test is not applicable.

© 2026 UNH–IOL

6. Configure the Testing Station to issue a Read command specifying an LBA in a namespace which has been formatted to allow metadata via a separate buffer, with the SGL Descriptor for the metadata having the Length (LEN) field set to 1.

Observable Results:

1. Verify the Read command fails with a status code of Metadata SGL Length Invalid.

Case 6: Data Transfer Size Exceeds Total SGL Length (OF-FYI, FYI)

Test Procedure:

1. Configure the Testing Station to issue an Identify Controller command to record the value of the SGL Support (SGLS) field. Record the SGL types that are supported by the controller. If the SGLS field is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Write command to the controller in order to write a known data pattern to one LBA and NLB set to 1 on the NVMe Device. Configure the Length (LEN) field in the SGL descriptor to be set to 1 to ensure the Host is attempting to Write more data than is expected in the SGLs requested data transfer length.
    a. Set the PSDT bit to 10b in CDW0 to specify the use of SGLs
    b. Use one of the listed supported SGL types as indicated in the SGLS field in the Identify Controller Data Structure returned in step 1.

Observable Results:

1. Verify the Write command fails with a status code of Data SGL Length Invalid.
2. Verify the SGL Support (SGLS) field in the Identify Controller data structure is not set to 00b for NVMe over Fabrics connections.

© 2026 UNH–IOL

Group 4: Controller Registers

Overview:
This section describes a method for performing conformance verification for NVMe products implementing the NVMe Controller Registers.

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).

© 2026 UNH–IOL

Test 4.1 – Offset 00h: CAP – Memory Page Size Maximum (MPSMAX) (OF, M)

Purpose: To validate the MPSMAX field of the Controller Capabilities (CAP) register of an NVMe Controller.

References:

Old Ref: [1] NVMe Specification 3.1.1
NVM Express Base Specification 2.0a : 3.1.3.1

Resource Requirements:

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

Last Modification: June 24, 2015

Discussion: This field indicates the maximum host memory page size that the controller supports. The maximum memory page size is (2 ^ (12 + MPSMAX)) (i.e. $2^{12+MPSMAX}$). Therefore, the maximum memory page size which a controller may support is 128MB. The host shall not configure a memory page size in CC.MPS that is larger than this value.

Test Setup: See.

Test Procedure:

1. Configure the NVMe Host to read the CAP.MPSMAX register field (bits 55:52) of the NVMe Controller.

Observable Results:

1. Verify that the value of the CAP.MPSMAX register field is greater than or equal to the value of the CAP.MPSMIN register field.

Possible Problems: None known

© 2026 UNH–IOL

Test 4.2 – Offset 00h: CAP – Memory Page Size Minimum (MPSMIN) (OF, M)

Purpose: To validate the MPSMIN field of the Controller Capabilities (CAP) register of an NVMe Controller.

References:

Old Ref: [1] NVMe Specification 3.1.1
NVM Express Base Specification 2.0a : 3.1.3.1

Resource Requirements:

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

Last Modification: June 24, 2015

Discussion: This field indicates the minimum host memory page size that the controller supports. The minimum memory page size is (2 ^ (12 + MPSMIN)) (i.e. $2^{12+MPSMIN}$). Therefore, the minimum memory page size which a controller may support is 4KB. The host shall not configure a memory page size in CC.MPS that is smaller than this value.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to read the CAP.MPSMIN register field (bits 51:48) of the NVMe Controller.

Observable Results:

1. Verify that the value of the CAP.MPSMAX register field is greater than or equal to the value of the CAP.MPSMIN register field.

Possible Problems: None known.

© 2026 UNH–IOL

Test 4.3 – Offset 00h: CAP – Command Sets Supported (CSS) (OF, M)

Purpose: To validate the Command Sets Supported (CSS) field of the Controller Capabilities (CAP) register of an NVMe Controller.

References:

Old Ref: [1] NVMe Specification 3.1.1
NVM Express Base Specification 2.0a : 3.1.3.1

Resource Requirements:

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

Last Modification: January 4, 2021

Discussion: This field indicates the I/O Command Set(s) that the controller supports. A minimum of one command set shall be supported. The field is bit significant (bit definitions can be found in Table 7). If a bit is set to '1', then the corresponding I/O Command Set is supported. If a bit is cleared to '0', then the corresponding I/O Command Set is not supported.

Table 7 – Command Sets Supported Bit Definitions

Bit	Definition
37	NVM command set
38	Reserved
39	Reserved
40	Reserved
41	Reserved
42	Reserved
43	Controller Supports one or more I/O Command Sets
44	No I/O Command Set is Supported.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to read the CAP.CSS register field (bits 37:44) of the NVMe Controller.
2. Configure the NVMe host to retrieve the device under test version.

Observable Results:

1. Verify that Bit 37 of the CAP.CSS register field is set to '1' to indicate that the NVM Command Set is supported by the controller. Controllers that support the NVM Command Set shall set this bit even if bit 43 is set to '1'.
2. If the version of the device under test is 2.0 or greater then verify that bit 7 of CSS is cleared to 0h.

Possible Problems: None known.

© 2026 UNH–IOL

Test 4.4 – Offset 00h: CAP – Doorbell Stride (DSTRD) (OF, M)

Purpose: To validate the Doorbell Stride (DSTRD) field of the Controller Capabilities (CAP) register of an NVMe Controller.

References:

Old Ref: [1] NVMe Specification 3.1.1
NVM Express Base Specification 2.0a : 3.1.3.1

Resource Requirements:

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

Last Modification: March 2, 2016

Discussion: Each Submission Queue and Completion Queue Doorbell register is 32–bits in size. This register indicates the stride between doorbell registers. The stride is specified as (2 ^ (2 + DSTRD)) (i.e. $2^{2+DSTRD}$) in bytes. A value of 0h indicates a stride of 4 bytes, where the doorbell registers are packed without reserved space between each register.

Since there is no means to validate the value stored in this register field, this test is designed purely to determine the value the NVMe Controller returns when the register is read, and is therefore considered an informative test.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to read the CAP.DSTRD register field (bits 35:32) of the NVMe Controller.
2. Report the value of the CAP.DSTRD register field.

Observable Results: None.

Possible Problems: There are no required values for the CAP.DSTRD register field at any given time and so this test is run as informative with no pass/fail criteria.

© 2026 UNH–IOL

Test 4.5 – Offset 00h: CAP – Timeout (TO) (OF, M)

Purpose: To validate the Timeout (TO) field of the Controller Capabilities (CAP) register of an NVMe Controller.

References:

Old Ref: [1] NVMe Specification 3.1.1
NVM Express Base Specification 2.0a : 3.1.3.1

Resource Requirements:

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

Last Modification: June 24, 2015

Discussion: This is the worst case time that host software shall wait for CSTS.RDY to transition from:

a) '0h' to '1h' after CC.EN transitions from '0h' to '1h'; or
b) '1h' to '0h' after CC.EN transitions from '1h' to '0h'

This worst case time may be experienced after events such as an abrupt shutdown or activation of a new firmware image; typical times are expected to be much shorter. This field is in 500 millisecond units.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to read the CAP.TO register field (bits 31:24) of the NVMe Controller.
2. Configure the NVMe Host to clear the CC.EN register field of the NVMe Controller to '0h' to initiate a controller reset.
3. Configure the NVMe Host to query the CSTS.RDY register field every 100 milliseconds until it transitions from '1h' to '0h'.
4. Configure the NVMe Host to set the CC.EN register field of the NVMe Controller to '1h' to re–enable the controller (be sure to set required registers field prior to re-enabling the controller).
5. Configure the NVMe Host to query the CSTS.RDY register field every 100 milliseconds until it transitions from '0h' to '1h'.

Observable Results:

1. Determine the elapsed time it takes for the CSTS.RDY register field value to transition after toggling the value of the CC.EN field. Verify that this value is less than or equal to the timeout value calculated from the CAP.TO register field read from the NVMe Controller.

Possible Problems: None known.

© 2026 UNH–IOL

Test 4.6 – Offset 00h: CAP – Arbitration Mechanism Supported (AMS) (OF, M)

Purpose: To validate the Arbitration Mechanism Supported (AMS) field of the Controller Capabilities (CAP) register of an NVMe Controller.

References:

Old Ref: NVMe Specification 3.1.1
NVM Express Base Specification 2.0a : 3.1.3.1

Resource Requirements:

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

Last Modification: June 24, 2015

Discussion: This field is bit significant (bit definitions can be found in Table 8) and indicates the optional arbitration mechanisms supported by the controller. If a bit is set to '1', then the corresponding arbitration mechanism is supported by the controller. Refer to section 4.11 of the NVMe Specification for arbitration details.

Table 8 – CAP.AMS Bit Definitions

Bit	Definition
17	Weighted Round Robin with Urgent Priority Class
18	Vendor Specific

The round robin arbitration mechanism is not listed since all controller shall support this arbitration mechanism.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to read the CAP.AMS register field (bits 18:17) of the NVMe Controller.
2. For each arbitration mechanism supported by the NVMe Controller as indicated by the CAP.AMS register field, configure the NVMe Host to set the CC.AMS register field to the corresponding value.

Observable Results:

1. For each supported arbitration mechanism, verify that the NVMe Host is able to successfully set the CC.AMS register field to the corresponding value.

Possible Problems: The weighted Round Robin with Urgent and Vendor Specific arbitration mechanisms are optional to support. Therefore, it is valid for both bits of the CAP.AMS field to be cleared to '0'.

© 2026 UNH–IOL

Test 4.7 – Offset 00h: CAP – Contiguous Queues Required (CQR) (OF, M)

Purpose: To validate the Contiguous Queues Required (CQR) field of the Controller Capabilities (CAP) register of an NVMe Controller.

References:

Old Ref: NVMe Specification 3.1.1
NVM Express Base Specification 2.0a : 3.1.3.1

Resource Requirements:

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

Last Modification: March 2, 2016

Discussion: This field is set to '1' if the controller requires that I/O Submission Queues and I/O Completion Queues are required to be physically contiguous. This field is cleared to '0' if the controller supports I/O Submission Queues and I/O Completion Queues that are not physically contiguous. If this field is set to '1', then the Physically Contiguous bit (CDW11.PC) in the Create I/O Submission Queue and Create I/O Completion Queue commands shall be set to '1'.

Since there is no means to validate the value stored in this register field, this test is designed purely to determine the value the NVMe Controller returns when the register is read, and is therefore considered an informative test.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to read the CAP.CQR register field (bit 16) of the NVMe Controller.
2. Report the value of the CAP.CQR register field.

Observable Results: None.

Possible Problems: There are no required values for the CAP.CQR register field at any given time and so this test is run as informative with no pass/fail criteria.

© 2026 UNH–IOL

Test 4.8 – Offset 00h: CAP – Maximum Queue Entries Supported (MQES) (OF, M)

Purpose: To validate the Maximum Queue Entries Supported (MQES) field of the Controller Capabilities (CAP) register of an NVMe Controller.

References:

Old Ref: NVMe Specification 3.1.1
NVM Express Base Specification 2.0a : 3.1.3.1

Resource Requirements:

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

Last Modification: June 24, 2015

Discussion: This field indicates the maximum individual queue size that the controller supports. This value applies to each of the I/O Submission Queues and I/O Completion Queues that host software may create. This is a 0's based value. The minimum value is 1h, indicating two entries.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to read the CAP.MQES register field (bits 15:00) of the NVMe Controller.

Observable Results:

1. Verify that the value of the CAP.MQES register field is 1h or greater indicating 2 or more queue entries.

Possible Problems: None known.

© 2026 UNH–IOL

Test 4.9 – Offset 0Ch–10h: INTMS – Interrupt Mask Set and INTMC – Interrupt Mask Clear (OF, M)

Purpose: To validate the Interrupt Mask Set (INTMS) and Interrupt Mask Clear (INTMC) register of an NVMe Controller.

References:

Old Ref: NVMe Specification 3.1.3, 3.1.4
NVM Express Base Specification 2.0a : 3.1.3

Resource Requirements:

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

Last Modification: June 24, 2015

Discussion: The INTMS register is used to mask interrupts when using pin–based interrupts, single message MSI, or multiple message MSI. When using MSI–X, the interrupt mask table defined as part of MSI–X should be used to mask interrupts. Host software shall not access this register when configured for MSI–X; any accesses when configured for MSI–X is undefined.

The INTMC register is used to unmask interrupts when using pin–based interrupts, single message MSI, or multiple message MSI. When using MSI–X, then interrupt mask table defined as part of MSI–X should be used to unmask interrupts. Host software shall not access this register when configure for MSI–X; any accesses when configured for MSI–X is undefined.

The INTMS and INTMC registers both have a single field each: the Interrupt Vector Mask Set (IVMS) and Interrupt Vector Mask Clear (IVMC) fields respectively. Both fields are bit significant. If a '1' is written to a bit in the IVMS field, then the corresponding interrupt vector is masked from generating an interrupt or reporting a pending interrupt in the MSI Capability Structure. If a '1' is written to a bit in the IVMC field, then the corresponding interrupt vector is unmasked. Writing a '0' to a bit has no effect. When read, these fields return the current interrupt mask value within the controller (not the value of their register). If a bit has a value of '1', then the corresponding interrupt vector is masked. If a bit has a value of '0', then the corresponding interrupt vector is not masked.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to read the INTMS and INTMC registers of the NVMe Controller.
2. Configure the NVMe Host to write all zeros to both the INTMS and INTMC registers of the NVMe Controller.
3. Configure the NVMe Host to read the INTMS and INTMC registers of the NVMe Controller once again.

Observable Results:

1. Verify that the values returned by the registers do not change on subsequent reads after writing zeroes to their bits.

Possible Problems: None known

© 2026 UNH–IOL

Test 4.10 – Offset 14h: CC – I/O Completions Queue Entry Size (IOCQES) (OF, M)

Purpose: To validate the I/O Completion Queue Entry Size (IOCQES) field of the Controller Configuration (CC) register of an NVMe Controller.

References:

Old Ref: NVMe Specification 3.1.5.
NVM Express Base Specification 2.0a : 3.1.3.5

Resource Requirements:

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

Last Modification: April 2, 2019

Discussion: This field defines the I/O Completion Queue entry size that is used for the selected I/O Command Set. The required and maximum values for this field are specified in the Identify Controller data structure for each I/O Command Set. The value is in bytes and is specified as a power of two (2^*n*) (i.e. 2^n).

Test Setup: See Appendix A.

Test Procedure:

1. Check Identify Controller Data Structure for required minimum and maximum values of IOCQES.
2. Configure the NVMe Host to read the CC.IOCQES register field (bits 23:20) of the NVMe Controller.

Observable Results:

1. Verify that the maximum value for IOCQES advertised in the Identify Controller Data Structure is greater than or equal to the required minimum value
2. Verify that the value of the CC.IOCQES register field falls within the range specified for IOCQES as determined by the maximum and minimum values advertised in the Identify Controller Data Structure.

Possible Problems: None known.

© 2026 UNH–IOL

Test 4.11 – Offset 14h: CC – I/O Submission Queue Entry Size (IOSQES) (OF, M)

Purpose: To validate the I/O Submission Queue Entry Size (IOCQES) field of the Controller Configuration (CC) register of an NVMe Controller.

References:

Old Ref: NVMe Specification 3.1.5.
NVM Express Base Specification 2.0a : 3.1.3.5

Resource Requirements:

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

Last Modification: April 2, 2019

Discussion: This field defines the I/O Submission Queue entry size that is used for the selected I/O Command Set.

The required and maximum values for this field are specified in the Identify Controller data structure for each I/O Command Set. The value is in bytes and is specified as a power of two (2^*n*) (i.e. 2^n).

Test Setup: See Appendix A.

Test Procedure:

1. Check Identify Controller Data Structure for required minimum and maximum values of IOSQES.
2. Configure the NVMe Host to read the CC.IOSQES register field (bits 23:20) of the NVMe Controller.

Observable Results:

1. Verify that the maximum value for IOSQES advertised in the Identify Controller Data Structure is greater than or equal to the required minimum value
2. Verify that the value of the CC.IOSQES register field falls within the range specified for IOSQES as determined by the maximum and minimum values advertised in the Identify Controller Data Structure.

Possible Problems: None known.

© 2026 UNH–IOL

Test 4.12 – Offset 14h: CC – Shutdown Notification (SHN) (OF, M)

Purpose: To validate the Shutdown Notification (SHN) field of the Controller Configuration (CC) register of an NVMe Controller.

References:

Old Ref: NVMe Specification 3.1.5.
NVM Express Base Specification 2.0a : 3.1.3.5

Resource Requirements:

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

Last Modification: November 26, 2018

Discussion: This field is used to initiate shutdown processing when a shutdown is occurring, (i.e., a power down condition is expected). For a normal shutdown notification, it is expected that the controller is given time to process the shutdown notification. For an abrupt shutdown notification, the host may not wait for shutdown processing to complete before power is lost. The shutdown notification values are defined in Table 9.

Table 9 – CC.SHN Field Values

Value	Definition
00b	No notification; no effect.
01b	Normal shutdown notification.
10b	Abrupt shutdown notification.
11b	Reserved.

This field should be written by host software prior to any power down condition and prior to any change of the PCI power management state. It is recommended that this field also be written prior to a warm reboot. To determine when shutdown processing is complete, refer to CSTS.SHST.

It is recommended that the host wait a minimum of the RTD3 Entry Latency reported in the Identify Controller data structure for the shutdown operations to complete; if the value reported in RTD3 Entry Latency is 0h, then the host should wait for a minimum of one second. It is not recommended to disable the controller via the CC.EN field. This causes a Controller Reset which may impact the time required to complete shutdown processes.

It is safe to power off the controller when CSTS.SHST indicates shutdown processing is complete. It remains safe to power off the controller until CC.EN transitions from ‘0’ to ‘1’. To start executing commands on the controller after a shutdown operation, a Controller Reset is required. This initializat5ion sequence should then be executed.

Test Setup: See Appendix A.

Test Procedure:

1. Stop submitting any new I/O commands to the DUT and allow any outstanding commands to complete.
2. Perform a Delete I/O Submission Queue command to all existing I/O Submission queues.
3. Perform a Delete I/O Completion Queue command to all existing I/O Completion queues.
4. Configure the NVMe Host to read the CC.SHN register field (bits 15:14) of the NVMe Controller.
5. Configure the NVMe Host to write a value of 01b to the CC.SHN register field in order to initiate shutdown processing with a normal shutdown notification.
6. After the CSTS.SHST register field value transitions back to a value of 10b (shutdown processing complete), configure the NVMe Host to perform a full Controller Reset.
7. Configure the NVMe Host to issue an Identify command to the NVMe Controller specifying CNS value 01h.
8. Configure the NVMe Host to write a value of 10b to the CC.SHN register field in order to initiate shutdown processing with an abrupt shutdown notification.

© 2026 UNH–IOL

9. After the CSTS.SHST register field value transitions back to a value of 10b (shutdown processing complete), configure the NVMe Host to perform a full Controller Reset.
10. Configure the NVMe Host to issue an Identify command to the NVMe Controller specifying CNS value 01h.

Observable Results:

1. Verify that the value of the CC.SHN register field is set to its default value of 00b (no notification) after a Controller Level Reset.
2. After the NVMe Host sets the value of the CC.SHN register field, verify that the controller sets the CSTS.SHST register field to its appropriate value as indicated in Table 12.
3. Verify that, after the NVMe Host performs a Controller Reset while CSTS.SHST indicates normal operation, the NVMe Host is able to issue commands to the NVMe Controller.

Possible Problems: None known.

© 2026 UNH–IOL

Test 4.13 – Offset 14h: CC – Arbitration Mechanism Selected (AMS) (OF, M)

Purpose: To validate the Arbitration Mechanism Selected (AMS) field of the Controller Configuration (CC) register of an NVMe Controller.

References:

Old Ref: [1] NVMe Specification 3.1.5.
NVM Express Base Specification 2.0a : 3.1.3.5

Resource Requirements:

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

Last Modification: February 1, 2018

Discussion: This field selects the arbitration mechanism to be used. This value shall only be changed when EN is cleared to '0'. Host software shall only set this field to supported arbitration mechanisms indicated in CAP.AMS. If this field is set to an unsupported value, the behavior is undefined. The arbitration mechanism values are defined in Table 10.

Table 10 – CC.AMS Field Values

Value	Definition
000b	Round Robin
001b	Weighted Round Robin with Urgent
010b – 110b	Reserved
111b	Vendor Specific.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to set CC.EN to 0.
2. Configure the NVMe Host to read the CC.AMS register field of the NVMe Controller.
3. Configure the NVMe Host to read the CAP.AMS register field of the NVMe Controller.
4. For each optional arbitration mechanism reported to be supported by the CAP.AMS register field:
    a. Configure the NVMe Host to set the CC.AMS register field to the appropriate value for that mechanism.
    b. Configure the NVMe Host to read the CC.AMS register field of the NVMe Controller.

Observable Results:

1. Verify that the value of the CC.AMS register field is set to its default value of 000b when CC.EN is set to 0.
2. Verify that the NVMe Controller properly allows the NVMe Host to set supported values in the CC.AMS register field.

Possible Problems: The weighted Round Robin with Urgent and Vendor Specific arbitration mechanisms are optional to support. Therefore, it is valid for both bits of the CAP.AMS field to be cleared to '0'.

© 2026 UNH–IOL

Test 4.14 – Offset 14h: CC – I/O Command Set Selected (CSS) (OF, M)

Purpose: To validate the I/O Command set Selected (CSS) field of the Controller Configuration (CC) register of an NVMe Controller.

References:

Old Ref: [1] NVMe Specification 3.1.5.
NVM Express Base Specification 2.0a : 3.1.3.5

Resource Requirements:

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

Last Modification: March 2, 2016

Discussion: This field specifies the I/O Command Set that is selected for use for the I/O Submission Queues. Host software shall only select a supported I/O Command Set, as indicated in CAP.CSS. This field shall only be changed when the controller is disabled (CC.EN is cleared to '0'). The I/O Command Set selected shall be used for all I/O Submission Queues. The command set values are defined in Table 11.

Table 11 – CC.CSS Field Values

Value	Definition
000b	NVM Command Set
001b – 111b	Reserved

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to read the CC.CSS register field (bits 06:04) of the NVMe Controller.
2. For each I/O Command Set supported by the NVMe Controller as indicated by the CAP.CSS register field:
    a. Configure the NVMe Host to set the CC.CSS register field to the appropriate value for that I/O Command Set.
    b. Configure the NVMe Host to read the CC.CSS register field of the NVMe Controller.

Observable Results:

1. Verify that the value of the CC.CSS register field is set to its default value of 000b after the NVMe Host clears the CC.EN register field to '0' in order to initiate a Controller Reset (the host must set the CC.CSS field to a valid value prior to re-enabling the controller).
2. Verify that the NVMe Controller properly allows the NVMe Host to set supported values in the CC.CSS field.

Possible Problems: None known.

© 2026 UNH–IOL

Test 4.15 – Offset 14h: CC – Enable (EN) (OF, M)

Purpose: To validate the Enable (EN) field of the Controller Configuration (CC) register of an NVMe Controller.

References:

Old Ref: [1] NVMe Specification 3.1.5.
NVM Express Base Specification 2.0a : 3.1.3.5

Resource Requirements:

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

Last Modification: June 21, 2012

Discussion: When set to '1', the controller shall process commands based on Submission Queue Tail doorbell writes. When cleared to '0', the controller shall not process commands nor post completion queue entries to Completion Queues. When this field transitions from '1' to '0', the controller is reset (referred to as a Controller Reset). The reset deletes all I/O Submission Queues and I/O Completion Queues, resets the Admin Submission Queue and Completion Queue, and brings the hardware to an idle state. The reset does not affect PCI Express registers nor the Admin Queue registers (AQA, ASQ, or ACQ). All other controller registers defined in section 3 of the NVMe specification and internal controller state (e.g., Feature values defined in section 5 of the NVMe specification that are not persistent across power states) are reset to their default values. The controller shall ensure that there is no data loss for commands that have had corresponding completion queue entries posted to an I/O Completion Queue prior to the reset operation. Refer to section 7.3 for reset details.

When this field is cleared to '0', the CSTS.RDY bit is cleared to '0' by the controller once the controller is ready to be re–enabled. When this field is set to '1', the controller sets CSTS.RDY to '1' when it is ready to process commands. CSTS.RDY may be set to '1' before namespace(s) are ready to be accessed.

Setting this field from a '0' to a '1' when CSTS.RDY is a '1', or setting this field from a '1' to a '0' when CSTS.RDY is a '0' has undefined results. The Admin Queue registers (AQA, ASQ, and ACQ) shall only be modified when this field is cleared to '0'.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to read the CC.EN register field (bit 00) of the NVMe Controller.

Observable Results:

1. Verify that when Bit 0 of the CC.EN register field is set to '1', the controller processes commands based on Submission Queue Tail doorbell writes.
2. Verify that when Bit 0 of the CC.EN register field is cleared to '0', the controller does not process commands nor post completion queue entries to Completion Queues.

Possible Problems: None known

© 2026 UNH–IOL

Test 4.16 – Offset 1Ch: CSTS – Shutdown Status (SHST) (OF, M)

Purpose: To validate the Shutdown Status (SHST) field of the Controller Status (CSTS) register of an NVMe Controller.

References:

Old Ref: NVMe Specification 3.1.6.
NVM Express Base Specification 2.0a : 3.1.3.6

Resource Requirements:

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

Last Modification: March 2, 2016

Discussion: This field indicates the status of shutdown processing that is initiated by the host setting the CC.SHN register field. The shutdown status values are defined in Table 12.

Table 12 – CSTS.SHST Field Values

Value	Definition
00b	Normal operation (no shutdown has been requested)
00b	Normal operation (no shutdown has been requested)
00b	Normal operation (no shutdown has been requested)

To start executing commands on the controller after a shutdown operation (CSTS.SHST set to 10b), a reset (CC.EN cleared to '0') is required. If host software submits commands to the controller without issuing a reset, the behavior is undefined.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to read the CSTS.SHST register field (bits 03:02) of the NVMe Controller.
2. Configure the NVMe Host to write a value of 01b to the CC.SHN register field in order to initiate shutdown processing with a normal shutdown notification.
3. Configure the NVMe Host to read the CSTS.SHST register field of the NVMe Controller.
4. After the CSTS.SHST register field value transitions back to a value of 10b (shutdown processing complete), configure the NVMe Host to perform a full Controller Reset.
5. Configure the NVMe Host to read the CSTS.SHST register field of the NVMe Controller.
6. Configure the NVMe Host to issue an Identify command to the NVMe Controller specifying CNS value 01h.
7. Configure the NVMe Host to write a value of 10b to the CC.SHN register field in order to initiate shutdown processing with an abrupt shutdown notification.
8. Configure the NVMe Host to read the CSTS.SHST register field of the NVMe Controller.
9. After the CSTS.SHST register field value transitions back to a value of 10b (shutdown processing complete), configure the NVMe Host to perform a Controller Reset.
10. Configure the NVMe Host to read the CSTS.SHST register field of the NVMe Controller.
11. Configure the NVMe Host to issue an Identify command to the NVMe Controller specifying CNS value 01h.

Observable Results:

1. Verify that the value of the CSTS.SHST register field is set to its default value of 00b (normal operation) after a Controller Level Reset.
2. After the NVMe Host sets the value of the CC.SHN register field to either 01b or 10b, verify that the NVMe Controller sets the CSTS.SHST register field to 01b to indicate that shutdown processing is occurring.
3. Record the time it takes for the NVMe Controller to transition the value of the CSTS.SHST register field from 01b to 10b (i.e. the shutdown processing time) for both normal and abrupt shutdown notifications.

© 2026 UNH–IOL

4. Verify that, after the NVMe Host performs a Controller Reset while the CSTS.SHST register field indicates normal operation, the NVMe Host is able to issue commands to the NVMe Controller.

Possible Problems: None known.

© 2026 UNH–IOL

Test 4.17 – Offset 1Ch: CSTS – Controller Fatal Status (CFS) (OF, M)

Purpose: To validate the Controller Fatal Status (CFS) field of the Controller Status (CSTS) register of an NVMe Controller.

References:

Old Ref: NVMe Specification 3.1.6.
NVM Express Base Specification 2.0a : 3.1.3.6

Resource Requirements:

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

Last Modification: March 2, 2016

Discussion: This field is set to '1' when a fatal controller error occurred that could not be communicated in the appropriate Completion Queue. This field is cleared to '0' when a fatal controller error has not occurred. Refer to section 9.5.

The reset value of this field is '1' when a fatal controller error is detected during controller initialization.

Since there is no means for a host to trigger a controller fatal status, there is no means to validate the value stored in this register field, and so this test is designed purely to determine the value the NVMe Controller returns when the register is read, and is therefore considered an informative test.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to read the CSTS.CFS field (bit 01) of the NVMe Controller.
2. Report the value of the CSTS.CFS field.

Observable Results: None.

Possible Problems: There are no explicit conditions for which the NVMe Host can generate a controller fatal status. Therefore this test is run as informative with no pass/fail criteria.

© 2026 UNH–IOL

Test 4.18 – Offset −08h: CAP – Version (VS) (OF, M)

Purpose: To validate the Version (VS) field of the Capabilities (CAP) register of an NVMe Controller.

References:

Old Ref: NVMe Specification 3.1.2
NVM Express Base Specification 2.0a : 3.1

Resource Requirements:

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

Last Modification: November 14, 2017

Discussion: This register indicates the major and minor version of the NVM Express specification that the controller implementation supports. Valid versions of the specification are: 1.0, 1.1, 1.2, 1.3, 1.4 and 2.0.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to read the VS.CAP field of the NVMe Controller.
2. Report the value of the VS.CAP field.
3. 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.
4. Report the value of the VER field in the Identify Controller data structure returned by the DUT.

Observable Results:

1. Verify that the DUT reports a valid value for VS. Valid values are 1.0, 1.1, 1.2, 1.3, 1.4, 2.0
2. Verify that the version reported in the Identify Controller data structure matches the value reported in the VS. CAP field.

© 2026 UNH–IOL

Test 4.19 – Offset 1Ch: Controller Processing Paused (PP) (FY-OF, M)

Purpose: To validate the Processing Paused (PP) field of the Controller Status (CSTS) register of an NVMe Controller.

References:

NVM Express Base Specification 2.0c : 3.1

Resource Requirements:

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

Last Modification: January 18, 2024

Discussion: This register indicates if a controller is able to process commands. If CSTS.PP is set to 1, then the controller will not be able to process commands. If the controller is processing commands, CSTS.PP must be cleared to 0.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to read the CC.EN and CSTS.RDY register fields of the NVMe Controller. If either the CC.EN or CSTS.RDY register fields are not set to 1, then this test is unable to be run and is not applicable.
2. Configure the NVMe Host to read the CSTS.PP register field of the NVMe Controller
3. Configure the NVMe host to send an Identify command with CNS 01h.

Observable Results:

1. Verify that the CSTS.PP register field is set to 0
2. Verify that the Identify command in step 3 of the procedure completes successfully.

© 2026 UNH–IOL

Test 4.20 – Offset 68h: Controller Ready Independent of Media Timeout (CRIMT) (FYI-OF, M)

Purpose: To validate the Controller Ready Independent of Media Timeout (CRIMT) field of the Controller Ready Timeouts (CRTO) register of an NVMe Controller.

References:

NVM Express Base Specification 2.0c : 3.1.3.21

Resource Requirements:

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

Last Modification: January 18, 2024

Discussion: This register indicates when the worst case timeout value for host software should to wait after the controller to transitions CC.EN from 0 to 1 to begin processing all commands that do not access attached namespaces and NVMe Admin Commands that do not require access to media when the controller is in Controller Ready Independent of Media mode (i.e., the CC.CRIME bit is set to '1').

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to read the CAP.CRMS.CRIMS register field of the NVMe Controller. If the CAP.CRMS.CRIMS register is not cleared to zero, this test is not applicable.
2. Configure the NVMe Host to read the CRTO.CRIMT register field of the NVMe Controller.

Observable Results:

1. Verify that the CRTO.CRIMT register field is cleared to zero.

© 2026 UNH–IOL

Test 4.21 – Offset 00h: NVM Subsystem Shutdown Enhancements (FYI-OF, FYI)

Purpose: To validate the requirements of NVM Subsystem Shutdown Enhancements introduced in TP4169.

References:

NVM Express Base Specification 2.1 : 3.6.3

Resource Requirements:

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

Last Modification: January 23, 2025

Discussion: The NVM Subsystem Shutdown Feature defined in NVM Express Base Specification version 2.1 includes some functionality that differs from the functionality of the NVM Subsystem Shutdown Feature defined in revision 2.0 of the NVM Express Base Specification. This test is designed to test for the requirements when a controller sets the NVM Subsystem Shutdown Enhancements Supported (CAP.NSSES) bit to '1'.

Test Setup: See Appendix A.

Case 1: NVM Subsystem Shutdown Enhancements VER > 2.0 (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify Command to retrieve the Identify Controller (CNS 01h) Data Structure. If the controller reports support for NVM Express Base Specification revision 2.0 or earlier, then this test is not applicable.
2. Configure the NVMe Host to read the Controller Capabilities (CAP) register (Offset 0h). If the NSSS bit is cleared to 0b, then this test is not applicable.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the NSSES bit is set to 1b.

Case 2: NVM Subsystem Shutdown Enhancements Supported (FYI)

Test Procedure:

1. Configure the NVMe Host to read the Controller Capabilities (CAP) register (Offset 0h). If the NSSES bit is cleared to 0b, then this test is not applicable.

Observable Results:

1. Verify that the NSSS bit is set to 1b.

Case 3: NVM Subsystem Shutdown Enhancements Not Supported (M)

Test Procedure:

1. Configure the NVMe Host to read the Controller Capabilities (CAP) register (Offset 0h). If the NSSRS and NSSS bits are both set to 1b, then this test is not applicable.

Observable Results:

1. Verify that the NSSES bit is set to 0b.

Case 4: Controller Level Reset - ST=0, SHST=10b (FYI)

Test Procedure:

© 2026 UNH–IOL

1. Configure the NVMe Host to issue an Identify Command to retrieve the Identify Controller (CNS 01h) Data Structure. If the controller reports support for NVM Express Base Specification revision 2.0 or earlier, then this test is not applicable.
2. Configure the NVMe Host to perform a Controller Shutdown by writing 01b to the Shutdown Notification (SHN) field in the Controller Configuration (CC) register (Offset 14h).
3. Configure the NVMe Host to read the Controller Status (CSTS) register (Offset 1Ch). Record the values of Shutdown Status (SHST) and Shutdown Type (ST).
4. Configure the NVMe Host to poll the Controller Status (CSTS) register (Offset 1Ch) until the value of the Shutdown Status (SHST) field transitions to 10b.
5. Configure the NVMe Host to perform a Controller Level Reset.
6. Configure the NVMe Host to poll the Controller Status (CSTS) register (Offset 1Ch) until RDY changes from 0 to 1. Record the values of SHST and RDY with each read.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the ST bit is set to 0b in step 2.
3. Verify that the SHST field is set to 10b after step 4.
4. Verify that the SHST field is cleared to 00b either in a read prior to the read where RDY is set to 1, or in the same read.

Case 5: Controller Level Reset Doesn't Affect Shutdown Status (FYI)

Test Procedure:

1. Configure the NVMe Host to read the Controller Capabilities (CAP) register (Offset 0h). Record the values of NSSS and NSSES. If NSSS is cleared to 0b, then this test is not applicable.
2. Configure the NVMe Host to write the value 4E564D65h to the NVM Subsystem Reset Control (NSSRC) field in the NVM Subsystem Reset (NSSR) register (Offset 20h).
3. Configure the NVMe Host to read the Controller Status (CSTS) register (Offset 1Ch). Record the values of Shutdown Status (SHST) and Shutdown Type (ST).
4. Configure the NVMe Host to perform a Controller Level Reset and wait for it to complete.
5. Configure the NVMe Host to read the Controller Status (CSTS) register (Offset 1Ch). Record the value of Shutdown Status (SHST).

Observable Results:

1. Verify that the ST bit and the SHST field are cleared to 0b in step 3.
2. Verify that the SHST field remains cleared to 00b in step 5.

Case 6: Shutdown Type Remains Set Until NVM Subsystem Reset (FYI)

Test Procedure:

1. Configure the NVMe Host to read the Controller Capabilities (CAP) register (Offset 0h). Record the values of NSSS and NSSES. If NSSS is cleared to 0b, then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to retrieve the Identify Controller (CNS 01h) Data Structure. Record the value of NSSL.
3. Configure the NVMe Host to write the value 4E726D6Ch to the NVM Subsystem Shutdown Control (NSSC) field in the NVM Subsystem Shutdown (NSSD) register (Offset 64h).
4. Configure the NVMe Host to read the Controller Status (CSTS) register (Offset 1Ch). Record the values of Shutdown Status (SHST) and Shutdown Type (ST).
5. Configure the NVMe Host to poll the Controller Status (CSTS) register (Offset 1Ch) until the value of the Shutdown Status (SHST) field transitions to 10b.
6. Configure the NVMe Host to perform a Controller Level Reset and wait for it to complete.
7. Configure the NVMe Host to read the Controller Status (CSTS) register (Offset 1Ch). Record the values of Shutdown Status (SHST) and Shutdown Type (ST).

© 2026 UNH–IOL

8. Configure the NVMe Host to write the value 4E564D65h to the NVM Subsystem Reset Control (NSSRC) field in the NVM Subsystem Reset (NSSR) register (Offset 20h).
9. Configure the NVMe Host to read the Controller Status (CSTS) register (Offset 1Ch). Record the values of Shutdown Status (SHST) and Shutdown Type (ST).

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the ST bit was set to 1b in steps 4 and 5.
3. If NSSL = 0, verify that the Shutdown Status (SHST) field transitioned to 10b in less than 30 seconds during step 5. Otherwise, verify that the Shutdown Status (SHST) field transitioned to 10b in less than 10*NSSL microseconds during step 5.
4. Verify that the SHST field is set to 10b and the ST bit is set to 1b in step 7.
5. Verify that the ST bit and the SHST field are cleared to 0b in step 9.

Possible Problems: Regarding Observable Result 3, If NSSL= 0, NVM Express Base Specification revision 2.1 recommends the host to wait for a minimum of 30 seconds. If the SHST does not change within 30 seconds, this test will fail to avoid an indefinite shutdown timeout.

Case 7: CLR Not Initiated by NVM Subsystem Reset Does Not Abort NVM Subsystem Shutdown (FYI)

Test Procedure:

1. Configure the NVMe Host to read the Controller Capabilities (CAP) register (Offset 0h). Record the values of NSSS and NSSES. If NSSS or NSSES are cleared to 0b, then this test is not applicable.
2. Configure the NVMe Host to write the value 4E726D6Ch to the NVM Subsystem Shutdown Control (NSSC) field in the NVM Subsystem Shutdown (NSSD) register (Offset 64h).
3. Configure the NVMe Host to read the Controller Status (CSTS) register (Offset 1Ch). Record the values of Shutdown Status (SHST) and Shutdown Type (ST).
4. Configure the NVMe Host to perform a Controller Level Reset.
5. Configure the NVMe Host to read the Controller Status (CSTS) register (Offset 1Ch). Record the values of Shutdown Status (SHST) and Shutdown Type (ST).
6. Configure the NVMe Host to write the value 4E564D65h to the NVM Subsystem Reset Control (NSSRC) field in the NVM Subsystem Reset (NSSR) register (Offset 20h).
7. Configure the NVMe Host to read the Controller Status (CSTS) register (Offset 1Ch). Record the values of Shutdown Status (SHST) and Shutdown Type (ST).

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the SHST field is set to 01b or 10b and the ST bit is set to 1b in step 3.
3. Verify that the SHST field and ST bit are not cleared to 0b in step 5.
4. Verify that the SHST field and ST bit are cleared to 0b in step 7.

© 2026 UNH–IOL

Group 5: System Memory Structure

Overview:
This section describes a method for performing conformance verification for NVMe products implementing the NVMe System Memory Structure.

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).

© 2026 UNH–IOL

Test 5.1 – Page Base Address and Offset (PBAO) (OF, M)

Purpose: To validate the Page Base Address and Offset field of PRP entries issued to an NVMe Controller.

References:

Old Ref: NVMe Specification 4.3
NVM Express Base Specification 2.0a : 4.1.1
NVM Express NVM Command Set Specification 1.0a : 3.2

Resource Requirements:

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

Last Modification: February 1, 2018

Discussion: A physical region page (PRP) entry is a point to a physical memory page. PRPs are used as a scatter/gather mechanism for data transfers between the controller and memory. To enable efficient out of order data transfers between the controller and the host, PRP entries are a fixed size.

The size of the physical memory page is configured by host software in CC.MPS. shows the layout of a PRP entry that consists of a Page Base Address and an Offset. The size of the Offset field is determined by the physical memory page size configured in CC.MPS.

The Page Base Address and Offset (PBAO) field indicates the 64–bit physical memory page address. The lower bits (*n*:2) of this field indicate the offset within the memory page. If the memory page size is 4KB, then bits 11:02 form the Offset; if the memory page size is 8KB, then bits 12:02 form the Offset, etc. (i.e. n = CC.MPS + 11). If this entry is not the first PRP entry in the command or a PRP List pointer in a command, then the Offset portion of this field shall be cleared to 0h.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to issue an Identify command with properly formatted PRP entries to the NVMe Controller.

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.

Possible Problems: None known.

© 2026 UNH–IOL

Test 5.2 – Completion Queue Entry (M)

Purpose: To verify that an NVMe Controller can properly post a completion queue entry to the proper Completion Queue.

References:

Old Ref: NVMe Specification 4.6
NVM Express Base Specification 2.0a : 3

Resource Requirements:

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

Last Modification: November 30, 2020

Discussion: An entry in the Completion Queue is 16 bytes in size. The NVMe specification describes the layout of the Completion Queue Entry data structure. The contents of Dword 0 are command specific. If a command uses Dword 0, then the definition of this Dword is contained within the associated command definition. If a command does not use Dword 0, then the field is reserved. Dword 1 is reserved. Dword 2 is defined in Figure 26 and Dword 3 is defined in Figure 27 of the NVMe specification. Any additional I/O Command Set defined in the future may use an alternate Completion Queue entry size or format.

The Status Field (SF) of Dword 3 is tested in Test 5.3 – Status Field Definition.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to setup and create an I/O Submission Queue paired with a I/O Completion Queue.
2. Configure the NVMe Host to issue a Write command to the NVMe Controller through the IOSQ created in the previous step.
3. After completion of the Write Command, delete both the I/O Submission Queue and I/O Completion Queue.
4. Configure the NVMe Host to setup and create a new I/O Submission Queue paired with a new I/O Completion Queue.
5. Configure the NVMe Host to issue a Write command to the NVMe Controller through the new IOSQ.

Observable Results:

1. Verify that after the completion of each command, the controller posts a completion queue entry to the I/O Completion Queue indicating the status for the command.
2. Verify that the Submission Queue Identifier (SQID) field (bits 31:16) of Dword 2 of each completion queue entry is the SQID of the Submission Queue used to issue the command.
3. Verify that the Phase Tag (P) field (bit 16) of Dword 3 is set to 1 for the first round of completion queue entries arriving at the IOCQ and cleared to 0 for the second round of completion queue entries after the controller had wrapped around the top of the Completion Queue.
4. Verify that the Command Identifier (CID) field (bits 15:00) of Dword 3 for each completion queue entry matches the command ID assigned to the command by the NVMe Host when it was issued to the Submission Queue.

Possible Problems: None known.

© 2026 UNH–IOL

Test 5.3 – Status Field Definition (OF, M)

Purpose: To verify that an NVMe Controller can properly return the status in the completion queue entry for a command.

References:

Old Ref: NVMe Specification 4.6.1
NVM Express Base Specification 2.0a : 3.3.3.2

Resource Requirements:

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

Last Modification: June 24, 2015

Discussion: The Status Field (SF) defines the status for the command indicated in the completion queue entry. If the SCT and SC fields are cleared to 0h, then this indicates a successful command completion, with no fatal or non–fatal error conditions.

The Status Field is bits 31:17 of Dword 3 of all completion queue entries.

The SCT and SC fields are further tested in subsequent tests.

Test Setup: See Appendix A.

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to the NVMe Controller.
2. After the NVMe Controller has processed the command, configure the NVMe Host to read the completion queue entry for the Identify command from the Admin Completion Queue.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
2. Verify that the Status Code Type (SCT) field of the Status Field is 0h to indicate a Generic Command Status. Verify that the Status Code (SC) field of the Status Field is 0h to indicate successful completion of the command.

Possible Problems: None known.

© 2026 UNH–IOL

Test 5.4 – Generic Command Status Definition (OF-FYI, M)

Purpose: To verify that an NVMe Controller can properly return Status Code values for the Generic Command Status type.

References:

Old Ref: NVMe Specification 4.6.1.2.1
NVM Express Base Specification 2.0a : 3.3.3.2.1.1

Resource Requirements:

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

Last Modification: April 2, 2019

Discussion: Completion queue entries with a Status Code type of Generic Command Status (0h) indicate a status value associated with the command that is generic across all command types. The Status Code values for the Generic Command Status type are defined in Table 14 and Table 13.

Table 13 – Generic Command Status Values, NVM Command Set

Value	Description
80h	**LBA Out of Range:** The command references an LBA that exceeds the size of the namespace.
81h	**Capacity Exceeded:** Execution of the command has caused the capacity of the namespace to be exceeded. This error occurs when the Namespace Utilization exceeds the Namespace Capacity.
82h	**Namespace Not Ready:** The namespace is not ready to be accessed. The Do Not Retry bit indicates whether re–issuing the command at a later time may succeed.
83h	**Reservation Conflict:** The command was aborted due to a conflict with a reservation held on the accessed namespace.
84h	**Format In Progress.** The namespace is currently being formatted. The Do Not Retry bit shall be cleared to '0' to indicate that the command may succeed if it is resubmitted.
85h	**Invalid Value Size**: See the applicable I/O Command Set for the description
86h	**Invalid Key Size**: See the applicable I/O Command Set for the description
87h	**KV Key Does Not Exist**: See the applicable I/O Command Set for the description
88h	**Unrecoverable Error**: See the applicable I/O Command Set for the description
89h	**Key Exists**: See the applicable I/O Command Set for the description
90h to BFh	Reserved
C0h – FFh	Vendor Specific

Test Setup: See Appendix A.

Case 1: Read Success (OF-FYI, M)

Test Procedure:

1. Send a Read Command to the DUT.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status Success (00h) for the command.

© 2026 UNH–IOL

Case 2: Invalid Opcode Error (OF-FYI, M)

Test Procedure:

1. Send a command with opcode 0x7E to the DUT.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status 'Invalid Opcode' (01h) for the command.

Case 3: Invalid Field Error (OF-FYI, M)

Test Procedure:

1. Send an Identify command with a CNS value of 0xFF to the DUT.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status 'Invalid Field in Command' (02h) for the command.

Case 4: Invalid Namespace or Format Error (OF-FYI, M)

Test Procedure:

1. Send an IO command with an NSID of 0xFFFFFFFE to the DUT.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status 'Invalid Namespace or Format' (0Bh) for the command.

Case 5: Command Sequence Error (OF-FYI, M)

Test Procedure:

1. This test is only applicable if the DUT supports NVMe v1.4 or higher.
2. Allow the DUT to initialize and setup completion and submission queues.
3. Send a Set Feature command to FID 0x07 to the DUT.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status 'Command Sequence Error' (0Ch) for the command.

Case 6: LBA Out of Range (OF-FYI, M)

Test Procedure:

1. Send an IO command with the SLBA field set to all 1's.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status 'LBA Out of Range' (80h) for the command.

Possible Problems: If the NVMe specification does not explicitly state the exact conditions for when an NVMe Controller should use some of the defined Status Codes. Such status codes cannot be tested.

© 2026 UNH–IOL

Table 14 – Generic Command Status Values

Value	Description
00h	**Successful Completion:** The command completed successfully.
01h	**Invalid Command Opcode:** The associated command opcode field is not valid.
02h	**Invalid Field in Command:** An invalid field in the command parameters.
03h	**Command ID Conflict:** The command identifier is already in use. Note: It is implementation specific how many commands are searched for a conflict.
04h	**Data Transfer Error:** Transferring the data or metadata associated with a command had an error.
05h	**Commands Aborted due to Power Loss Notification:** Indicates that the command was aborted due to a power loss notification.
06h	**Internal Error:** The command was not completed successfully due to an internal error. Details on the internal device error are returned as an asynchronous event. Refer to section 5.2
07h	**Command Abort Requested:** The command was aborted due to a Command Abort command being received that specified the Submission Queue Identifier and Command Identifier of this command.
08h	**Command Aborted due to SQ Deletion:** The command was aborted due to a Delete I/O Submission Queue request received for the Submission Queue to which the command was submitted.
09h	**Command Aborted due to Failed Fused Command:** The command was aborted due to the other command in a fused operation failing.
0Ah	**Command Aborted due to Missing Fused Command:** The command was aborted due to the companion fused command not being found as the subsequent Submission Queue entry.
0Bh	**Invalid Namespace or Format:** The namespace or the format of that namespace is invalid.
0Ch	**Command Sequence Error:** The command was aborted due to a protocol violation in a multi–command sequence (e.g. a violation of the Security Send and Security Receive sequencing rules in the TCG Storage Synchronous Interface Communications protocol).
0Dh	**Invalid SGL Last Segment Descriptor:** The command includes an invalid SGL Last Segment. This may occur when the SGL segment pointed to by an SGL Last Se4gment descriptor contains an SGL Segment descriptor or an SGL Last Segment descriptor. This may occur when an SGL Last Segment descriptor contains an invalid length (i.e., a length of zero or one that is not a multiple of 16).
0Eh	**Invalid Number of SGL Descriptors:** The number of SGL descriptors or SGL Last Segment descriptors in a SGL segment is greater than one.
0Fh	**Data SGL Length Invalid:** The length of a Data SGL is too short or too long.
10h	**Metadata SGL Length Invalid:** The length of a Metadata SGL is too short or too long.
11h	**SGL Descriptor Type Invalid:** The type of an SGL Descriptor is a type that is not supported by the controller.
12h	**Invalid Use of Controller Memory Buffer:** The attempted use of the Controller Memory Buffer is not supported by the controlle
13h	**PRP Offset Invalid:** The Offset field for a PRP entry is invalid. This may occur when there is a PRP entry with a non–zero offset after the first entry.
14h	**Atomic Write Unit Exceeded:** The length specified exceeds the atomic write unit size.
15h – 7Fh	Reserved
80h – BFh	I/O Command Set Specific
C0h – FFh	Vendor Specific

© 2026 UNH–IOL

Test 5.5 – Command Specific Errors Definition (OF-FYI, M)

Purpose: To verify that an NVMe Controller can properly return the Status Code values for the Command Specific Status type.

References:

Old Ref: 4.6.1.2.2 Figure 33, 5.21.1.9 (Case 7)
NVM Express Base Specification 2.0a : 3.3.3.2.1.2, 5.27.1.7

Resource Requirements:

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

Last Modification: April 2, 2019

Discussion: Completion queue entries with a Status Code Type of Command Specific Errors (01h) indicate an error that is specific to a particular command opcode. These status values may indicate additional processing is required. The Status Code values for the Command Specific Status Type are shown below.

For each of the Status Code values defined in Table 15 and Table 17 configure the NVMe Host to issue a command to the NVMe Controller which will cause the controller to return the Status Code in the completion queue entry for the command.

Table 15 – Command Specific Status Values, NVM Command Set

Value	Description	Commands Affected	Test Coverage
80h	Conflicting Attributes	Dataset Management, Read, Write	
81h	Invalid Protection Information	Compare, Read, Write, Write Zeroes	
82h	Attempted Write to Read Only Range	Dataset Management, Write, Write Uncorrectable, Write Zeroes	
83h	Command Size Limit Exceeded	Get/Set Feature	
84h – B7h	Reserved	N/A	

© 2026 UNH–IOL

Table 16 – Command Specific Status Values

Value	Description	Commands Affected	Test Coverage
00h	Completion Queue Invalid	Create I/O Submission Queue	Test 1.4 Case 7
01h	Invalid Queue Identifier	Create I/O Submission Queue, Create I/O Completion Queue, Delete I/O Submission Queue, Delete I/O Completion Queue	Test 1.4 Case 2 and 7
02h	Invalid Queue Size	Create I/O Submission Queue, Completion Queue Create I/O	Test 1.4 Case 4 and 5
03h	Abort Command Limit Exceeded	Abort	Test 5.5 Case 1
04h	Reserved	N/A	Test 5.5 Case 1
05h	Asynchronous Event Request Limit Exceeded	Asynchronous Event Request	N/A
06h	Invalid Firmware Slot	Firmware Commit	Test 5.5 Case 2
07h	Invalid Firmware Image	Firmware Commit	Test 5.5 Case 3
08h	Invalid Interrupt Vector	Create I/O Completion Queue	Not Implemented
09h	Invalid Log Page	Get Log Page	Test 1.4 Case 8
0Ah	Invalid Format	Format NVM	Test 1.3 Case 2 and 3
0Bh	Firmware Activation Requires Conventional Reset	Firmware Commit	Test 1.6 Case 5
0Ch	Invalid Queue Deletion	Delete I/O Completion Queue	Not Implemented
0Dh	Feature Identifier Not Saveable	Set Features	Test 1.4 Case 3
0Eh	Feature Not Changeable	Set Features	Test 5.5 Case 4
0Fh	Feature Not Namespace Specific	Set Features	Test 5.5 Case 5
10h	Firmware Activation Required NVM Subsystem Reset	Firmware Commit	Test 5.5 Case 6 and 7
11h	Firmware Activation Requires Reset	Firmware Commit	Not Implemented
12h	Firmware Activation Requires Maximum Time Violation	Firmware Commit	Not Implemented
13h	Firmware Activation Prohibited	Firmware Commit	Not Implemented
14h	Overlapping Range	Firmware Commit, Firmware Image Download, Set Features	Not Implemented
15h	Namespace Insufficient Capacity	Namespace Management	Test 5.5 Case 8
16h	Namespace Identifier Unavailable	Namespace Management	Test 9.2 Case 3
17h	Reserved	N/A	Test 9.2 Case 1
18h	Namespace Already Attached	Namespace Attachment	N/A
19h	Namespace Is Private	Namespace Attachment	Test 9.3 Case 1
1Ah	Namespace Not Attached	Namespace Attachment	Not Implemented
1Bh	Thin Provisioning Not Supported	Namespace Management	Test 9.3 Case 2
1Ch	Controller List Invalid	Namespace Attachment	Not Implemented
1Dh	Device Self-test In Progress	Device Self-test	Not Implemented
1Eh	Boot Partition Write Prohibited	Firmware Commit	Not Implemented
[illegible]	Invalid Controller Identifier	Virtualization Management	Not Implemented
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	Memory Region is Enabled	Namespace Management	[illegible]

© 2026 UNH–IOL

Test Setup: See Appendix A.

Case 1: Abort Command Limit Exceeded (OF-FYI, M)

Test Procedure:

1. Determine the Abort Command Limit Exceeded value, *n*, by examining the Identify Controller Data Structure of the DUT.
2. Configure the NVMe Host to issue *n+1* Abort commands to the NVMe Controller.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status for the command.
2. Verify that the SCT field of the Status Field is set to Abort Command Limit Exceeded, 03h for one command only.

Possible Problems: Section 5.1 of the NVM Express Base specification 2.0 states that the controller may complete any excess Abort commands with Abort Limit Exceeded status.

Case 2: Asynchronous Event Request Limit Exceeded (OF-FYI, M)

Test Procedure:

1. Determine the Asynchronous Event Request Limit value, *n*, by examining the Identify Controller Data Structure of the DUT.
2. Configure the NVMe Host to issue *n+1* Asynchronous Event Request commands to the NVMe Controller.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status for the command.
2. Verify that the SCT field of the Status Field is set to Asynchronous Event Request Limit Exceeded, 05h for one command only.

Case 3: Invalid Firmware Slot (OF-FYI, M)

Test Procedure:

1. Check that OACS Bit 2 = 1. If not, then test is not applicable.
2. Determine the number of firmware slots the DUT supports by examining the Firmware Update field in the Identifier Controller Data Structure.
3. Configure the NVMe Host to issue a Firmware Commit with a Slot ID of one greater than the number of slots supported by the DUT.
    a. If the DUT supports the maximum number of slots, determine if Slot 1 is Read-only. If Slot 1 is read-only, issue a Firmware Commit to Slot 1. If Slot 1 is not read-only, then this test is not applicable.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status for the command.
2. Verify that the SCT field of the Status Field is set to Invalid Firmware Slot, 06h.

Case 4: Feature Identifier Not Saveable (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue a Get Feature command to the NVMe Controller for each supported Feature ID.
2. For each Feature indicated as Not Saveable, issue a Set Feature Command to save that feature.

Observable Results:

© 2026 UNH–IOL

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status for the command.
2. Verify that the SCT field of the Status Field is set to Feature Identifier Not Saveable, 0Dh.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

Case 5: Feature Not Changeable (OF-FYI, M)

Test Procedure:

5. Configure the NVMe Host to issue a Get Feature command to the NVMe Controller for each supported Feature ID.
6. For each Feature indicated as Not Changeable, issue a Set Feature Command to change the value of that feature to a value different than the current value.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status for the command.
2. Verify if the DUT indicates that certain supported features are unchangeable, that the SCT field of the Status Field is set to Feature Not Changeable, 0Eh when the DUT responds to Set Features commands sent for that Feature. If no features are indicated as Not Changeable, then this test case is Not Applicable.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

Case 6: Feature Not Namespace Specific IV=1 (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue a Set Feature – Interrupt Vector Configuration command to the NVMe Controller with IV=1 and CD=0.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status for the command.
2. Verify that the SCT field of the Status Field is set to Feature Not Namespace Specific, 0Fh.

Case 7: Overlapping Range (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue a Set Feature command to the NVMe Controller with an overlapping LBA range. See Possible Problems description for Case 7 below.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status for the command.
2. Verify that the SCT field of the Status Field is set to Overlapping Range, 14h.

Possible Problems: The NVMe specification does not explicitly state the exact conditions for when an NVMe Controller should use some of the defined Status Codes. Such status codes cannot be tested. Additionally, some of the Status Codes can only be used for optional commands which the NVMe controller may or may not support.

For Case 5, some instances had been observed where if a Feature was not changeable, and the Host tried to change the feature value, but the value offered was the same as the current value, the controller accepted the Set Feature command. If this occurs, the result may be dependent on the order of field checking that the controller uses, which is beyond the scope of the NVMe specification. The test case has been updated to avoid this condition.

© 2026 UNH–IOL

For Case 7, requirements on reporting Overlapping range errors were relaxed in NVMe v1.4 and returned to a Shall in 2.0 relative to earlier versions of the specification. While controllers should report overlapping range errors, it is no longer a ‘shall’ requirements to do so. Therefore this test should be considered FYI, and informative only for 1.4 devices.

© 2026 UNH–IOL

Test 5.6 – Media and Data Integrity Errors Definition (OF-FYI, M)

Purpose: To verify that an NVMe Controller can properly return the status for the command in the completion queue entry.

References:

Old Ref: NVMe Specification 4.6.1.2.3
NVM Express Base Specification 2.0a : 3.3.3.2.1.3

Resource Requirements:

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

Last Modification: April 2, 2019

Discussion: Completion queue entries with a Status Code Type of Media and Data Integrity Errors (02h) indicate a media specific error that occurred in the NVM or data integrity type errors. The Status Code values for the Media and Data Integrity Errors Status Type are defined in Table 17 and Table 18.

Table 17 – Media and Data Integrity Error Values

Value	Definition
00h – 7Fh	Reserved
80h – BFh	I/O Command Set Specific
C0h – FFh	Vendor Specific

Table 18 – Media and Data Integrity Error Values, NVM Command Set

Value	Definition
80h	**Write Fault:** The write data could not be committed to the media.
81h	**Unrecovered Read Error:** The read data could not be recovered from the media.
82h	**End–to–end Guard Check Error:** The command was aborted due to an end–to–end guard check failure.
83h	**End–to–end Application Tag Check Error:** The command was aborted due to an end–to–end application tag check failure
84h	**End–to–end Reference Tag Check Error:** The command was aborted due to an end–to–end reference tag check failure.
85h	**Compare Failure:** The command failed due to a miscompare during a Compare command.
86h	**Access Denied:** Access to the namespace and/or LBA range is denied due to lack of access rights. Refer to TCG SIIS.
87h	**Deallocated or Unwritten Logical Block:** The command failed due to an attempt to read from an LBA range containing a deallocated or unwritten logical block.
88h	**End-to-End Storage Tag Check Error:** The command was aborted due to an end-to-end storage tag check failure.
89h – BFh	Reserved

Test Setup: See Appendix A.

© 2026 UNH–IOL

Case 1: Media and Data Integrity Errors Definition (OF-FYI, M)

Test Procedure:

1. For each of the Status Code values defined in Table 17 and Table 18, configure the NVMe Host to issue a command to the NVMe Controller which will cause the controller to return the Status Code in the completion queue entry for the command.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status for the command.
2. Verify that the SCT field of the Status Field is set to 2h. Verify that the SC field of the Status field matches the expected Status Code.

Possible Problems: The NVMe specification does not explicitly state the exact conditions for when an NVMe Controller should use some of the defined Status Codes. Such status codes cannot be tested. Additionally, some of the Status Codes can only be used for optional commands which the NVMe controller may or may not support.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 5.6.1 and expect the same observable results.

1. **Write Fault (80h)**
2. **Unrecovered Read Error (81h)**
3. **End–to–End Guard Check Error (82h)**
4. **End–to–End Application Tag Check Error (83h)**
5. **End–to–End Reference Tag Check Erro (84h)**
6. **Compare Failure (85h)**
7. **Access Denied (86h)**
8. **Deallocated or Unwritten Logical Block (87h)**
9. **End–to-End Storage Tag Check Error (88h)**

© 2026 UNH–IOL

Group 6: Controller Architecture

Overview:
This section describes a method for performing conformance verification for NVMe products implementing the NVMe Controller Architecture.

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).

© 2026 UNH–IOL

Test 6.1 – Controller Level Reset – Controller Reset (PCIeC 6.3) (OF, M)

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

References:

Old Ref: NVMe Specification 7.3.2
NVM Express Base Specification 2.0a : 3.7.1

Resource Requirements:

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

Last Modification: June 24, 2015

Discussion: When a Controller Level Reset occurs, the Host and Controller are required to take specific action.

A Controller Reset is initiated when the CC.EN controller register field transitions from ‘1h’ to ‘0h’. This is performed by the NVMe Host by writing a value of ‘0h’ to the CC.EN while the CC.EN field is set to ‘1h’.

Test Setup: See Appendix A.

Test Procedure

1. Configure the NVMe Host to write a value of ‘0h’ to the CC.EN controller register field.
2. When the reset is complete, configure the NVMe Host to issue a Write and then a Read command to the NVMe Controller.

Observable Results:

1. Verify that the NVMe Controller is able to properly execute NVMe Write and Read commands after the reset is complete.
2. Verify that the NVMe Controller performs the following actions when the Controller Reset is initiated:
    a. The controller stops processing any outstanding Admin or I/O commands.
    b. All I/O Submission Queues are deleted.
    c. All I/O Completion Queues are deleted.
    d. The controller is brought to an Idle state. When this is complete, CSTS.RDY is cleared to ‘0h’.
    e. The Admin Queue registers (AQA, ASQ, or ACQ) are not reset as part of a controller reset. All other controller registers defined in section 3 of the NVMe Specification and internal controller state are reset.

Possible Problems: In NVMe Base 2.1 and future revisions, the host is no longer required to update the transport specific state and controller property state, set CC.EN to ‘1h’, wait for CSTS.RDY to be set to ‘1h’, configure the controller using Admin commands, create I/O Completion and Submission Queues, or proceed with normal I/O operations in order to continue after a Controller Level Reset, although it is highly recommended.

© 2026 UNH–IOL

Test 6.2 – Self-Reported Drive Power (OF, M)

Purpose: To verify that an NVMe Controller properly implements power measurement extended capabilities if supported.

References:

NVM Express Base Specification 2.3: 8.1.19

Resource Requirements:

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

Last Modification: February 5th, 2026

Discussion: The Self-Reported Drive Power feature enables the NVMe Controller to report its own power consumption to the Host without requiring external power instrumentation. This information is intended for host power management, thermal budgeting, fleet analytics, and validation. The Host and Controller must follow defined behaviors for when and how the value is produced, latched, read, and interpreted.

Test Setup: See Appendix A.

Case 1: Interval Power Measurement Value Set to Non-Zero Value When Required by Related Fields (OF-FYI, FYI)

Test Procedure

1. Configure the NVMe host to issue a Get Log Page command for Feature Identifiers Supported and Effects (Log Page Identifier 12h) to determine if Power Threshold feature (FID 24h) and Power Measurement feature (FID 25h) is supported. If neither FID 24h or FID 25h is supported, this test is not applicable.
2. Issue an Identify Controller command and record the value of PMS within CTRATT. If this is cleared to 0h, this test is not applicable.
3. Configure the NVMe Host to issue a Get Log Page command with LID=02h (SMART / Health Information). Obtain the Operational Lifetime Energy Consumed (OLEC) field. If OLEC is cleared to 0h, this test is not applicable.
4. Configure the NVMe host to issue a Set Features command with FID=24h (Power Threshold) and EPT=1.
5. Issue an Identify Controller command to obtain the IPMSR field.

Observable Results:

1. Verify that the value returned by the IPMSR field in step 5 is non-zero.

Case 2: Power Histogram Descriptors in Ascending Order (OF-FYI, FYI)

Test Procedure

1. Configure the NVMe Host to issue a Get Log Page command with LID=25h (Power Measurement), and for each Power Histogram Descriptor in the Power Histogram Descriptor list:
    a. Use the PHBLT field to record the Power Scale (PWRS) and the Power Value (PWRV).
    b. Multiply the value recorded in the PWRV field by the scale returned in the PWRS field to determine the power threshold in Watts of each Power Histogram Descriptor.

Observable Results:

1. Verify that the value returned in step 1b is in ascending order for each Power Histogram Descriptor.

Case 3: Power Measurement Action Field Start Measurement Clears Number Power Histogram Descriptors (OF-FYI, FYI)

Test Procedure

© 2026 UNH–IOL

1. Configure the NVMe Host to issue a Get Log Page command with LID=25h (Power Measurement). Record the Power Measurement Count (PMC) and Number of Power Histogram Descriptors Supported (NPHDS) fields. If either PMC or NPHDS is cleared to 0h, this test is not applicable.
2. Use the Get Log Page command from step 1 to get the Power Value (PWRV) field of the Power Histogram Bin Lower Threshold (PHBLT) of the first Power Histogram Descriptor.
3. Using the Get Log Page command from step 1, determine the power in the PHBLT field of the last Power Histogram Descriptor (NPHD - 1) by multiplying the value in the PWRV field by the scale returned from the PWRS field.
4. Configure the NVMe host to issue a Set Features command with FID=25h (Power Measurement) with Start Power Measurements (Action 1h).
5. Configure the NVMe Host to issue a Get Log Page command with LID=25h (Power Measurement).

Observable Results:

1. Verify that the power returned in step 3 is greater than the power of every other Power Histogram Descriptor.
2. Verify that the power returned by each Power Histogram Descriptor does not have any gaps between each Power Histogram Descriptor.
3. Verify that the NPHD field is cleared to 0 after the Set Features command in step 4.
4. Verify all commands completed successfully.

Case 4: Power Measurement Log Page Validation (OF-FYI, FYI)

Test Procedure

1. Configure the NVMe Host to issue a Get Log Page command with LID=25h (Power Measurement), and record the values returned.
2. Configure the NVMe host to issue a Set Features command with FID=25h (Power Measurement) and action set to 1h.
3. Configure the NVMe Host to issue a Get Log Page command with LID=25h (Power Measurement), and record the values returned.
4. Repeat steps 2-3 256 times.

Observable Results:

1. Verify that the VER field of the data structure returned in step 1 is cleared to 0h.
2. Verify that the PMGN field gets incremented after each Set Features command in step 2.
3. Verify that when the PMGN field rolls over when it exceeds a value of FFh after the Set Features in step 2.
4. Verify that the NCPDF, AIPWRV, and MIPWRV bits are cleared to 0 when the PMC field is cleared to 0h.
5. Verify that the SMTR field is cleared to 0 if the PME bit is cleared to 0h.
6. Verify that the PHBS field is set to 250.
7. Verify that the NPHD field is cleared to 0 when NPHDS field is cleared to 0h.
8. Verify that the PHDOC field is cleared to 0 if the PHDO bit is cleared to 0h.
9. Verify the PHBC field of the first Power Histogram Descriptor is cleared to 0h.
10. Verify all commands completed successfully.

Case 5: Product of Sample Rate Scale and Sample Rate Value <= 1 Second (OF-FYI, FYI)

Test Procedure

1. Configure the NVMe host to issue an Identify Controller command to obtain the Interval Power Measurement Sample Rate (IPMSR) field. If this field is cleared to 0, the test is not applicable.
2. Using the IPMSR field, store the Sample Rate Value (SRV) and Sample Rate Scale (SRS) fields. Multiply SRV by SRS and store the results.

Observable Results:

1. Verify that the value stored in step 2 is less than or equal to one second.
2. Verify that if the SRS field is cleared to 0, the SRV field is also cleared to 0h.

© 2026 UNH–IOL

Case 6: Power Threshold Exceeded Triggers Asynchronous Event (OF-FYI, FYI)

Test Procedure

1. Configure the NVMe host to issue an Get Log Page command for Feature Identifiers Supported and Effects (Log Page Identifier 12h) to determine if Power Threshold feature (FID 24h) is supported. If FID 24h is not listed as supported, this test is not applicable.
2. Configure the NVMe host to issue a Set Features command with FID=24h (Power Threshold) to set the Enable Power Threshold (EPT) bit to 1, PMTS = 0, PTS = 0.
3. Configure the NVMe host to issue an Get Features command with FID 24h and SEL = 00b.

Observable Results:

1. Verify that a Power Threshold Exceeded Asynchronous Event is triggered.
2. Verify that if a Power Threshold Exceeded Asynchronous Event is triggered, the EPT bit returned in the completion queue entry of step 3 is cleared to 0h.
3. Verify all commands completed successfully.

Case 7: Starting Power Measurement Clears and Changes Stop Measurement Time Remaining Field (OF-FYI, FYI)

Test Procedure

1. Configure the NVMe Host to issue a Get Log Page command with LID=25h (Power Measurement). If the Power Measurement Enable (PME) bit is set to 1h, configure the NVMe host to issue a Set Features command with FID=25h (Power Measurement) and action set to 0h to stop power measurements.
2. Record the Stop Measurement Time Remaining (SMTR) field.
3. Configure the NVMe host to issue a Set Features command with FID=25h (Power Measurement) and action set to 1h to start power measurements.
4. Configure the NVMe host to issue a Set Features command with FID=25h (Power Measurement) and action set to 0h to stop power measurements.
5. Configure the NVMe Host to issue a Get Log Page command with LID=25h (Power Measurement). Record the SMTR and PME fields.

Observable Results:

1. Verify that all commands completed successfully.
2. Verify that SMTR and PME are cleared to 0h after the Set Features command in step 4.

Case 8: Power Measurement Timer SMTR Decrements Over Time (OF-FYI, FYI)

Test Procedure

1. Configure the NVMe host to issue a Set Features command with FID=25h (Power Measurement) and action set to 1h, and SMT = 1.
2. Configure the NVMe Host to issue a Get Log Page command with LID=02h (SMART / Health Information). If the Interval Power Measurement (IPM) field is cleared to 0h, this test is not applicable.
3. Configure the NVMe Host to issue a Get Log Page command with LID=25h (Power Measurement). Record the Power Measurement Count (PMC) field.
4. Configure the NVMe host to issue a Get Features command with FID=25h (Power Measurement) with SEL = 00b. Record the Power Measurement Enable (PME) bit, Stop Measurement Time (SMT) field, and Stop Measurement Time Remaining (SMTR) field.

Observable Results:

1. Verify that the PMC field is set to 1h and the PME is set to 1h in step 4.
2. Verify that the SMTR field in step 4 is less than the SMT field set in step 1.
3. Verify all commands completed successfully.

© 2026 UNH–IOL

Case 9: Stop Measurement Time Greater Than The Maximum (OF-FYI, FYI)

Test Procedure

1. Issue an Identify Controller command to obtain the MSMT field. If this is cleared to 0h, this test is not applicable.
2. Issue a Set Features command with FID 25h, the Action field set to 1h, and SMT greater than MSMT.

Observable Results:

1. Verify that the Set Features command in step 2 returns a status code of 'Invalid Field in Command' (02h).
2. Verify all commands completed successfully.

Case 10: Recording Power Measurements Change Values in Log Page (OF-FYI, FYI)

Test Procedure

1. Configure the NVMe Host to issue a Set Features command for FID 25h with Action = 1 to start recording power measurements.
2. Issue a Get Log Page with LID 25h. Record the AIPWR, MIPWR, MIPWRT, the PHBC of the first Power Histogram Descriptor, PMC, and the PHDOC fields.
3. Configure the NVMe Host to issue a second Set Features command for FID 25h with Action = 1 to start recording power measurements.
4. Issue a second Get Log Page with LID 25h, record the AIPWR, MIPWR, MIPWRT, the PHBC of the first Power Histogram Descriptor, PMC, and the PHDOC fields. Record the maximum power indicated by the last Power Histogram Descriptor.
5. Issue a SMART / Health log and record the Interval Power Measurement value.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that the following fields are updated after the Set Features command in step 3:
    a. AIPWR
    b. MIPWR, if supported (i.e. not cleared to 0 in the Power Measurement Log Page),
    c. MIPWRT, if supported (i.e. not cleared to 0 in the Power Measurement Log Page),
    d. PHBC,
    e. PMC

© 2026 UNH–IOL

Test 6.3 – Controller Level Reset – Controller Reset (PCIeC 6.3) (OF, M)

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

References:

Old Ref: NVMe Specification 7.3.2
NVM Express Base Specification 2.0a : 3.7.1

Resource Requirements:

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

Last Modification: June 24, 2015

Discussion: When a Controller Level Reset occurs, the Host and Controller are required to take specific action.

A Controller Reset is initiated when the CC.EN controller register field transitions from ‘1h’ to ‘0h’. This is performed by the NVMe Host by writing a value of ‘0h’ to the CC.EN while the CC.EN field is set to ‘1h’.

Test Setup: See Appendix A.

Test Procedure

3. Configure the NVMe Host to write a value of ‘0h’ to the CC.EN controller register field.
4. When the reset is complete, configure the NVMe Host to issue a Write and then a Read command to the NVMe Controller.

Observable Results:

3. Verify that the NVMe Controller is able to properly execute NVMe Write and Read commands after the reset is complete.
4. Verify that the NVMe Controller performs the following actions when the Controller Reset is initiated:
    a. The controller stops processing any outstanding Admin or I/O commands.
    b. All I/O Submission Queues are deleted.
    c. All I/O Completion Queues are deleted.
    d. The controller is brought to an Idle state. When this is complete, CSTS.RDY is cleared to ‘0h’.
    e. The Admin Queue registers (AQA, ASQ, or ACQ) are not reset as part of a controller reset. All other controller registers defined in section 3 of the NVMe Specification and internal controller state are reset.

© 2026 UNH–IOL

Group 7: Reservations

Overview:
This section describes a method for performing conformance verification for NVMe products implementing NVMe Reservations. These tests are not applicable to devices that do not claim to support 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).

© 2026 UNH–IOL

Test 7.1 – Reservation Report Command (OF-FYI, M)

Purpose: To determine if an NVMe Controller properly reports the status of a reservation when processing a Reservation Report command.

References:

Old Ref: NVMe Specification 8.8, 5.14.1.15, 6.13, NVMe v1.3 ECN 003
NVM Express Base Specification 2.0a : 6.13
NVM Express NVM Command Set Specification 1.0a : 5.5

Resource Requirements:

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

Last Modification: April 8, 2019

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 (OF-FYI, M)

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 (REGSTRNT) field of the Reservation Status data structure is set to 0 to indicate that no hosts are registrants of the namespace.
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.

© 2026 UNH–IOL

5. Verify that the controller supports the Host Identifier feature.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

Case 2: Host is a Registrant (OF-FYI, M)

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 (REGSTRNT) 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 REGSTRNT 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.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

Case 3: 64 Bit Host Identifier (OF-FYI, FYI)

Test Procedure:

© 2026 UNH–IOL

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. 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' (18h).
2. Verify that the controller supports the Host Identifier feature.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

Case 4: 128 Bit Host Identifier (OF-FYI, 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' (18h).

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

© 2026 UNH–IOL

Case 5: Dynamic Controller Not Associated with Host (OF-FYI, 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. 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 known.

© 2026 UNH–IOL

Test 7.2 – Reservation Registration (OF-FYI, M)

Purpose: To determine if an NVMe Controller properly supports registering hosts via the Reservation Register command.

References:

Old Ref: NVMe Specification 8.8, 5.14.1.15, 6.11
NVM Express Base Specification 2.0a : 7.3

Resource Requirements:

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

Last Modification: April 8, 2019

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 (OF-FYI, M)

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.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of ‘Command Sequence Error’ (0Ch) on some later 2.0 Base drives.

© 2026 UNH–IOL

Case 2: Re-registration (OF-FYI, M)

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.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of ‘Command Sequence Error’ (0Ch) on some later 2.0 Base drives.

Case 3: Replace Registration Key (OF-FYI, M)

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:

© 2026 UNH–IOL

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 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.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

Case 4: Multiple Hosts Dual Port Devices Only (FYI)

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.

© 2026 UNH–IOL

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 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 (OF-FYI, 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.

© 2026 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 known.

Case 6: Controller ID field FFFFh when Registrant Namespace is Detached (OF-FYI, FYI)

Test Procedure:

1. Check the specification version supported by the DUT. If the DUT supports v2.1 or higher this test is applicable, otherwise this test is not applicable.
2. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management then this test is not applicable
3. Check the ONCS field to determine if the controller supports reservations. If the controller does not support reservations then this test is not applicable.
4. Configure the NVMe Host to issue a Set Features command with the Host Identifier feature to the NVMe Controller in order to register either a 64 or 128 bit Host Identifier for the controller.
5. 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.
6. Configure the NVMe Host to issue a Namespace Attachment command to detach the namespace with the reservation from the controller.
7. Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
8. Restore namespace configuration to pre-test conditions.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that the returned Registrant Data Structure from the command sent in step 7 successfully reports the reservation for the detached namespace and that the Controller ID (CNTLID) field is set to FFFFh.

© 2026 UNH–IOL

Test 7.3 – Unregistering (OF-FYI, M)

Purpose: To determine if an NVMe Controller properly supports reservations.

References:

Old Ref: NVMe Specification 8.8, 5.14.1.15, 6.11
NVM Express Base Specification 2.0a : 8.19.4

Resource Requirements:

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

Last Modification: April 8, 2019

Discussion: A host that is a registrant of a namespace may unregister with the namespace by executing a Reservation Register command on the namespace with the Reservation Register Action (RREGA) field set to 0001b (i.e., Unregister Reservation Key) and supplying its current reservation key in the CRKEY field. If the contents of the CRKEY field do not match the key currently associated with the host or if the host is not a registrant, then the command shall be aborted with a status of Reservation Conflict. A host may unregister without regard to its current reservation key value by setting the IEKEY bit to '1' in the Reservation Register command.

Successful completion of an unregister operation causes the host to no longer be a registrant of that namespace.

Unregistering due to preemption or a registration clear is verified in subsequent tests.

Test Setup: See Appendix A.

Case 1: Unregistering with Reservation Register Command (OF-FYI, M)

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 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.
        iii. 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.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

© 2026 UNH–IOL

Case 2: Unregistering due to Preemption Dual Port Devices Only (OF-FYI, FYI)

If a preemption occurs and there is no reservation held on the namespace, then execution of the Reservation Acquire command with Preempt action causes registrants whose reservation key match the value of the PRKEY field to be unregistered.

See the Preempting a Reservation test for more information on reservation preemption.

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.
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 2 to issue a Reservation Acquire command, setting the RACQA field to 001b (Preempt), and supplying the current reservation key associated with the NVMe Host 2 in the CRKEY field and setting the Preempt Reservation Key (PRKEY) field to the current reservation key associated with NVMe Host 1 to NVMe Controller 2 in order to preempt the NVMe Host 1's reservation.
7. Configure NVMe Host 2 to issue a Reservation Report command to NVMe Controller 2.
8. 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 NVMe Host 2 was not unregistered from the namespace and NVMe Host 1 was successfully unregistered from the namespace.

Possible Problems: A reliable means of performing this test has not been determined. Therefore, this test should not be included in any industry approved determination of conformance.

© 2026 UNH–IOL

Test 7.4 – Acquiring a Reservation (OF-FYI, M)

Purpose: To determine if an NVMe Controller properly allows acquisition of reservations via the Reservation Register command.

References:

Old Ref: NVMe Specification 8.8, 5.14.1.15, 6.10
NVM Express Base Specification 2.0a : 7.2, 8.19.5

Resource Requirements:

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

Last Modification: April 8, 2019

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 (OF-FYI, M)

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:
            i. 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.
            ii. Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
            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,

© 2026 UNH–IOL

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.

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.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

Case 2: Error Conditions (OF-FYI, M)

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.

© 2026 UNH–IOL

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 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' (02h).
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.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

Case 3: Multiple Hosts Dual Port Devices Only (OF-FYI, FYI)

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.

© 2026 UNH–IOL

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 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 known.

© 2026 UNH–IOL

Test 7.5 – Releasing a Reservation (OF-FYI, M)

Purpose: To determine if an NVMe Controller properly releases reservations.

References:

Old Ref: NVMe Specification 8.8, 5.14.1.15, 6.12
NVM Express Base Specification 2.0a : 7.4, 8.19.6

Resource Requirements:

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

Last Modification: April 8, 2019

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 (OF-FYI, M)

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.

Observable Results:

© 2026 UNH–IOL

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.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

Case 2: Reservation Release Command Error Conditions (OF-FYI, M)

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 '1h' 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 '1h' 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

© 2026 UNH–IOL

being released, supplying the correct reservation key that is associated with the host in the CRKEY field, and setting the IEKEY bit to '0h' to the NVMe Controller in order to release the reservation held by the host.

x. RES_REP4: 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_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.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

Case 3: Multiple Hosts Dual Port Devices Only (OF-FYI, FYI)

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.

© 2026 UNH–IOL

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.
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 (OF-FYI, M)

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.

© 2026 UNH–IOL

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of ‘Command Sequence Error’ (0Ch) on some later 2.0 Base drives.

© 2026 UNH–IOL

Test 7.6 – Preempting a Reservation (OF-FYI, FYI)

Purpose: To determine if an NVMe Controller properly preempts reservations.

References:

Old Ref: NVMe Specification 8.8, 5.14.1.15, 6.10
NVM Express Base Specification 2.0a : 7.2, 8.19.7

Resource Requirements:

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

Last Modification: April 8, 2019

Discussion: A host that is a registrant may preempt a reservation and/or registration by executing a Reservation Acquire command, setting the Reservation Acquire (RACQA) field to 001b (Preempt), and supplying the current reservation key associated with the host in the Current Reservation Key (CRKEY) field. The preempt actions that occur are dependent on the type of reservation held on the namespace, if any, and the value of the Preempt Reservation Key (PKEY) field in the command.

If the CRKEY value does not match, then the command is aborted with status Reservation Conflict.

The case of preemption when no reservation is held is covered in the Unregistering test.

Test Setup: See Appendix A.

Case 1: Write Exclusive - All Registrants or Exclusive Access - All Registrants Dual Port Devices Only (OF-FYI, FYI)

If the existing reservation type is Write Exclusive - All Registrants or Exclusive Access - All Registrants, then the actions performed by the command depend on the value of the PRKEY field as follows:

1. If the PRKEY field value is zero, then the following occurs as an atomic operation:
    a. All registrants other than the host that issued the command are unregistered,
    b. The reservation is released, and
    c. A new reservation is created for the host of the type specified by the Reservation Type (RTYPE) field in the command.
2. If the PRKEY value is non-zero, then each registrant whose reservation key matches the value of the PRKEY field are unregistered.

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.
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.

© 2026 UNH–IOL

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 the Current Reservation Key (CRKEY) field, and setting the Reservation Type (RTYPE) field to Write Exclusive - All Registrants or Exclusive Access - All Registrants to NVMe Controller 1 in order for NVMe Host 1 to acquire a reservation on the namespace.
7. Configure NVMe Host 2 to issue a Reservation Acquire command, setting the RACQA field to 001b (Preempt), and supplying the current reservation key associated with the NVMe Host 2 in the CRKEY field and setting the Preempt Reservation Key (PRKEY) field to 0 to NVMe Controller 2 in order to preempt NVMe Host 1's reservation.
8. Configure NVMe Host 2 to issue a Reservation Report command to NVMe Controller 2.
9. Configure NVMe Host 2 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 NVMe Host 2 in the Current Reservation Key (CRKEY) field to NVMe Controller 2 in order to release the reservation held by NVMe Host 2.
10. Configure NVMe Host 1 to issue a Reservation Register command with the Register Reservation Key action and supplying a reservation key in the NRKEY field to NVMe Controller 1 for the shared namespace in order to make the host a registrant of that namespace.
11. Configure NVMe Host 1 to issue a Reservation Acquire command with the Acquire action, supplying the reservation key currently associated with NVMe Host 1 in the CRKEY field, and setting the RTYPE field to Write Exclusive - All Registrants or Exclusive Access - All Registrants to NVMe Controller 1 in order for NVMe Host 1 to acquire a reservation on the namespace.
12. Configure NVMe Host 2 to issue a Reservation Acquire command with the Preempt action, and supplying the current reservation key associated with the NVMe Host 2 in the CRKEY field and setting the PRKEY field to the current reservation key associated with NVMe Host 1 to NVMe Controller 2 in order to preempt NVMe Host 1's reservation.
13. Configure NVMe Host 2 to issue a Reservation Report command to NVMe Controller 2.
14. 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 first Reservation Report command indicates that NVMe Host 2 was not unregistered from the namespace, NVMe Host 1 no longer holds a reservation and was successfully unregistered from the namespace, and that NVMe Host 2 holds a reservation of the type specified in the Reservation Acquire command with Preempt action.
3. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the second Reservation Report command indicates that NVMe Host 2 was not unregistered from the namespace and NVMe Host 1 was successfully unregistered from the namespace.

Case 2: Other Registration Types Dual Port Devices Only (OF-FYI, FYI)

If the existing reservation type is not Write Exclusive - All Registrants and not Exclusive Access - All Registrants (i.e. is Write Exclusive, Exclusive Access, Write Exclusive - Registrants Only, or Exclusive Access - Registrants Only), then the actions performed by the command depend on the value of the PRKEY as follows:

1. If the PRKEY field value matches the reservation key of the current reservation key of the current reservation holder, then the following occur as an atomic operation:
    a. The reservation holder is unregistered,
    b. The reservation is released, and
    c. A new reservation is created of the type specified by the Reservation Type (RTYPE) field in the command for the host as the reservation key holder.
2. If the PRKEY value does not match that of the current reservation holder and is not equal to zero, then each registrant whose reservation key matches the value of the value of the PRKEY field are unregistered.

© 2026 UNH–IOL

3. If the PRKEY value does not match that of the current reservation holder and is equal to zero, then the command shall be aborted with status Invalid Field in Command.

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 the Current Reservation Key (CRKEY) field, and setting the Reservation Type (RTYPE) field to Write Exclusive, Exclusive Access, Write Exclusive - Registrants Only, or Exclusive Access - Registrants Only to NVMe Controller 1 in order for NVMe Host 1 to acquire a reservation on the namespace.
7. Configure NVMe Host 2 to issue a Reservation Acquire command, setting the RACQA field to 001b (Preempt), and supplying the current reservation key associated with the NVMe Host 2 in the CRKEY field and setting the Preempt Reservation Key (PRKEY) field to the reservation key associated with NVMe Host 1 to NVMe Controller 2 in order to preempt NVMe Host 1's reservation.
8. Configure NVMe Host 2 to issue a Reservation Report command to NVMe Controller 2.
9. Configure NVMe Host 2 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 NVMe Host 2 in the Current Reservation Key (CRKEY) field to NVMe Controller 2 in order to release the reservation held by NVMe Host 2.
10. Configure NVMe Host 1 to issue a Reservation Register command with the Register Reservation Key action and supplying a reservation key in the NRKEY field to NVMe Controller 1 for the shared namespace in order to make the host a registrant of that namespace.
11. Configure NVMe Host 1 to issue a Reservation Acquire command with the Acquire action, supplying the reservation key currently associated with NVMe Host 1 in the CRKEY field, and setting the RTYPE field to Write Exclusive, Exclusive Access, Write Exclusive - Registrants Only, or Exclusive Access - Registrants Only to NVMe Controller 1 in order for NVMe Host 1 to acquire a reservation on the namespace.
12. Configure NVMe Host 2 to issue a Reservation Acquire command with the Preempt action, and supplying the current reservation key associated with the NVMe Host 2 in the CRKEY field and setting the PRKEY field to a reservation key other than NVMe Host 1 or NVMe Host 2's reservation keys to NVMe Controller 2.
13. Configure NVMe Host 2 to issue a Reservation Report command to NVMe Controller 2.
14. Configure NVMe Host 1 to issue a Reservation Acquire command with the Acquire action, supplying the reservation key currently associated with NVMe Host 1 in the CRKEY field, and setting the RTYPE field to Write Exclusive, Exclusive Access, Write Exclusive - Registrants Only, or Exclusive Access - Registrants Only to NVMe Controller 1 in order for NVMe Host 1 to acquire a reservation on the namespace.
15. Configure NVMe Host 2 to issue a Reservation Acquire command with the Preempt action, and supplying the current reservation key associated with the NVMe Host 2 in the CRKEY field and setting the PRKEY field to 0 to NVMe Controller 2 in order to preempt NVMe Host 1's reservation.
16. Configure NVMe Host 2 to issue a Reservation Report command to NVMe Controller 2.
17. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

© 2026 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 first Reservation Report command indicates that NVMe Host 2 was not unregistered from the namespace, NVMe Host 1 no longer holds a reservation and was successfully unregistered from the namespace, and that NVMe Host 2 holds a reservation of the type specified in the Reservation Acquire command with Preempt action.
3. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the second Reservation Report command indicates that both hosts are still registered and NVMe Host 1 still holds a reservation on the namespace.
4. Verify that the completion queue entry for the third Reservation Acquire command with Preempt action indicates status Invalid Field in Command.
5. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the third Reservation Report command indicates that both hosts are still registered and NVMe Host 1 still holds a reservation on the namespace.

Case 3: Self-preemption Dual Port Devices Only (OF-FYI, FYI)

A reservation holder may preempt itself using the above mechanism. When a host preempts itself, the following occurs as an atomic operation:

1. Registration of the host is maintained,
2. The reservation is released, and
3. A new reservation is created for the host of the type specified by the RTYPE field.

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. 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 of either WRITE EXCLUSIVE ALL REGISTRANTS or EXCLUSIVE ACCESS ALL REGISTRANTS.
        iii. Configure the NVMe Host to issue a Reservation Acquire command, setting the RACQA field to 001b (Preempt), and supplying the current reservation key associated with the host in the CRKEY field and setting the Preempt Reservation Key (PRKEY) field to 0 to the NVMe Controller in order to preempt the host's reservation.
        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 not unregistered from the namespace and that the host holds a reservation of the type specified in the Reservation Acquire command with Preempt action.

© 2026 UNH–IOL

Case 4: Preempt and Abort Dual Port Devices Only (OF-FYI, FYI)

A host may abort commands as a side effect of preempting a reservation by executing a Reservation Acquire command and setting the RACQA field to 010b (Preempt an Abort). The behavior of such a command is exactly the same as that described above with the RACQA field set to 001b (Preempt), except that commands that target the namespace are aborted by controllers associated with hosts whose reservation or registration is preempted. As with the Abort Admin command, abort as a side effect of preempting a reservation is best effort; the commands to abort may have already completed, currently be in execution, or may be deeply queued.

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 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 in order for NVMe Host 1 to acquire a reservation on the namespace.
7. Configure NVMe Host 1 to issue 10 NVMe Read commands to NVMe Controller 1.
8. Configure NVMe Host 2 to issue a Reservation Acquire command, setting the RACQA field to 010b (Preempt and Abort), and supplying the current reservation key associated with the NVMe Host 2 in the CRKEY field and setting the Preempt Reservation Key (PRKEY) field to the reservation key associated with NVMe Host 1 to NVMe Controller 2 in order to preempt NVMe Host 1's reservation.
9. Configure NVMe Host 2 to issue a Reservation Report command to NVMe Controller 2.
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. Determine the status of each of the NVMe commands issued by NVMe Host 1.
3. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the Reservation Report command indicates that the proper actions were taken according to the reservation type acquired by NVMe Host 1.

Case 5: Preempt Attempt when CRKEY does not Match Dual Port Devices Only (OF-FYI, 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.

© 2026 UNH–IOL

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 the Current Reservation Key (CRKEY) field, and setting the Reservation Type (RTYPE) field to Write Exclusive - All Registrants or Exclusive Access - All Registrants to NVMe Controller 1 in order for NVMe Host 1 to acquire a reservation on the namespace.
7. Configure NVMe Host 2 to issue a Reservation Acquire command, setting the RACQA field to 001b (Preempt), and supplying the current reservation key associated with the NVMe Host 2 in the CRKEY field and setting the Preempt Reservation Key (PRKEY) field to 0 to NVMe Controller 2 in order to preempt NVMe Host 1's reservation.
8. Configure NVMe Host 2 to issue a Reservation Report command to NVMe Controller 2.
9. Configure NVMe Host 2 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 NVMe Host 2 in the Current Reservation Key (CRKEY) field to NVMe Controller 2 in order to release the reservation held by NVMe Host 2.
10. Configure NVMe Host 1 to issue a Reservation Register command with the Register Reservation Key action and supplying a reservation key in the NRKEY field to NVMe Controller 1 for the shared namespace in order to make the host a registrant of that namespace.
11. Configure NVMe Host 1 to issue a Reservation Acquire command with the Acquire action, supplying the reservation key currently associated with NVMe Host 1 in the CRKEY field, and setting the RTYPE field to Write Exclusive - All Registrants or Exclusive Access - All Registrants to NVMe Controller 1 in order for NVMe Host 1 to acquire a reservation on the namespace.
12. Configure NVMe Host 2 to issue a Reservation Acquire command with the Preempt action, and supplying an incorrect key in the CRKEY field and setting the PRKEY field to the current reservation key associated with NVMe Host 1 to NVMe Controller 2 in order to preempt NVMe Host 1's reservation.
13. Configure NVMe Host 2 to issue a Reservation Report command to NVMe Controller 2.
14. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations.

Observable Results:

1. Verify that the Reservation Acquire command with Preempt Action and the incorrect CRKEY is aborted with status Reservation Conflict.

Possible Problems: A reliable means of performing this test has not been determined. Therefore, this test should not be included in any industry approved determination of conformance.

© 2026 UNH–IOL

Test 7.7 – Clearing a Reservation (OF-FYI, M)

Purpose: To determine if an NVMe Controller properly supports clearing reservations.

References:

Old Ref: NVMe Specification 8.8, 5.14.1.15, 6.12
NVM Express Base Specification 2.0a : 8.19.8

Resource Requirements:

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

Last Modification: November 26, 2018

Discussion: A host that is a registrant may clear a reservation (i.e. force the release of a reservation held on the namespace and unregister all registrants) by executing a Reservation Release command, setting the Reservation Release Action (RRELA) field to 001b (i.e. Clear), and supplying the current reservation key associated with the host in the Current Reservation Key (CRKEY) field.

When a reservation is cleared, the following occur as an atomic operation:

1. On held on the namespace is released, and
2. All registrants are unregistered from the namespace.

Test Setup: See Appendix A.

Case 1: Basic Operation with Reservation Release Command (OF-FYI, M)

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.
        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.
        ii. Configure the NVMe Host to issue a Reservation Release command, setting the Reservation Release Action (RRELA) field to 001b (i.e. Clear) and supplying the current reservation key associated with the host in the Current Reservation Key (CRKEY) field to the NVMe Controller in order to clear the reservation.
        iii. 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.

© 2026 UNH–IOL

2. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the Reservation Report command indicates that the host no longer holds a reservation on the namespace and was successfully unregistered from the namespace.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

Case 2: Error Conditions (OF-FYI, M)

If the CRKEY value does not match that used by the host to register with the namespace, then the command shall be aborted with status Reservation Conflict. A host may clear 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 host is not a registrant, then the command shall be aborted with a status of Reservation Conflict.

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 Release command, setting the Reservation Release Action (RRELA) field to 001b (i.e. Clear) and setting the Current Reservation Key (CRKEY) field to a random reservation key value to the NVMe Controller.
        ii. 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.
        iii. Configure the NVMe Host to issue a Reservation Release command with the Clear action and supplying any reservation key other than the current reservation key associated with the host in the CRKEY field to the NVMe Controller.
        iv. Configure the NVMe Host to issue a Reservation Report command to the NVMe Controller.
        v. Configure the NVMe Host to issue a Reservation Release command with the Clear action, supplying any reservation key other than the current reservation key associated with the host in the CRKEY field, and setting the IEKEY bit to '1' to the NVMe Controller in order to clear the reservation.
        vi. 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 the first Reservation Release command with Clear action indicates status Reservation Conflict.
3. Verify that the completion queue entry for the second Reservation Release command with Clear action indicates status Reservation Conflict.
4. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the first Reservation Report command indicates that the host is still a registrant of the namespace.
5. Verify that the Reservation Status data structure returned by the NVMe Controller after completing the second Reservation Report command indicates that the host was successfully unregistered from the namespace.

Possible Problems: A reliable means of performing this test has not been determined. Therefore, this test should not be included in any industry approved determination of conformance. When setting the Host Identifier and it currently is

© 2026 UNH–IOL

non-zero value at the start of the test, the set feature will be aborted with a status of ‘Command Sequence Error’ (0Ch) on some later 2.0 Base drives.

© 2026 UNH–IOL

Test 7.8 – Command Behavior with Different Reservation Types (OF-FYI, M)

Purpose: To determine if an NVMe Controller exhibits proper command behavior in the presence of different reservation types.

References:

Old Ref: NVMe Specification 8.8, 5.14.1.15
NVM Express Base Specification 2.0a : 7.2, 8.19.1

Resource Requirements:

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

Last Modification: April 8, 2019

Discussion: NVMe supports six types of reservations:

- Write Exclusive
- Exclusive Access
- Write Exclusive - Registrants Only
- Exclusive Access - Registrants Only
- Write Exclusive - All Registrants
- Exclusive Access - All Registrants

The differences between these reservation types are: the type of access that is excluded (i.e., writes or all accesses), whether registrants have the same access rights as the reservation holder, and whether registrants are also considered to be reservation holders.

For the purposes of reservation types, the following commands are considered to be in the NVM Read Command Group:

- Read
- Compare
- Security Receive

And the following commands are considered to be in the NVM Write Command Group:

- Write
- Write Uncorrectable
- Dataset Management
- Flush
- Format NVM
- Namespace Attachment
- Namespace Management
- Security Send

Additionally, certain reservation commands have specific behavior and all other commands shall be allowed regardless of reservation status or type. The behavior of vendor specific commands is vendor specific.

Test Setup: See Appendix A.

Case 1: Write Exclusive Dual Port Devices Only (OF-FYI, M)

A Write Exclusive reservation disallows commands from the NVM Write Command Group by any host other than the registration holder. Any commands from the NVM Read Command Group are still allowed by any host.

This test requires there to be at least two controllers with a shared namespace in the NVM Subsystem.

Test Procedure:

© 2026 UNH–IOL

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 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 Write Exclusive to NVMe Controller 1 in order for NVMe Host 1 to acquire a reservation of type Write Exclusive.
6. Configure each host to issue each of the commands in the NVM Read Command Group above to their respective controllers.
7. Configure each host to issue each of the commands in the NVM Write Command Group above to their respective controllers.
8. 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.
9. Configure NVMe Host 2 to issue each of the commands in the NVM Read Command Group above to NVMe Controller 2.
10. Configure NVMe Host 2 to issue each of the commands in the NVM Write Command Group above to NVMe Controller 2.
11. 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 entries for all commands sent by NVMe Host 1 indicate status Successful Completion.
3. Verify that the completion queue entries for all commands from the NVM Read Command Group sent by NVMe Host 2 indicate status Successful Completion.
4. Verify that the completion queue entries for all commands from the NVM Write Command Group sent by NVMe Host 2 indicate status Reservation Conflict.

Case 2: Exclusive Access Dual Port Devices Only (OF-FYI, M)

An Exclusive Access reservation disallows commands from both the NVM Write Command Group and the NVM Read Command Group by any host other than the registration holder.

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

© 2026 UNH–IOL

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 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 Exclusive Access to NVMe Controller 1 in order for NVMe Host 1 to acquire a reservation of type Exclusive Access.
6. Configure each host to issue each of the commands in the NVM Read Command Group above to their respective controllers.
7. Configure each host to issue each of the commands in the NVM Write Command Group above to their respective controllers.
8. 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.
9. Configure NVMe Host 2 to issue each of the commands in the NVM Read Command Group above to NVMe Controller 2.
10. Configure NVMe Host 2 to issue each of the commands in the NVM Write Command Group above to NVMe Controller 2.
11. 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 entries for all commands sent by NVMe Host 1 indicate status Successful Completion.
3. Verify that the completion queue entries for all commands sent by NVMe Host 2 indicate status Reservation Conflict.

Case 3: Write Exclusive - Registrants Only or Write Exclusive - All Registrants Dual Port Devices Only (OF-FYI, M)

A Write Exclusive - Registrants Only or a Write Exclusive - All Registrants reservation disallows commands from the NVM Write Command Group by any non-registrant of the namespace. Any commands from the NVM Read Command Group are still allowed by any host and registrants of the namespace are still allowed to issue commands from the NVM Write Command Group.

The difference between the Write Exclusive - Registrants Only and Write Exclusive - All Registrants reservation types is that all registrants are also considered reservation holders with the Write Exclusive - All Registrants type.

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 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

© 2026 UNH–IOL

Current Reservation Key (CRKEY) field, and setting the Reservation Type (RTYPE) field to Write Exclusive - Registrants Only or Write Exclusive - All Registrants to NVMe Controller 1 in order for NVMe Host 1 to acquire a reservation of the specified type.

6. Configure each host to issue each of the commands in the NVM Read Command Group above to their respective controllers.
7. Configure each host to issue each of the commands in the NVM Write Command Group above to their respective controllers.
8. 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.
9. Configure NVMe Host 2 to issue each of the commands in the NVM Read Command Group above to NVMe Controller 2.
10. Configure NVMe Host 2 to issue each of the commands in the NVM Write Command Group above to NVMe Controller 2.
11. 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 entries for all commands sent by NVMe Host 1 indicate status Successful Completion.
3. Verify that the completion queue entries for all commands from the NVM Read Command Group sent by NVMe Host 2 indicate status Successful Completion.
4. Verify that the completion queue entries for all commands from the NVM Write Command Group sent by NVMe Host 2 while not a registrant of the namespace indicate status Reservation Conflict.
5. Verify that the completion queue entries for all commands from the NVM Read Command Group sent by NVMe Host 2 while a registrant of the namespace indicate status Successful Completion.

Case 4: Exclusive Access - Registrants Only or Exclusive Access - All Registrants Dual Port Devices Only (OF-FYI, M)

An Exclusive Access - Registrants Only or an Exclusive Access - All Registrants reservation disallows commands from both the NVM Write Command Group and NVM Read Command Group by any non-registrant of the namespace. Registrants of the namespace are still allowed to issue commands from either command group.

The difference between the Exclusive Access - Registrants Only and Exclusive Access - All Registrants reservation types is that all registrants are also considered reservation holders with the Write Exclusive - All Registrants type.

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 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

© 2026 UNH–IOL

the Current Reservation Key (CRKEY) field, and setting the Reservation Type (RTYPE) field to Exclusive Access - Registrants Only or Exclusive Access - All Registrants to NVMe Controller 1 in order for NVMe Host 1 to acquire a reservation of the specified type.

6. Configure each host to issue each of the commands in the NVM Read Command Group above to their respective controllers.
7. Configure each host to issue each of the commands in the NVM Write Command Group above to their respective controllers.
8. 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.
9. Configure NVMe Host 2 to issue each of the commands in the NVM Read Command Group above to NVMe Controller 2.
10. Configure NVMe Host 2 to issue each of the commands in the NVM Write Command Group above to NVMe Controller 2.
11. 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 entries for all commands sent by NVMe Host 1 indicate status Successful Completion.
3. Verify that the completion queue entries for all commands sent by NVMe Host 2 while not a registrant of the namespace indicate status Reservation Conflict.
4. Verify that the completion queue entries for all commands sent by NVMe Host 2 while a registrant of the namespace indicate status Successful Completion.

Possible Problems: A reliable means of performing this test has not been determined. Therefore, this test should not be included in any industry approved determination of conformance.

© 2026 UNH–IOL

Test 7.9 – Reservation Notification Log Page (OF-FYI, FYI)

Purpose: To determine if an NVMe Controller properly uses the Reservation Notification Log Page.

References:

Old Ref: NVMe Specification 5.14.1.9.1, NVMe v1.3 ECN 001
NVM Express Base Specification 2.0a : 5.16.1.24

Resource Requirements:

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

Last Modification: April 8, 2019

Discussion: The Reservation Notification log page reports one log page from a time ordered queue of reservation notification log pages, if available. A new Reservation Notification log page is created and added to the end of the queue of reservation notifications whenever an unmasked reservation notification occurs on any namespace that is attached to the controller. The Get Log Page command:

- Returns a data buffer containing a log page corresponding to the oldest log page in the reservation notification queue (i.e., the log page containing the lowest Log Page Count field; accounting for wrapping); and
- Removes that Reservation Notification log page from the queue.

If there are no available Reservation Notification log page entries when a Get Log command is issued, then an empty log page (i.e., all fields in the log page cleared to 0h) shall be returned.

Test Setup: See Appendix A.

Case 1: Retrieve Log (OF-FYI, 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. Configure the Testing Station such that it is acting as 2 hosts accessing a shared namespace on the DUT.
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 set its Host Identifier for that controller.
    b. For each shared namespace attached to the NVMe Controller:
        i. Configure at least 2 NVMe Hosts 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 a shared namespace in order to make the hosts registrants 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 same 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 same NVMe Host to issue a Reservation Report command to the NVMe Controller.

© 2026 UNH–IOL

4. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations. The Reservation release command is expected to generate a Reservation Notification.
5. Repeat steps 1-3 a second time.
6. Perform a Get Log Page for the Reservation Notification Log Page.

Observable Results:

1. Verify that the Reservation Notification Log Page returned indicated the following:
    a. Log Page Count of 2
    b. Log Page Type of Reservation Released.
    c. The number of Available Log Pages was 1.
    d. Correct Namespace ID.

Case 2: Empty Log (OF-FYI, 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. Configure the Testing Station such that it is acting as 2 hosts accessing a shared namespace on the DUT.
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 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 one 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 Write Exclusive or Exclusive Access 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.
4. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations. The Reservation release command is expected to generate a Reservation Notification.
5. Perform a Get Log Page for the Reservation Notification Log Page.
6. Perform a second Get Log Page for the Reservation Notification Log Page.

Observable Results:

1. Verify that the second Reservation Notification Log Page was returned with all 0's, since the Log Page was empty.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of 'Command Sequence Error' (0Ch) on some later 2.0 Base drives.

Case 3: Wrapped Log Count (OF-FYI, 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.

© 2026 UNH–IOL

2. Configure the Testing Station such that it is acting as 2 hosts accessing a shared namespace on the DUT.
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 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 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.
4. Perform a Reservation Release command with the RRELA field set to 001b (Clear) for all existing reservations. The Reservation release command is expected to generate a Reservation Notification.
5. Repeat steps 1-3 1000 times until the Reservation Notification Log Queue wraps.
6. Perform a Get Log Page for the Reservation Notification Log Page.

Observable Results:

1. If the DUT claims support for NVMe v1.4 or later, verify that if the Reservation Notification log 64 bit Log Page Count wrapped, it rolled over to a new value of 1. Otherwise verify that the Log Page Count incremented correctly.

Possible Problems: When setting the Host Identifier and it currently is non-zero value at the start of the test, the set feature will be aborted with a status of ‘Command Sequence Error’ (0Ch) on some later 2.0 Base drives.

© 2026 UNH–IOL

Group 8: Namespace Management

Overview:
This section describes a method for performing conformance verification for NVMe products implementing NVMe Namespace Management features.

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).

© 2026 UNH–IOL

Test 8.1 – Namespace Management Identify Command (PCIeC 9.1) (OF-FYI, M)

Purpose: To determine if an NVMe Controller properly implements the features of the Identify command relating to namespace management.

References:

Old Ref: NVMe Specification 6.1.6, 8.12, NVMe v1.3 ECN 003
NVM Express Base Specification 2.0a : 3.2.1.6, 8.11

Resource Requirements:

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

Last Modification: May 19, 2020

Discussion: Additional CNS values were added to the Identify command relating to namespace management.

Test Setup: See Appendix A.

Case 1: CNS 10h & 11h – Namespace Lists (OF, M)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 10h and CDW1.NSID 00h to the NVMe Controller Under Test in order to receive back a Namespace List containing all allocated namespaces.
3. For each namespace returned in the Namespace List from the previous step, configure the NVMe Host to issue an Identify command specifying CNS value 11h and setting CDW1.NSID to the namespace identifier of the namespace to the NVMe Controller Under Test in order to receive back an Identify Namespace data structure for the specified namespace.

Observable Results:

1. Verify that the requested data structures are posted to the memory buffer indicated in PRP Entry 1, PRP Entry 2, and Command Dword 10, and that a command completion queue entry is posted to the Admin Completion Queue for each command issued to the NVMe Controller Under Test.
2. Verify that all Identify Namespace data structures returned by the controller are not zero filled.
3. Verify that unused entries in the Namespace List are zero filled.

Case 2: CNS 12h – Controller List – Controllers Attached to a Namespace (OF, M)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 02h and CDW1.NSID 00h to the NVMe Controller Under Test in order to receive back a Namespace List containing active namespaces.
3. For each namespace returned in the Namespace List from the previous step, configure the NVMe Host to issue an Identify command specifying CNS value 12h, CDW10.CNTID value 00h, and setting CDW1.NSID to the namespace identifier of the namespace to the NVMe Controller Under Test in order to receive back a Controller List containing the controller identifiers of all controllers attached to the namespace.

Observable Results:

© 2026 UNH–IOL

1. Verify that the requested data structures are posted to the memory buffer indicated in PRP Entry 1, PRP Entry 2, and Command Dword 10, and that a command completion queue entry is posted to the Admin Completion Queue for each command issued to the NVMe Controller Under Test.
2. Verify that the controller identifier of the NVMe Controller Under Test is contained within each Controller List returned by the controller.
3. Verify that each Controller List contains valid values and that unused entries in the Controller List are zero filled.

Case 3: CNS 13h – Controller List – All Controllers (OF-FYI, M)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 13h and CDW10.CNTID value 00h to the controller in order to receive back a Controller List containing the controller identifiers of all controllers in the NVM subsystem. 0h is a valid controller identifier.

Observable Results:

1. Verify that the requested data structure is posted to the memory buffer indicated in PRP Entry 1, PRP Entry 2, and Command Dword 10, and that a command completion queue entry is posted to the Admin Completion Queue.
2. Verify that the controller identifier of the NVMe Controller Under Test is contained within the Controller List returned by the controller.
3. Verify that the Controller List contains valid values and that unused entries in the Controller List are zero filled.

Case 4: Common Namespace Data Structure (OF, M)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 00h and CDW1.NSID value FFFFFFFFh to the controller in order to receive back an Identify Namespace data structure that specifies capabilities that are common across namespaces.

Observable Results:

1. Verify that the requested data structure is posted to the memory buffer indicated in PRP Entry 1, PRP Entry 2, and Command Dword 10, and that a command completion queue entry is posted to the Admin Completion Queue.

Case 5: NSID Uniqueness (OF, M)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 02h and CDW1.NSID value 00h to the controller in order to receive back a list of Namespace IDs. Repeat this setup until all NSIDs are retrieved. For some systems this may require more than 1 Identify Command with CNS=02h.
3. Check if all namespaces have unique NSIDs.
    a. If not all NSIDs are unique, check that the not-unique NSIDs are in fact shared namespaces by writing a pattern to the namespace through one controller, and then checking that that same pattern can be read through all other controller that are sharing that namespace.

Observable Results:

1. Verify that all namespaces have unique NSIDs.

© 2026 UNH–IOL

Case 6: Namespace management not supported (OF, M)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT supports Namespace Management then this test is not applicable.
2. This test is only applicable to devices that support NVMe v1.3 or higher.
3. Configure the NVMe Host to issue an Identify command specifying CNS value 00h and CDW1.NSID value FFFFFFFFh to the controller.

Observable Results:

1. Verify that the controller returned status 'Invalid Namespace or Format' (0Bh) to the Identify command.

Possible Problems: None known.

Case 7: Namespace Management, OAES bit 8 (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify Command to CNS=01h, check the OACS field to determine if the DUT supports Namespace Management. If OACS Bit 3 is cleared to 0h, meaning Namespace Management is not supported, then this test is not applicable.

Observable Results:

1. Verify the command completes successfully.
2. Verify Bit 8 of the OAES field is set to 1, to indicate support for Namespace Attribute Changed asynchronous event.

Possible Problems: None known.

Case 8: Namespace Identifier with Invalid Namespace or Format (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue an Identify Command with CNS=01h, check the OACS field to determine if the DUT supports Namespace Management. If OACS Bit 3 is cleared to 0h, meaning Namespace Management is not supported, then this test is not applicable.
2. Configure the NVMe Host to issue Identify commands (02h, 10h, 12h) with Namespace Identifier set to FFFFFFFFh.

Observable Results:

1. Verify that each of the Identify (CNS 12h) command is aborted with a status code of Invalid Field in Command and that the Identify (CNS 02h & 10h) command is aborted with a status code of Invalid Namespace or Format.

Possible Problems: None known.

© 2026 UNH–IOL

Test 8.2 – Namespace Management Command (PCIeC 9.2) (OF-FYI, M)

Purpose: To determine if an NVMe Controller properly implements the Namespace Management command.

References:

NVMe Specification 6.1, 8.12, NVMe v1.3 ECN 002
NVM Express Base Specification 2.0a : 3.2.1.6, 8.11

Resource Requirements:

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

Last Modification: April 9, 2019

Discussion: The Namespace Management command is used for managing namespaces. It can be used for creating and deleting namespaces. The Select (SEL) field of Command Dword 10 is used to specify the type of operation for the Namespace Management command.

The Namespace Management command uses the PRP Entry 1, PRP Entry 2, and Dword 10 fields. All other command specific fields are reserved.

Test Setup: See Appendix A.

Case 1: Namespace Creation – Exceed Number Supported (OF-FYI, M)

The create operation creates a new namespace. The new namespace will not be attached to any controller. The Namespace Attachment command must be used to attach a new namespace to a controller if desired.

The data structure used for the created operation is defined in the NVMe Specification and has the same format as the Identify Namespace data structure. However, the host is only allowed to set the following fields in the data structure:

- Namespace Size (NSZE)
- Namespace Capacity (NCAP)
- Formatted LBA Size (FLBAS)
- End-to-end Data Protection Type Settings (DPS)
- Namespace Multi-path I/O and Namespace Sharing Capabilities (NMIC)

All other fields are reserved and shall be cleared to 0.

For the creation operation, the CDW1.NSID field is reserved and shall be cleared to 0. The controller selects the next available namespace identifier to use for the new namespace. The namespace identifier of the new namespace is returned in Dword 0 of the completion queue entry for the command.

If creation of a new namespace would cause the number of namespace to exceed the number of supported namespaces, then the controller shall return status Namespace Identifier Unavailable.

If the size of the new namespace exceeds the amount of available space on the device, then the controller shall return status Namespace Insufficient Capacity and the Command Specific Information field of the Error Information Log specifies the total amount of NVM capacity required to create the namespace in bytes.

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management then this test is not applicable.
2. Check the FNA field Bit 0. If set to 1, ensure any namespaces created in the following steps are created with the same format as all other existing namespaces on the device, if any. If set to 0, the device supports format on a namespace basis and it is not necessary to ensure that any namespaces created in the following steps are created with the same format as all other existing namespaces on the device.

© 2026 UNH–IOL

3. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create) and valid values for the attached data structure to the NVMe Controller Under Test in order to create a new namespace.
4. Configure the NVMe Host to issue an Identify command specifying CNS field value 11h and CDW1.NSID value set to the namespace identifier of the newly created namespace to the NVMe Controller Under Test in order to receive back the Identify Namespace data structure for that namespace.
5. Configure the NVMe Host to issue Namespace Management commands specifying Select field value 0h (Create) and valid values for the attached data structure to the controller until the number of namespaces exceeds the number of namespaces supported by the NVM subsystem.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status for the command.
2. Verify that a new inactive namespace is created and that the capabilities returned in the Identify Namespace data structure match the appropriate values set using the Namespace Management command.
3. Verify that the Namespace Identifier Unavailable status is returned when the number of namespaces exceeds the number of supported namespaces.

Case 2: Namespace Deletion (OF-FYI, M)

The delete operation deletes an existing namespace. As a side effect of the delete operation, the namespace is detached from any controller as it is no longer present in the system. Namespaces detached due to a delete operation will become an inactive namespace.

There is no data structure transferred for the delete operation.

The CDW1.NSID field specifies the namespace to delete.

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management then this test is not applicable.
2. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 1h (Delete) and CDW1.NSID field set to an active namespace in order to delete the specified namespace. If no Namespace exists, then the NVMe Host should first create a Namespace.
3. Configure the NVMe Host to issue an Identify command specifying CNS field value 02h and CDW1.NSID value 00h to the controller in order to receive back a Namespace List containing active namespace IDs attached to the controller.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status for the command.
2. Verify that the deleted namespace is not contained within the Namespace List.

Case 3: Namespace Creation – Insufficient Capacity (OF-FYI, M)

The create operation creates a new namespace. The new namespace will not be attached to any controller. The Namespace Attachment command must be used to attach a new namespace to a controller if desired.

The data structure used for the created operation is defined in the NVMe Specification and has the same format as the Identify Namespace data structure. However, the host is only allowed to set the following fields in the data structure:

- Namespace Size (NSZE)
- Namespace Capacity (NCAP)
- Formatted LBA Size (FLBAS)
- End-to-end Data Protection Type Settings (DPS)
- Namespace Multi-path I/O and Namespace Sharing Capabilities (NMIC)

© 2026 UNH–IOL

All other fields are reserved and shall be cleared to 0.

For the creation operation, the CDW1.NSID field is reserved and shall be cleared to 0. The controller selects the next available namespace identifier to use for the new namespace. The namespace identifier of the new namespace is returned in Dword 0 of the completion queue entry for the command.

If creation of a new namespace would cause the number of namespace to exceed the number of supported namespaces, then the controller shall return status Namespace Identifier Unavailable.

If the size of the new namespace exceeds the amount of available space on the device, then the controller shall return status Namespace Insufficient Capacity and the Command Specific Information field of the Error Information Log specifies the total amount of NVM capacity required to create the namespace in bytes.

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management then this test is not applicable.
2. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create) and valid values for the attached data structure to the NVMe Controller Under Test in order to create a new namespace.
3. Configure the NVMe Host to issue an Identify command specifying CNS field value 11h and CDW1.NSID value set to the namespace identifier of the newly created namespace to the NVMe Controller Under Test in order to receive back the Identify Namespace data structure for that namespace.
4. Configure the NVMe Host to issue Namespace Management commands specifying Select field value 0h (Create) and valid values for the attached data structure to the controller until the amount of available space on the device is exceeded.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status for the command.
2. Verify that a new inactive namespace is created and that the capabilities returned in the Identify Namespace data structure match the appropriate values set using the Namespace Management command.
3. Verify that the Namespace Insufficient Capacity status is returned if the size of a new namespace would exceed the available space on the device.

Case 4: Namespace Deletion when NSID=FFFFFFFFh (OF-FYI, M)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS field value 02h and CDW1.NSID value 00h to the controller in order to receive back a Namespace List containing active namespace IDs attached to the controller. Verify that no valid Namespaces exist.
3. Perform a Namespace Management command with action of Delete, and NSID=FFFFFFFFh.

Observable Results:

1. Verify that after the completion of the Namespace Management command, the controller posts a completion queue entry to the appropriate Completion Queue indicating status of success for the command. DUT's claiming to support NVMe v1.3 or earlier may implement this functionality, but DUT's claiming to support NVMe v1.4 or higher must implement this functionality.

Possible Problems:

<== For Case 1, there may not be enough capacity remaining in the NVM subsystem or the number of namespace limit may already be reached which would prevent creation of a new namespace. In this case, the NVMe Host may delete an existing namespace to allow for creation of a new namespace.

© 2026 UNH–IOL

<== For Case 2, there may be no active namespaces available for deletion. In this case, the NVMe Host may create and set up an active namespace in order to perform procedure steps which require such a namespace.

<== The NVMe Host should return the state of the device to its state from prior to running these test cases in order to prevent issues with other tests.

Case 5: Creation with NVMSETID & ENDGID Cleared to Zero (OF-FYI, M)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 0h. Store returned list information
3. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create) and valid values for the attached data structure to the NVMe Controller Under Test in order to create a new namespace. Configure the NVM Set Identifier (NVMSETID) and Endurance Group Identifier (ENDGID) are both cleared to '0' in the Namespace Management – Host Software Specified Fields data structure for the creation
4. Configure the NVMe Host to issue an Identify command specifying CNS field value 11h and CDW1.NSID value set to the namespace identifier of the newly created namespace to the NVMe Controller Under Test in order to receive back the Identify Namespace data structure for that namespace.
5. Configure the NVMe Host to issue an Endurance Group Get Log with LID=09h and CDW1.NSID from step 4.
6. Configure the NVMe Host to issue an Identify command specifying CNS value of 04h, NVM Set List.

Observable Results:

1. Verify that a new inactive namespace is created and that the capabilities returned in the Identify Namespace data structure match the appropriate values set using the Namespace Management command
2. Verify that the controller creates the new namespace will choose the Endurance Group and NVMSet from which to allocate the namespace. Verify that the Endurance Group Get Log completes successfully and that the Identify Command NVM Set List contains the NVM Set Id

Case 6: Creation with NVMSETID=0 & valid ENDGID (OF-FYI, M)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command specifying CNS value 19h, using the Endurance Group Identifier equal to '0'. Store returned list information.
3. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create) and valid values for the attached data structure to the NVMe Controller Under Test in order to create a new namespace. Configure the NVM Set Identifier (NVMSETID) is cleared to '0' and Endurance Group Identifier (ENDGID) is set to a valid endurance group in the Namespace Management – Host Software Specified Fields data structure for the creation
4. Configure the NVMe Host to issue an Identify command specifying CNS field value 11h and CDW1.NSID value set to the namespace identifier of the newly created namespace to the NVMe Controller Under Test in order to receive back the Identify Namespace data structure for that namespace.
5. Configure the NVMe Host to issue an Identify command specifying CNS value 19h, using the Endurance Group Identifier equal to '0'. Compare the previous returned list and this returned list for differences for CNS value 19h.

Observable Results:

1. Verify that the controller creates the new namespace and will choose the NVM Set within the Specified Endurance Group from which to allocate the namespace. This is accomplished by comparing the CNS value 19h returns in procedures steps above

© 2026 UNH–IOL

Case 7: Creation with NVMSETID=non-zero value & ENDGID=0 (OF-FYI, FYI)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
2. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create) and valid values for the attached data structure to the NVMe Controller Under Test in order to create a new namespace. Configure the NVM Set Identifier (NVMSETID) to be a non-zero value and Endurance Group Identifier (ENDGID) to be cleared to '0' in the Namespace Management – Host Software Specified Fields data structure for the creation

Observable Results:

1. Verify that the controller aborts the Namespace Management command with a status of 'Invalid Field in Command' (02h).

Case 8: Creation with valid NVMSETID existing in ENDGID (OF-FYI, FYI)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
2. Perform an Identify Command with CNS 0h, using the default namespace id, to retrieve the ENDGID and NVMSETID for that namespace.
3. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create) and valid values for the attached data structure to the NVMe Controller Under Test in order to create a new namespace. Configure the NVM Set Identifier (NVMSETID) to be set to the valid NVM Set Identifier in step 2 and Endurance Group Identifier (ENDGID) to be set to the valid Endurance Group in step 2 in the Namespace Management – Host Software Specified Fields data structure for the creation
4. Perform a get on the Identify Namespace data structure for an Allocated Namespace ID (CNS 11h) and retrieve the new ENDGID and NVMSETID.

Observable Results:

1. Verify that the Namespace Management command completed successfully
2. Verify that the controller created the namespace in the specified NVM Set for the associated Endurance Group

Case 9: Creation with valid existing in ENDGID, Invalid NVMSETID (OF-FYI, FYI)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
2. Perform an Identify Command with CNS 0h, using the default namespace id, to retrieve the ENDGID and NVMSETID for that namespace.
3. Configure the NVMe Host to issue a Namespace Management command specifying Select field value 0h (Create) and valid values for the attached data structure to the NVMe Controller Under Test in order to create a new namespace. Configure the NVM Set Identifier (NVMSETID) to be set to the non-zero value not associated with any NVM Set from step 2 and Endurance Group Identifier (ENDGID) to be set to the valid Endurance Group in step 2 in the Namespace Management – Host Software Specified Fields data structure for the creation

Observable Results:

1. Verify that the controller shall abort the Namespace Management command with a status of 'Invalid Field in Command' (02h).

Case 10: Creation of Namespace, LBAFEE=0h & ELBAS=1h (OF-FYI, FYI)

Test Procedure:

© 2026 UNH–IOL

1. Perform an Identify Command to CNS=01h. Verify the product under test supports Namespace Management and ELBAS isn't cleared to 0h otherwise this test is not applicable.
2. Configure the NVMe Host to issue a Get Feature for FID 16h. If FID 16h is not supported or LBAFEE in the Host Behavior Support – Data Structure is not set to 0h this test is not applicable.
3. Configure the NVMe Host to create a namespace with an extended protection information format that the controller supports (16b Guard Protection with non-zero STS, 32b Guard Protection or 64b Guard Protection).

Observable Results:

1. Verify that the Namespace Management command aborts with the status code of Invalid Namespace or Format.

Possible Problems: If no currently created namespaces exist that support extended protection, this test may require deleting and creating a namespace that does.

© 2026 UNH–IOL

Test 8.3 – Namespace Attachment Command (PCIeC 9.3) (OF-FYI, M)

Purpose: To determine if an NVMe Controller properly implements the Namespace Attachment command.

References:

NVMe Specification 6.1, 8.12

Resource Requirements:

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

Last Modification: March 2, 2016

Discussion: The Namespace Attachment command is used to attach and detach controllers from a namespace.

The Select field of command Dword 10 of the Namespace Attachment command is used to specify the operation of the command (attach or detach). The data structure used in the command is a 4096 byte Controller List which specifies the controllers that are to be attached or detached as part of the command. If the Controller List data structure transferred with the Namespace Attachment command is not properly formatted or is otherwise invalid then the command shall return status Controller List Invalid.

The Namespace Attachment command uses the Command Dword 10 field. All other command specific fields are reserved.

Test Setup: See Appendix A.

Case 1: Namespace Attachment (OF-FYI, M)

If a Namespace Attachment command is issued with the Controller Attach action selected and the namespace is already attached to a controller specified in the Controller List data structure transferred with the command, then the command shall return status Namespace Already Attached.

If a Namespace Attachment command is issued with the Controller Attach action selected and the namespace is private and already attached to another controller, then the namespace shall not be attached to any of the controllers specified in the Controller List data structure transferred with the command and the command shall return status Namespace Is Private.

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management then this test is not applicable.
2. Configure the NVMe Host to issue a Namespace Attachment command, specifying Select field value 0h (Attach), CDW0.NSID field value of an inactive namespace, and the controller identifier of the NVMe Controller Under Test in the attached data structure, to the controller in order to attach the specified namespace to the controller.
3. Configure the NVMe Host to issue an Identify command specifying CNS field value 02h and CDW1.NSID value 00h to the NVMe Controller Under Test in order to receive back a Namespace List containing active namespace IDs attached to the controller.
4. Configure the NVMe Host to issue a Namespace Attachment command, specifying Select field value 0h (Attach), CDW0.NSID field value of the same namespace used in the previous step, and the controller identifier of the NVMe Controller in the attached data structure, to the controller.

Observable Results:

1. Verify that after the completion of the command, the controller posts a completion queue entry to the appropriate Completion Queue indicating the status for the command.
2. Verify that the namespace attached to the controller via the Namespace Attachment command is contained within the returned Namespace List to signify that the namespace was successfully attached.

© 2026 UNH–IOL

3. Verify that the status of the second Namespace Attachment command is Namespace Already Attached.

Case 2: Namespace Detachment (OF-FYI, M)

When a namespace is detached from a controller it becomes an inactive namespace on that controller. Previously submitted but uncompleted or subsequently submitted commands to the affected namespace are handled by the controller as if they were issued to an inactive namespace.

If a Namespace Attachment command is issued with the Controller Detach action selected and the namespace is not attached to a controller specified in the Controller List data structure transferred with the command, then the command shall return status Namespace Not Attached.

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management then this test is not applicable.
2. Configure the NVMe Host to issue a Namespace Attachment command, specifying Select field value 1h (Detach), CDW0.NSID field value of an inactive namespace, and the controller identifier the NVMe Controller Under Test in the attached data structure, to the controller in order to attempt to detach the specified namespace from the controller.
3. Configure the NVMe Host to issue a Namespace Attachment command, specifying Select field value 1h (Detach), CDW0.NSID field value of a namespace attached to the controller, and the controller identifier of the NVMe Controller Under Test in the attached data structure, to the controller in order to detach the specified namespace from the controller.
4. Configure the NVMe Host to issue an Identify command specifying CNS field value 02h and CDW1.NSID value 00h to the NVMe Controller Under Test in order to receive back a Namespace List containing active namespace IDs attached to the controller.

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 status of the first Namespace Attachment command is Namespace Not Attached.
3. Verify that the namespace identifier of the namespace which was detached from the controller via the Namespace Attachment command is not contained within the returned Namespace List to signify that the namespace was successfully detached from the NVMe Controller Under Test.

Case 3: Namespace Attachment, Incorrect Command Set (OF-FYI, FYI)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management then this test is not applicable.
2. Check Bit 43 of the CAP.CSS field. If this bit is set to 0 this test is not applicable.
3. Configure the NVMe Host to issue a Namespace Attachment command, specifying Select field value 0h (Attach), CDW0.NSID field value of an active namespace, to a controller that does not support the I/O Command Set for the Namespace being attached. Ensure that the command set is enabled on the controller via the I/O Command Set Profile feature.

Observable Results:

1. Verify that that the Namespace Attach command completes with a status of I/O Command Set Not Supported.

Case 4: Namespace Attachment, Command Set not Enabled (OF-FYI, FYI)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
2. Check Bit 43 of the CAP.CSS field. If this bit is set to 0 this test is not applicable.

© 2026 UNH–IOL

3. Configure the NVMe Host to issue an Identify Command specifying CNS 1Ch (I/O Command Set data structure) to get the Identify I/O Command Set Data Structure. If there is only one I/O Command Set Combination, this test is not applicable. Determine all I/O Command Sets supported by the controller.
4. Configure the NVMe Host to issue a Namespace Attachment command with SEL=1h to detach all namespaces from the controller.
5. Configure the NVMe Host to issue a Set Feature command specifying FID 19h (I/O Command Set Profile) to select a profile index that has at least one I/O Command Sets disabled (i.e. I/O Command Set Vector bit for a supported command set is cleared to 0).
6. If there are no namespaces present that have the command set that is supported but not enabled in the I/O Command Set Profile, configure the NVMe Host to issue a Namespace Management command with SEL=0h (Create) and the CSI set to the value of the supported but disabled CSI.
7. Configure the NVMe Host to issue a Namespace Attachment command, specifying Select field value 0h(Attach), CDW0.NSID field value of a namespace which is configured to have a CSI of a supported but disabled I/O Command Set, to a controller that supports the I/O Command Set for the Namespace being attached, but the Command Set is not enabled via the I/O Command Set profile feature

Observable Results:

1. Verify that that the Namespace Attach command completes with a status of I/O Command Set Not Enabled.
2. Verify that all other commands completed successfully.

Case 5: MAXDNA Namespace Attachment Limit Exceeded (OF-FYI, FYI)

Test Procedure:

1. If the DUT does not have multiple controllers within a single domain this test is not applicable.
2. Check the MAXDNA field. If MAXDNA is set to 0h this test is not applicable.
3. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
4. Configure the NVMe Host to issue a Namespace Attachment command, specifying Select field value 0h (Attach), CDW0.NSID field value of an active namespace, in order to attach the specified namespace to the Domain.
5. Repeat the previous step, attaching Namespaces until the MAXDNA limit is exceeded.

Observable Results:

1. Verify that the final Namespace Attach command completes with Status Namespace Attachment Limit Exceeded.

Case 6: MAXCNA Namespace Attachment Limit Exceeded (OF-FYI, FYI)

Test Procedure:

1. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, then this test is not applicable.
2. Check the MAXCNA field. If MAXCNA is set to 0 this test is not applicable.
3. Configure the NVMe Host to issue a Namespace Attachment command, specifying Select field value 0h (Attach), CDW0.NSID field value of an active namespace, in order to attach the specified namespace to the controller.
4. Repeat the previous step, attaching Namespaces until the MAXCNA limit is exceeded.

Observable Results:

1. Verify that the final Namespace Attach command completes with Status Namespace Attachment Limit Exceeded.

Possible Problems:

There may not be an inactive namespace in the NVM subsystem. In this case, the NVMe Host may create an inactive namespace in order to perform procedure steps which require such a namespace.

© 2026 UNH–IOL

Test 8.4 – Dispersed Namespaces (OF-FYI, FYI)

Purpose: To determine if an NVMe Controller properly implements the Dispersed Namespaces command.

References:

NVM Express Base Specification 2.0 : 8.10

Resource Requirements:

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

Last Modification: January 04, 2023

Discussion: The Dispersed Namespaces command provides hosts with access to the same namespace using multiple participating NVM Subsystem.

This command is read only and cannot be modified by standard methods from the NVMe Technical Specifications of NVMeExpress.org

Test Setup: See Appendix A.

Case 1: Get Log LID 17h, Invalid namespaces (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a get log page to LID 0h, if LID 17h is not reported as supported this test is not applicable.
2. Configure the NVMe Host to issue a get log page to LID 17h and NSID FFFFFFFFh.
3. Configure the NVMe Host to issue a get log page to LID 17h and NSID 0h.

Observable Results:

1. Verify both get log pages to LID 17h abort with Invalid Namespace or Format.

Case 2: Get Log LID 17h, Non-Dispersed Namespaces (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a get log page to LID 0h, if LID 17h is not reported as supported this test is not applicable.
2. For each active namespace, issue an Identify Namespace with CNs 08h, record the DISNS field in NMIC.
3. Configure the NVMe Host to issue a get log page to LID 17h to a NSID that is associated with a namespace that has DISNS cleared to 0h.

Observable Results:

1. Verify the Get Log Page sent in step 2 is abort with Invalid Field in Command.

Possible Problem: If no currently active namespaces have DISNS cleared to 0h, this test may require deleting and/or creating a namespace that is a non-dispersed namespace.

Case 3: Get Log LID 17h NQN matches ID CTRL (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a get log page to LID 0h, if LID 17h is not reported as supported this test is not applicable.
2. Configure the NVMe Host to issue a get feature to FID 16h, if HDISNS is cleared to 0h this test is not applicable.
3. For each active namespace, issue an Identify Namespace with CNs 08h, record the DISNS field in NMIC.

© 2026 UNH–IOL

4. Configure the NVMe Host to issue a get log to LID 17h to a NSID that is associated with a namespace that has DISNS set to 1h, if none exist this test is not applicable.
5. Configure the NVMe Host to issue an Identify Controller command.

Observable Results:

1. Verify the Participating NVM Subsystem Entry 0 field from the Get Log Page in step 3 contains the NQN of the NVM subsystem containing the controller processing the command.

Case 4: DISNSRS 0h sent to Dispersed Namespace (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify Controller command, then check bit 5 of the ONCS field to determine if the controller supports reservations. If the controller does not support reservations, then this test is not applicable.
2. Configure the NVMe Host to issue a get feature to FID 16h, if FID 16h is not supported or HDISNS is cleared to 0h this test is not applicable.
3. Configure the NVMe Host to issue a valid Reservation Register Command with DISNSRS cleared to 0h to a dispersed namespace, if no dispersed namespaces exist this test is not applicable.

Observable Results:

1. Verify the reservation register command aborts with status code of Namespace Is Dispersed

© 2026 UNH–IOL

Group 9: Flexible Data Placement

Overview:
This section describes a method for performing conformance verification for NVMe products implementing NVMe Flexible Data Placement features.

Notes:
The preliminary draft descriptions for the Flexible Data Placement (FDP) tests defined in this group are considered complete, and the tests are Implemented. FDP allows the host to reduce write amplification by aligning user data usage with the physical media.

Purpose: To determine if an NVMe Controller properly implements the Flexible Data Placement feature capabilities correctly.

References: NVM Express Base Specification 2.0d : 8.13

Resource Requirements:
- Tools capable of monitoring and decoding traffic on the NVMe interface.

Last Modification: July 12, 2023

Discussion: Here are a few areas worth noting for this optional capability feature:
- Firstly, the feature must be supported by the device.
- The scope of the feature is the ‘endurance group’, the user must choose an endurance group where the feature will be enabled and then delete all existing namespaces on that chosen endurance group.
- A Get Log command specifying FDP Configurations is sent by the host and returns a list of configurations to be used for the feature.
- With the chosen configuration and endurance group in mind, a Set Feature specifying Flexible Data Placement is sent with the configuration index, and the Flexible Data Placement Enable (FDPE) bit set to ‘1h’.
- The next several steps deal with namespace creation.
    - A Get Feature command specifying FDP Events is sent to acquire a list of supported and enabled FDP events.
    - The Get Feature command is then followed by a Set Feature command specifying FDP Events to select which events are to be enabled and required by the Reclaim Unit Handle(s).
    - An Identify Namespace command specifying the Identify Namespace Data Structure (CNS 00h) is sent to determine which LBA formats are supported.
    - Then, for each namespace to be created in the endurance group, a Namespace Management Command set to ‘Create’ is sent, specifying the User Data Format and the Placement Handle List.
    - Lastly, a Directive Send command specifying the Data Placement Directive is sent for each created namespace where the Data Placement Directive is to be written to.

Test Setup: See Appendix A.

© 2026 UNH–IOL

Test 9.1 – FDP Configuration Log Page, Log Identifier 20h (OF-FYI, FYI)

Case 1: FDP Configuration Log Page, Version (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Log Page command for the Flexible Data Placement Configuration (FDP) Log Page (LID 20h).

Observable Results:

1. Verify the Get Log Page command executed successfully.
2. Verify the Version field of the Flexible Data Placement Configuration Log Page is cleared to 0.

Case 2: FDP Configuration Log Page, FDPVWC=1 (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.
2. Perform a Get Log Page command for the Flexible Data Placement Configuration (FDP) Log Page (LID 20h).
3. For each FDP Configuration Descriptor, in the FDP Configuration Log Page record FDP Volatile Write Cache Present (FDPVWC). If FDPVWC is cleared to 0h for all Configuration Descriptors, this test is not applicable.
4. Perform an Identify Controller command (CNS 01h). Record the Volatile Write Cache Present (VWP) field.

Observable Results:

1. Verify both log page commands executed successfully.
2. Verify that if FDPVWC is set to 1h within any of the FDP Configuration Descriptors, then bit 0 of VWP, is also set to 1h.

Case 3: FDP Configuration Log Page, NRG>1 & RGIF not 0 (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.
2. Perform a Get Log Page command for the Flexible Data Placement Configuration (FDP) Log Page (LID 20h).
3. For each FDP Configuration Descriptor, in the FDP Configuration Log Page, record the Reclaim Group Identifier Format (RGIF) and the Number of Reclaim Groups (NRG).

Observable Results:

1. Verify the Get Log Page command executed successfully.
2. Verify that each RGIF from step 3 is set to a non-zero value, if its corresponding NRG is greater than 1.

Case 4: FDP Configuration Log Page, NRG not 0 (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Log Page command for the Flexible Data Placement Configuration (FDP) Log Page (LID 20h).
3. For each FDP Configuration Descriptor, in the FDP Configuration Log Page record the Number of Reclaim Groups (NRG).

Observable Results:

1. Verify the Get Log Page command executed successfully.
2. Verify that all NRG fields are non-zero.

© 2026 UNH–IOL

Case 5: FDP Configuration Log Page, NRUH not 0 (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.
2. Perform a Get Log Page command for the Flexible Data Placement Configuration (FDP) Log Page (LID 20h).
3. For each FDP Configuration Descriptor, in the FDP Configuration Log Page record Number of Reclaim Group Handles (NRUH).

Observable Results:

1. Verify the Get Log Page command executed successfully.
2. Verify that all NRUH are non-zero.

Case 6: FDP Configuration Log Page, MAXPIDS<NRG*NRUH (OF-FYI, FYI) {#case-9-1-6 }

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Log Page command for the Flexible Data Placement Configuration (FDP) Log Page (LID 20h).
3. For each FDP Configuration Descriptor, in the FDP Configuration Log Page record:
    1. Number of Reclaim Groups (NRG)
    2. Number of Reclaim Group Handles (NRUH)
    3. Max Placement Identifiers (MAXPIDS)

Observable Results:

1. Verify the Get Log Page command executed successfully.
2. Verify that for all FDP Configuration Descriptors MAXPIDS is less than the product of the NRG field and the NRUH field.

Case 7: FDP Configuration Log Page, Number of Namespaces Supported (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.
2. Perform a Get Log Page command for the Flexible Data Placement Configuration (FDP) Log Page (LID 20h).
3. For each FDP Configuration Descriptor, in the FDP Configuration Log Page record the Number of Namespaces Supported.
4. Perform an Identify Controller command (CNS 01h), record Maximum Number of Allowed Namespaces (MNAN). If MNAN is equal to zero record Number of Namespace Supported (NN).

Observable Results:

1. Verify all commands executed successfully.
2. Verify that for all FDP Configuration Descriptors, Number Of Namespaces Supported recorded in step 3, is less than or equal to the value determined in step 4.

Case 8: FDP Configuration Log Page, 8-byte Boundary Cleared to 0 (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Log Page command for the Flexible Data Placement Configuration (FDP) Log Page (LID 20h).
3. For each FDP Configuration Descriptor, in the FDP Configuration Log Page record:
    1. (VSS)
    2. (NRUH)

© 2026 UNH–IOL

Observable Results:

1. Verify the Get Log Page command executed successfully.
2. Verify that all for FDP Configuration Descriptors, the 8 bytes of padding starting at byte: (NRUH*4)+64+VSS is cleared to 0h.

© 2026 UNH–IOL

Test 9.2 – Reclaim Unit Handle Usage Log Page, Log Identifier 21h (OF-FYI, FYI)

Case 1: Reclaim Unit Handle Usage Log Page, Invalid NSID (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
3. Perform an Identify command for CNS 10h, Allocated Namespace ID list. Select a NSID value not contained in the returned list.
4. Perform a Get Log Page command for the Reclaim Unit Handle Usage log page (LID 21h) specifying the invalid NSID based on step 3.

Observable Results:

1. Verify the Get Log Page command completed with a status code of 'Invalid Field in Command' (02h).

Case 2: Reclaim Unit Handle Usage Log Page, FDP Disabled (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Feature command for Flexible Data Placement (FID 1Dh), specifying any endurance group, to determine if FDP is disabled. If Flexible Data Placement Enable (FDPE) is set to 1. Configure the DUT to disable FDP for that endurance group.
3. Perform a Get Log Page command for the Reclaim Unit Handle Usage log page (LID 21h).

Observable Results:

1. Verify the Get Feature command executed successfully.
2. Verify the controller rejected the Get Log Page command with status: "FDP Disabled".

Case 3: Reclaim Unit Handle Usage Log Page, NRUH (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Log Page command for the FDP Configuration Log Page (LID 20h), select any configuration index. For that configuration record the Number of Reclaim Units (NRUH) field.
3. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.
4. Perform a Get Log Page command for the Reclaim Unit Handle Usage log page (LID 21h), specifying the same endurance group. Record the NRUH field.

Observable Results:

1. Verify that both Log Page commands executed successfully, and that both NRUH fields recorded are identical.

Case 4: Reclaim Unit Handle Usage Log Page, Reclaim Unit Handle Attributes (OF-FYI, FYI)

Test Procedure:

© 2026 UNH–IOL

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.
3. Perform a Get Log Page command for the Reclaim Unit Handle Usage log page (LID 21h), specifying the same endurance group. For each Reclaim Unit Handle Usage Descriptor, record the Reclaim Unit Handle Attributes.

Observable Results:

1. Verify the Get Log Page command executed successfully.
2. Verify that there is no more than one Reclaim Unit Handle Attributes field set to 2h.

© 2026 UNH–IOL

Test 9.3 – FDP Statistics Log Page, Log Identifier 22h (OF-FYI, FYI)

Case 1: FDP Statistics Log Page, FDP Disabled (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.
2. Perform a Get Feature command for Flexible Data Placement (FID 1Dh), specifying any endurance group, to determine if FDP is disabled. If Flexible Data Placement Enable (FDPE) is set to 1h. Configure the DUT to disable FDP for that endurance group.
3. Perform a Get Log Page command for the Flexible Data Placement (FDP) Statistics Log Page (LID 22h).

Observable Results:

1. Verify the Get Feature executed successfully.
2. Verify the controller rejected the Get Log Page command with status: 'FDP Disabled' (29h).

Case 2: FDP Statistics Log Page, Invalid NSID (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.
3. Perform an Identify command for CNS 10h, Allocated Namespace ID list. Select a NSID value not contained in the returned list.
4. Perform a Get Log Page command for the Flexible Data Placement (FDP) Statistics Log Page (LID 22h) specifying the invalid NSID based on step 3.

Observable Results:

1. Verify the Get Log Page command completed with a status code of Invalid Field in Command.

Case 3: FDP Statistics Log Page, Statistics Reset to 0h (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.
3. Perform 5 Write commands to the controller of a known data length to a namespace within the endurance group.
4. Perform a Sanitize command for any supported action.
5. Perform a Get Log Page command for the Flexible Data Placement (FDP) Statistics Log Page (LID 22h). Record:
    1. Host Bytes with Metadata Written (HBMW)
    2. Media Bytes with Metadata Written (MBMW)
    3. Media Bytes Erased (MBE)
6. Perform a Set Feature command with FDP=1Dh and FDPE=0h to disable FDP, then perform a second Set Feature with FDP=1Dh and FDPE=1h to re-enable FDP.
7. Perform a second Get Log Page command for the FDP Statistics Log Page (LID 22h). Record the same fields.

Observable Results:

1. Verify the Get Feature executed successfully.

© 2026 UNH–IOL

2. Verify the first Log Page returns non-zero values for all recorded fields.
3. Verify the second Log Page returns all recorded fields cleared to 0h.

Case 4: FDP Statistics Log Page, HBMW (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.
3. Perform a Get Log Page command for the Flexible Data Placement (FDP) Statistics Log Page (LID 22h). Record Host Bytes with Metadata Written (HBMW).
4. Perform 5 Write commands to the controller of a known data length to a namespace within the endurance group.
5. Perform a second Get Log Page command for the FDP Statistics log page (LID 22h). Record the same fields.

Observable Results:

1. Verify the Get Feature executed successfully.
2. Verify that HBMW is incremented by the amount of data written.

Case 5: FDP Statistics Log Page, MBE (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Check the SANICAP field in the Identify Controller Data Structure. If SANICAP is not set to 1, this test is not applicable.
3. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.
4. Perform a Write command to the controller with a data pattern of FFs to a namespace within the endurance group.
5. Perform a Get Log Page command for the Flexible Data Placement (FDP) Statistics log page (LID 22h). Record Media Bytes Erased (MBE).
6. Perform a Sanitize command for any supported action.
7. Perform a second Get Log Page command for the FDP Statistics log page (LID 22h). Record the same fields.

Observable Results:

1. Verify the Get Feature executed successfully.
2. Verify that MBE is incremented.

© 2026 UNH–IOL

Test 9.4 – FDP Events Log Page, Log Identifier 23h (OF-FYI, FYI)

Case 1: FDP Events Log Page, Events Disabled (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
3. Perform a Get Feature command for Flexible Data Placement Events (FID 1Eh) to determine the list of supported FDP Events. If there are no supported host FDP Events this test is not applicable.
4. Generate a supported, but not enabled FDP Host Event.
5. Perform a Get Log Page command for the Flexible Data Placement (FDP) Events Log Page (LID 23h).

Observable Results:

1. Verify the Get Feature executed successfully.
2. Verify the first Log Page contains no FDP Event entries and Number of Events (N) is cleared to 0h.

Case 2: FDP Events Log Page, Invalid NSID (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.
3. Perform an Identify command for CNS 10h, Allocated Namespace ID list. Select a NSID value not contained in the returned list.
4. Perform a Get Log Page command for the Flexible Data Placement (FDP) log page (LID 23h) specifying the invalid NSID based on step 3.

Observable Results:

1. Verify the Get Log Page command completed with a status code of Invalid Field in Command.

Case 3: FDP Events Log Page, FDP Disabled (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.
2. Perform a Get Feature command for Flexible Data Placement (FID 1Dh), specifying any endurance group, to determine if FDP is disabled. If Flexible Data Placement Enable (FDPE) is set to 1h. Configure the DUT to disable FDP for that endurance group.
3. Perform a Get Log Page command for the Flexible Data Placement (FDP) Events Log Page (LID 23h).

Observable Results:

1. Verify the controller rejected the Get Log Page command with status: ‘FDP Disabled’ (29h).
2. Verify the Get Feature executed successfully.

Case 4: FDP Events Log Page, N=0h (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.

© 2026 UNH–IOL

2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.
3. Perform a Get Feature command for Flexible Data Placement Events (FID 1Eh) to determine the list of supported FDP Events. If there are no supported host FDP Events this test is not applicable.
4. Perform a Set Feature command for Flexible Data Placement Events (FID 1Eh) specifying a single FDP Event Type to enable.
5. Fulfill the conditions required to generate the FDP Event enabled in step 4.
6. Perform a Get Log Page command for the FDP Events Log Page (LID 23h) with FDP Event Type (FDPET) set to 1h.
7. Configure the DUT to disable FDP for the endurance group. Then re-enable FDP.
8. Perform a Get Log Page command for the FDP Events Log Page (LID 23h) with FDP Event Type (FDPET) set to 1h.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that in the first Log Page the Number of Events (N) is set to 1h.
3. Verify that in the second Log Page N is cleared to 0h.

Case 5: FDP Events Log Page, N Increments (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
3. Perform a Get Log Page command for the Flexible Data Placement (FDP) Events Log Page (LID 23h).
4. Perform a Get Feature command for Flexible Data Placement Events (FID 1Eh) to determine the list of supported FDP Events. If there are no supported host FDP Events this test is not applicable.
5. Perform a Set Feature command for FID 1Eh specifying a single FDP Event Type to enable.
6. Generate the FDP Event enabled in step 4.
7. Perform a Get Log Page command for the FDP Events Log Page (LID 23h) with FDP Event Type (FDPET) set to 1h.
8. Generate a second FDP Event of the same type as the type generated in step 6.
9. Perform a Get Log Page command for the FDP Events Log Page (LID 23h) with FDP Event Type (FDPET) set to 1h.

Observable Results:

1. Verify all commands completed successfully.
2. Verify the first Log Page contains no FDP Event entries and Number of Events (N) is cleared to 0h.
3. Verify the second Log Page contains a single FDP Event entry, all fields are correctly formatted, and N is set to 1h.
4. Verify the third Log Page contains two FDP Event entries, all fields are correctly formatted for each, and N is set to 2h.

Case 6: FDP Events Log Page, Invalid Placement Identifier Event (03h) (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.

© 2026 UNH–IOL

3. For any namespace perform a Directive Send command specifying:
    a. The Directive Operation (DO) field set to Enable Directive (1h)
    b. The NSID of that namespace
    c. The DTYPE field in Command Dword 11 set to Identify (00h)
    d. The DTYPE field in Command Dword 12 set to Data Placement (02h)
    e. The Enable Directive (ENDIR) bit set to 1h.
4. Perform a Get Feature command for Flexible Data Placement Events (FID 1Eh) to determine the list of supported FDP Events. If the Invalid Placement Identifier (Event Type 3h) is not supported this test is not applicable.
5. Perform a Set Feature command for FID 1Eh specifying FDP Event Type 3h to be enabled.
6. Perform a write command with DTYPE set to 2h and DSPEC set to an invalid placement identifier.
7. Perform a Get Log Page command for the FDP Events Log Page (LID 23h) with FDP Event Type (FDPET) set to 1h.

Observable Results:

1. Verify all commands completed successfully.
2. Verify the Log Page contains a single FDP Event entry, and for that entry ensure:
    a. Event Type is set to 3h,
    b. FDP Event Flags (FDPEF) field is set to 11b,
    c. Placement Identifier (PID) is set to the value given for DSPEC in step 6, and
    d. Namespace Identifier (NSID) is set to the NSID written to in step 6.

Case 7: FDP Events Log Page, Order of Occurrence (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
3. Perform a Get Feature command for Flexible Data Placement Events (FID 1Eh) to determine the list of supported FDP Events. If there are no supported host FDP Events this test is not applicable.
4. Perform a Set Feature command for FID 1Eh specifying a single FDP Event Type to enable.
5. Generate the FDP Event enabled in step 4.
6. Perform a Get Log Page command for the FDP Events Log Page (LID 23h) with FDP Event Type (FDPET) set to 1h.
7. Generate a second FDP Event of the same type as the type generated in step 5.
8. Perform a Get Log Page command for the FDP Events Log Page (LID 23h) with FDP Event Type (FDPET) set to 1h.

Observable Results:

1. Verify all commands completed successfully.
2. Verify the first Log Page contains a single FDP Event entry, all fields are correctly formatted, and Number of Events (N) is set to 1h.
3. Verify the second Log Page contains two FDP Event entries, all fields are correctly formatted for each, and the events are listed in order of occurrence.

Case 8: FDP Events Log Page, events Discarded (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.

© 2026 UNH–IOL

2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
3. Perform a Get Feature command for Flexible Data Placement Events (FID 1Eh) to determine the list of supported FDP Events. If there are no supported host FDP Events this test is not applicable.
4. Perform a Set Feature command for FID 1Eh specifying a single FDP Event Type to enable.
5. Generate FDP Events of the type enabled in step 4 until Number of Events (N) has reached the maximum value.
6. Perform a Get Log Page command for the FDP Events Log Page (LID 23h) with FDP Event Type (FDPET) set to 1h.
7. Generate an additional FDP Event of the same type as the type generated in step 6.
8. Perform a Get Log Page command for the FDP Events Log Page (LID 23h) with FDP Event Type (FDPET) set to 1h.
9. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.
10. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
11. Perform an Identify command for CNS 10h, Allocated Namespace ID list. Select a NSID value not contained in the returned list.
12. Perform a Get Log Page command for the Flexible Data Placement (FDP) log page (LID 23h) specifying the invalid NSID based on step 3.

Observable Results:

1. Verify all commands are completed successfully.
2. Verify that the Log Page command in step 6 returns a list filled with FDP Events and the Number of FDP Events (N) is equal to the maximum value.
3. Verify that the Log Page command in step 8 has discarded the oldest event, the event generated in step 7 is put at the end of the list, and N is equal to the maximum value.

Case 9: FDP Events Log Page, FDPET (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
3. Perform a Get Feature command for Flexible Data Placement Events (FID 1Eh) to determine the list of supported FDP Events. If there are no supported host FDP Events this test is not applicable.
4. Perform a Set Feature command for FID 1Eh specifying a single Host FDP Event Type to enable.
5. Generate a FDP Event of the same type as the type generated in step 4.
6. Perform a Get Log Page command for the FDP Events Log Page (LID 23h) with FDP Event Type (FDPET) set to 0h.
7. Perform a Get Log Page command for the FDP Events Log Page (LID 23h) with FDP Event Type (FDPET) set to 1h.

Observable Results:

1. Verify all commands are completed successfully.
2. Verify the first Log Page contains no FDP Event entries and Number of Events (N) is cleared to 0h.
3. Verify the second Log Page contains a single FDP Event entry, all fields are correctly formatted, and Number of Events (N) is set to 1h.

© 2026 UNH–IOL

Test 9.5 – FDP Identify Controller, Log CNS 01h (OF-FYI, FYI)

Case 1: FDP Identify Controller, FDPS & FCM (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify command for CNS 01h, Identify Controller. Record the CTRATT field. If Fixed Capacity Management (FCM) and Flexible Data Placement Support (FDPS) are both cleared to 0h, then this test is applicable.

Observable Results:

1. Verify that one of these conditions are fulfilled by the DUT:
    a. If FCM is set to 1h, then FDPS is cleared to 0h; or
    b. If FDPS is set to 1h, then FCM is cleared to 0h.

© 2026 UNH–IOL

Test 9.6 – FDP Namespace Management Operations (OF-FYI, FYI)

Case 1: FDP Namespace Creation, NPHNDLS>NRUH or 128 (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Log Page command for the Flexible Data Placement (FDP) Configuration Log Page (LID 20h), select any configuration index. For that configuration, record the Number of Reclaim Unit Handles (NRUH).
3. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.configuration index collected.
4. Perform a Namespace Management command to create a namespace specifying a Number of Placement Handles (NPHNDLS) greater than the lessor of 128 or NRUH.

Observable Results:

1. Verify the controller aborts the Namespace Management command with a status code: ‘Invalid Placement Handle List’.
2. Verify all other commands completed successfully.

Case 2: FDP Namespace Creation, NPHNDLS=NRUH or 128 (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Log Page command for the Flexible Data Placement (FDP) Configuration Log Page (LID 20h), select any configuration index. For that configuration, record the Number of Reclaim Unit Handles (NRUH).
3. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature configuration index collected.
4. Perform a Namespace Management command to create a namespace specifying a Number of Placement Handles (NPHNDLS) equal to the lessor of 128 or NRUH, specifying a valid Placement Handle List.
5. Perform a Namespace Management command to attach the newly created namespace to any controller.
6. Perform an I/O Management Receive command specifying the newly created namespace and Management Operation 01h, Reclaim Unit Handle Status. Record the Reclaim Unit Handle Status Descriptor List for the namespace.

Observable Results:

1. Verify all commands completed successfully.
2. Verify the Placement Handle List passed into the Namespace Management command matches the Placement Identifiers reported in Reclaim Unit Handle Status Descriptor List.

Case 3: FDP Namespace Creation, NPHNDLS=0h (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.
3. Perform two Namespace Management commands to create two namespaces specifying Number of Placement Handles (NPHNDLS) cleared to 0h.
4. Perform two Namespace Management commands to attach the newly created namespaces to any controller.

© 2026 UNH–IOL

5. Perform two I/O Management Receive commands specifying the newly created namespaces and Management Operation 01h, Reclaim Unit Handle Status. Record the Reclaim Unit Handle Status Descriptor List for both namespaces.

Observable Results:

1. Verify all commands completed successfully.
2. Verify both Reclaim Unit Handle Status Descriptor Lists only contain one entry, and both Reclaim Unit Handles are identical.

Case 4: FDP Namespace Creation, NPHNDLS=0h (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.
3. Perform two Namespace Management commands to create two namespaces specifying Number of Placement Handles (NPHNDLS) cleared to 0h.
4. Perform two Namespace Management commands to attach the newly created namespaces to any controller.
5. Perform two I/O Management Receive commands specifying the newly created namespaces and Management Operation 01h, Reclaim Unit Handle Status. Record the Reclaim Unit Handle Status Descriptor List for both namespaces.

Observable Results:

1. Verify a new namespace is created and attached to a controller.
2. Verify the controller rejected the Namespace Management command from step 5 with a status code: ‘Invalid Placement Handle List’.

Possible Problems: If NRUH is greater than 128, additional namespaces will be required to put the DUT in a state where all Reclaim Unit Handles are occupied. If MNAN (or NN if MNAN=0) is less than NRUH his test is not applicable.

Case 5: FDP Namespace Creation, Controller Specified RUH in Placement List (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.
3. Perform a Namespace Management command to create a namespace specifying Number of Placement Handles (NPHNDLS) to be of 0h.
4. Perform a Namespace Management command to attach the newly created namespace to any controller.
5. Perform an I/O Management Receive command specifying the newly created namespace and Management Operation 01h, Reclaim Unit Handle Status. Record the Reclaim Unit Handle Status Descriptor List for the namespace. Record the Reclaim Unit Handle assigned by the controller.
6. Perform a Namespace Management command to create a namespace specifying a NPHNDLS value of 1h, and a Placement Handle List containing the Reclaim Unit Handle collected from step 5.

Observable Results:

1. Verify all commands before step 6 completed successfully.
2. Verify the controller rejected the Namespace Management command from step 6 with a status code: ‘Invalid Placement Handle List’.

© 2026 UNH–IOL

Case 6: FDP Namespace Creation, Duplicate Placement Handles (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Log Page command for the Flexible Data Placement (FDP) Configuration Log Page (LID 20h), select any configuration index. For that configuration, record the Number of Reclaim Unit Handles (NRUH).
3. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature, specifying the configuration index collected in step 2.
4. Perform a Namespace Management command to create a namespace specifying a Number of Placement Handles (NPHNDLS) set to 2h, and a Placement Handle List containing two duplicate entries.

Observable Results:

1. Verify the controller rejected the Namespace Management command with a status code: ‘Invalid Placement Handle List’

Case 7: FDP Namespace Creation, Duplicate Placement Handle > NRUH (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Log Page command for the Flexible Data Placement (FDP) Configuration Log Page (LID 20h), select any configuration index. For that configuration, record the Number of Reclaim Unit Handles (NRUH).
3. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature, specifying the configuration index collected in step 2.
4. Perform a Namespace Management command to create a namespace specifying a Number of Placement Handles (NPHNDLS) set to 2h, and a Placement Handle List containing two duplicate entries.

Observable Results:

1. Verify the controller rejected the Namespace Management command with a status code: ‘Invalid Placement Handle List’

Possible Problems: If NRUH is set to the maximum value of FFFFh then a Placement Handle cannot exceed it, thus making this test not applicable.

Case 8: FDP Namespace Creation, Shared Placement Handle, Different FLBAS (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform an Identify command with CNS 00h, Identify Namespace, specifying a NSID of FFFFFFFFh. If the Number of LBA Formats (NLBAF) is less than 1h this test is not applicable.
3. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.
4. Perform a Namespace Management command to create a namespace specifying Number of Placement Handles (NPHNDLS) to be 0h and Formatted LBA Size (FLBAS) to be 0h.
5. Perform a Namespace Management command to create a namespace specifying NPHNDLS cleared to 0h and FLBAS set to 1h.

Observable Results:

© 2026 UNH–IOL

1. Verify a new namespace is created in step 4.
2. Verify the controller rejected the Namespace Management command from step 5 with a status code of ‘Invalid Format’.
3. Verify all other commands completed successfully.

© 2026 UNH–IOL

Test 9.7 – FDP Writes (OF-FYI, FYI)

Case 1: FDP Write Command FDPE=1h, DTYPE=2h, Invalid DSPEC (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.
3. Perform a Directive Send command specifying:
    a. The Directive Operation field set to Enable Directive (1h),
    b. The NSID of a namespace inside the FDP Enabled endurance group,
    c. The DTYPE field in Command Dword 11 set to Identify (00h),
    d. The DTYPE field in Command Dword 12 set to Data Placement (02h),
    e. The Enable Directive bit set to 1.
4. Perform a Get Feature command for Flexible Data Placement Events (FID 1Eh) to determine the list of supported FDP Events. If the Invalid Placement Identifier (Event Type 3h) is not supported skip steps 5, 8, and observable 3.
5. Perform a Set Feature command for FID 1Eh specifying FDP Event Type 3h to be enabled.
6. Perform a Write command to write a known data pattern to LBA 0 with DTYPE set to 2h and DSPEC set to an invalid placement handle.
7. Perform a Read command to read the data pattern written to LBA 0.
8. Perform a Get Log Page command for the Flexible Data Placement (FDP) Events Log Page (LID 23h).

Observable Results:

1. Verify all commands completed successfully.
2. Verify that the known data pattern was read correctly from the namespaces exactly as it was written.
3. Verify the Get Log Page command contains an Invalid Placement Event entry.

Case 2: FDP Write Command FDPE=1h, DTYPE=0h, and DSPEC=0h (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.
3. Perform a Write command to write a known data pattern to LBA 0 with DTYPE and DSPEC set to 0h.
4. Perform a Read command to read the data pattern written to LBA 0.

Observable Results:

1. Verify that all commands completed successfully.
2. Verify that the known data pattern was read correctly from the namespaces exactly as it was written.

Case 3: FDP Write Command FDPE=1h, DTYPE=2h, and Valide Placement Identifier (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.

© 2026 UNH–IOL

2. Perform a Get Log Page command for the FDP Configuration Log Page (LID 20h), select any configuration index. For that configuration, record Reclaim Group Identifier Format (RGIF) and Number of Reclaim Groups (NRG).
3. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature, specifying the configuration from step 2.
4. Perform a Directive Send command specifying:
    1. The Directive Operation field set to Enable Directive (1h),
    2. The NSID of a namespace inside the FDP Enabled endurance group,
    3. The DTYPE field in Command Dword 11 set to Identify (00h),
    4. The DTYPE field in Command Dword 12 set to Data Placement (02h),
    5. The Enable Directive bit set to 1.
5. Perform a Get Feature command for Flexible Data Placement Events (FID 1Eh) to determine the list of supported FDP Events. If the Invalid Placement Identifier (Event Type 3h) is not supported skip steps 5, 8, and observable 3.
6. Perform a Set Feature command for FID 1Eh specifying FDP Event Type 3h to be enabled.
7. Perform a Write command to write a known data pattern to LBA 0 with DTYPE set to 2h and DSPEC set to a valid Placement Identifier determined using the RGIF and NRG values recorded from the configuration log.
8. Perform a Read command to read the data pattern written to LBA 0.

Observable Results:

1. Verify that all commands completed successfully.
2. Verify that the known data pattern was read correctly from the namespaces exactly as it was written.

Case 4: FDP Write Command FDPE=1h, DTYPE=2h, and Invalid DSPEC (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. 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.
3. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.
4. Perform a Directive Send command specifying:
    1. The Directive Operation field set to Enable Directive (1h),
    2. The NSID of a namespace inside the FDP Enabled endurance group,
    3. The DTYPE field in Command Dword 11 set to Identify (00h),
    4. The DTYPE field in Command Dword 12 set to Data Placement (02h),
    5. The Enable Directive bit set to 1.
5. Perform a Get Feature command for Flexible Data Placement Events (FID 1Eh) to determine the list of supported FDP Events. If the Invalid Placement Identifier (Event Type 3h) is not supported skip steps 6, 11, and observable 3.
6. Perform a Set Feature command for FID 1Eh specifying FDP Event Type 3h to be enabled.
7. Perform a Write Uncorrectable command to a particular LBA of a namespace within the FDP enabled endurance group, with DTYPE set to 2h and DSPEC set to an invalid placement handle.
8. Perform a Read command for the LBA on which the Write Uncorrectable command was performed.
9. Perform a Write command for the LBA on which the Write Uncorrectable command was performed in order to clear the invalid logical block status.
10. Perform a second Read command for the LBA on which the Write Uncorrectable command was performed.
11. Perform a Get Log Page command for the Flexible Data Placement (FDP) Events Log Page (LID 23h).

Observable Results:

© 2026 UNH–IOL

1. Verify that the first Read command returns with status 'Unrecovered Read Error' (81h).
2. Verify all other commands completed successfully.
3. Verify the Get Log Page command contains an Invalid Placement Event entry.

Case 5: FDP Write Uncorrectable Command FDPE=1h, DTYPE=0h, and DSPEC=0h (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. 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
3. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.
4. Perform a Write Uncorrectable command to a particular LBA of a namespace within the FDP enabled endurance group with DTYPE and DSPEC cleared to 0h.
5. Perform a Read command for the LBA on which the Write Uncorrectable command was performed.
6. Perform a Write command for the LBA on which the Write Uncorrectable command was performed in order to clear the invalid logical block status.
7. Perform a second Read command for the LBA on which the Write Uncorrectable command was performed.

Observable Results:

1. Verify that the first Read command returns with status 'Unrecovered Read Error' (81h).
2. Verify all other commands completed successfully.

Case 6: FDP Write Uncorrectable Command FDPE=1h, DTYPE=2h, and valid DSPEC (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0, then this test is not applicable.
2. 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
3. Perform a Get Log Page command for the FDP Configuration Log Page (LID 20h), select any configuration index. For that configuration, record Reclaim Group Identifier Format (RGIF) and Number of Reclaim Groups (NRG).
4. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature specifying the configuration from step 2.
5. Perform a Directive Send command specifying:
    1. The Directive Operation field set to Enable Directive (1h),
    2. The NSID of a namespace inside the FDP Enabled endurance group,
    3. The DTYPE field in Command Dword 11 set to Identify (00h),
    4. The DTYPE field in Command Dword 12 set to Data Placement (02h),
    5. The Enable Directive bit set to 1
6. Perform a Write Uncorrectable command to a particular LBA of a namespace within the FDP enabled endurance group with DTYPE set to 2h and DSPEC set to a valid placement handle.
7. Perform a Read command for the LBA on which the Write Uncorrectable command was performed.
8. Perform a Write command for the LBA on which the Write Uncorrectable command was performed in order to clear the invalid logical block status.

© 2026 UNH–IOL

9. Perform a second Read command for the LBA on which the Write Uncorrectable command was performed.

Observable Results:

1. Verify that the first Read command returns with status 'Unrecovered Read Error' (81h).
2. Verify all other commands completed successfully.

Case 7: FDP Write Zeros Command FDPE=1h, DTYPE=2h, and Invalid DSPEC (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.
2. 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.
3. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
4. Perform a Directive Send command specifying:
    a. The Directive Operation field set to Enable Directive (1h),
    b. The NSID of a namespace inside the FDP Enabled endurance group,
    c. The DTYPE field in Command Dword 11 set to Identify (00h),
    d. The DTYPE field in Command Dword 12 set to Data Placement (02h),
    e. The Enable Directive bit set to 1h.
5. Perform a Get Feature command for Flexible Data Placement Events (FID 1Eh) to determine the list of supported FDP Events. If the Invalid Placement Identifier (Event Type 3h) is not supported skip steps 6, 10, and observable 3.
6. Perform a Set Feature command for FID 1Eh specifying FDP Event Type 3h to be enabled.
7. Perform a Write command to the controller specifying a particular LBA of a namespace within the FDP enabled endurance group. The written data should be either all 1's, or an alternating pattern of 1's and 0's.
8. Perform a Write Zeroes command to the controller specifying the same LBA as the previous step with DTYPE set to 2h and DSPEC set to an invalid placement handle.
9. Perform a Read command to the controller specifying the same LBA as the previous step.
10. Perform a Get Log Page command for the Flexible Data Placement (FDP) Events Log Page (LID 23h).

Observable Results:

1. Verify that all commands complete successfully.
2. Verify the Read command returns data of zeroes only.
3. Verify the Get Log Page command contains an Invalid Placement Event entry.

Case 8: FDP Write Zeros Command FDPE=1h, DTYPE=0h, and DSPEC=0h (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.
2. 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.
3. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
4. Perform a Write command to the controller specifying a particular LBA of a namespace within the FDP enabled endurance group. The written data should be either all 1's, or an alternating pattern of 1's and 0's.

© 2026 UNH–IOL

5. Perform a Write Zeroes command to the controller specifying the same LBA as the previous step with DTYPE and DSPEC cleared to 0h.
6. Perform a Read command to the controller specifying the same LBA as the previous step.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify the Read command returns data of zeroes only.

Case 9: FDP Write Zeros Command FDPE=1h, DTYPE=2h, and Valid DSPEC (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.
2. 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.
3. Perform a Get Log Page command for the FDP Configuration Log Page (LID 20h), select any configuration index. For that configuration, record Reclaim Group Identifier Format (RGIF) and Number of Reclaim Groups (NRG).
4. Perform a Directive Send command specifying:
    a. The Directive Operation field set to Enable Directive (1h),
    b. The NSID of a namespace inside the FDP Enabled endurance group,
    c. The DTYPE field in Command Dword 11 set to Identify (00h),
    d. The DTYPE field in Command Dword 12 set to Data Placement (02h),
    e. The Enable Directive bit is set to 1h.
5. Perform a Write command to the controller specifying a particular LBA of a namespace within the FDP enabled endurance group. The written data should be either all 1's, or an alternating pattern of 1's and 0's.
6. Perform a Write Zeros command to the controller specifying the same LBA as the previous step with DTYPE set to 2h and DSPEC set to a valid placement identifier.
7. Perform a Read command to the controller specifying the same LBA as the previous step.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify the Read command returns data of zeroes only.

© 2026 UNH–IOL

Test 9.8 – FDP NVM Format Operations (OF-FYI, FYI)

Case 1: FDP Format NVM – Shared Placement Handle, Unique FLBAS (OF-FYI, FYI)

Test Procedure:

1. Check Flexible Data Placement Support (Bit 19) of the CTRATT field of the Identify Controller Data Structure. If this bit is cleared to 0h, then this test is not applicable.
2. Perform an Identify command with CNS 00h, Identify Namespace, specifying a NSID of FFFFFFFFh. If the Number of Logical Block Formats (NLBAF) is 0h this test is not applicable.
3. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable). If FDPE is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
4. Perform two Namespace Management commands to create two namespaces specifying (NPHNDLS) to be 0h, specifying a (FLBAS) of 0.
5. Perform two Namespace Management commands to attach the newly created namespaces to any controller.
6. Perform a Format NVM command to the second namespace created specifying (FLBAS) to be 1h.

Observable Results:

1. Verify that all commands, up until step 6, complete successfully.
2. Verify the controller rejected the Format NVM command from step 6 with a status code of Invalid Format.

© 2026 UNH–IOL

Test 9.9 – I/O Management Receive Management Operation (1h) (OF-FYI, FYI)

Case 1: FDP I/O Management Receive (MO=1h) (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Command and then check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0h, this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
3. Perform a I/O Management Send command with the Management Operation (MO) field set to 1h to Update the Reclaim Unit Handle with a chosen Number of Placement Identifiers (NPID).
4. Perform an I/O Management Receive command with the Management Operation (MO) field set to 1h to read Reclaim Unit Handle Status Data Structure.

Observable Results:

1. Verify all commands completed successfully.
2. Verify the I/O Management Receive in step 4 returns the same amount of Placement Identifiers specified in step 3.

Case 2: FDP I/O Management Receive (MO=1h), FDP Disabled (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Command and then check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0h, this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is not cleared to 0h, then disable FDP.
3. Perform an I/O Management Receive command with the Management Operation (MO) field set to 1h to read Reclaim Unit Handle Status Data Structure.

Observable Results:

1. Verify all commands prior to step 3 complete successfully.
2. Verify the I/O Management Receive command aborts with Status Code ‘FDP Disabled’ (29h).

Case 3: FDP I/O Management Receive (MO=1h), Invalid Namespace (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Command and then check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0h, this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If this field is cleared to 0h, then enable the feature.
3. Perform an I/O Management Receive command with the Management Operation (MO) field set to 1h to read Reclaim Unit Handle Status Data Structure with an NSID of 0xFFFFFFFF.
4. Perform an I/O Management Receive command with the Management Operation (MO) field set to 1h to read Reclaim Unit Handle Status Data Structure with an NSID of 0x0.

Observable Results:

1. Verify that all commands, before step 3, complete successfully.
2. Verify both I/O Management Receive commands aborts with Status Code ‘Invalid Namespace or Format’ (0Bh).

Case 4: FDP I/O Management Receive (MO=1h), Extra Zeros (OF-FYI, FYI)

Test Procedure:

© 2026 UNH–IOL

1. Perform an Identify Command and then check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0h, this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is cleared to 0, then enable the feature.
3. Perform an I/O Management Receive command with the Management Operation (MO) field set to 1h to read the Reclaim Unit Handle Status and the NUMD field set to a size larger than the size of the Reclaim Handle Unit Status Data Structure.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that the data past the size of the Reclaim Unit Handle Status Data Structure are filled with zeroes.
3. Verify that the number of Reclaim Unit Handle Status Descriptors returned in the Reclaim Unit Handle Status Structure matches the value indicated in the Number of Reclaim Unit Handle Status Descriptors (NRUHSD) field.

© 2026 UNH–IOL

Test 9.10 – I/O Management Send Management Operation (1h) (OF-FYI, FYI)

Case 1: FDP I/O Management Send (MO=1h), FDP Disabled (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Command and then check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0h, this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is not cleared to 0h, then disable the feature.
3. Perform an I/O Management Send command with the Management Operation (MO) field set to 1h to update the Reclaim Unit Handle.

Observable Results:

1. Verify all commands prior to step 3 completed successfully.
2. Verify the I/O Management Send command aborts with Status Code ‘FDP Disabled’ (29h).

Case 2: FDP I/O Management Send (MO=1h), NPIC>MAXPIDS (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Command and then check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0h, this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
3. Perform an I/O Management Send command with the Management Operation (MO) field set to 1h to update the Reclaim Unit Handle, while setting the Number of Placement Identifiers (NPID) to a value greater than the Max Placement Identifiers (MAXPIDS) of the enabled FDP configuration.

Observable Results:

1. Verify all commands prior to step 3 completed successfully.
2. Verify the I/O Management Send command fails with Status Code ‘Invalid Field in Command’ (02h).

Case 3: FDP I/O Management Send (MO=1h), NPID>NRG*NRUH (OF-FYI, FYI) {#case-9-10-3 }

Test Procedure:

1. Perform an Identify Command and then check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0h, this test is not applicable.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
3. Perform an I/O Management Send command with the Management Operation (MO) field set to 1h to update the Reclaim Unit Handle, while setting the Number of Placement Identifiers (NPID) to a value greater than the product of the Number of Reclaim Groups (NRG) and the Number of Reclaim Unit Handles (NRUH).

Observable Results:

1. Verify that all commands prior to step 3 complete successfully.
2. Verify the I/O Management Send command fails with Status Code ‘Invalid Field in Command’ (02h).

© 2026 UNH–IOL

Test 9.11 – FDP Enable Flexible Data Feature (1Dh) (OF-FYI, FYI)

Case 1: FDP Feature (FID 1Dh) (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify Command and then check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0h, this test is not applicable.
2. Perform a Get Log with LID 20h (Flexible Data Placement Configuration Log) and select desired Configuration. Record the Number of Namespaces Field (NN) for the desired configuration.
3. Perform a Namespace Management command to delete all existing Namespaces in the chosen Endurance Group.
4. Perform a Set Feature with FID 1Dh (Flexible Data Placement) with the chosen configuration acquired from the previous step.
5. Perform an Identify command with CNS=00h (Identify Namespace) and check the Identify Namespace Data Structure for supported LBA formats.
6. Using the NN field recorded from step 2: If the NN field is greater than 1, create 2 namespaces in the following step. Otherwise, create 1 namespace for the following step.
7. For each Namespace to be created, perform a Namespace Management command specifying the Select Field (SEL) set to 0h (Create) and Host Software Specified Fields containing a Placement Handle List configured according to the selected configuration.
8. For each Namespace created in the chosen Endurance Group:
    a. Perform a Get Feature with FID 1Eh (Flexible Data Placement Events) and select all supported Events.
    b. Perform a Set Feature with FID 1Eh (Flexible Data Placement Events) with all the supported Event(s) acquired from the previous step.
    c. Perform a Directive Send command with the following:
        i. Directive Operation (DO) field set to ‘Enable Directive’ (01h)
        ii. The NSID of the current Namespace
        iii. The DTYPE field in CDW 11 set to ‘Identify’ (00h)
        iv. The DTYPE field in CDW 12 set to ‘Data Placement’ (02h)
        v. The Enable Directive (ENDIR) bit set to ‘1h’
    d. Perform an I/O Management Receive command to retrieve the configurations of the placement handles configured by the host.

Observable Results:

1. Verify all commands complete with Status Code ‘Successful Completion’ (00h).
2. Verify the Flexible Data Feature is enabled, and functions as expected.
3. Verify the I/O management commands complete with Status Code ‘Successful Completion’ (00h), and return values previously set by the host.

Case 2: FDP Feature (FID 1Dh), Existing Namespace (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify command with CNS=01h (Identify Controller) and check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0h, then skip this test.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
3. Perform a Set Feature specifying Flexible Data Placement (FID=1Dh) with a value that is different from the current value of the Feature.

Observable Results:

1. Verify that all commands prior to step 3 complete successfully.

© 2026 UNH–IOL

2. Verify that the Set Features command aborts with Status Code ‘Command Sequence Error’ (0Ch).

Case 3: FDP Feature (FID 1Dh), SV=0 (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify command with CNS=01h (Identify Controller) and check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0h, then skip this test.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
3. Perform a Set Feature specifying Flexible Data Placement (FID=1Dh) with Save (SV) bit cleared to 0h.

Observable Results:

1. Verify that all commands prior to step 3 complete successfully.
2. Verify the Set Features command aborts with Status Code ‘Invalid Field in Command’ (02h).

Case 4: FDP Enabled, Streams Directive & Directive Operation 01h (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify command with CNS=01h (Identify Controller) and check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0h, then skip this test.
2. Perform a Directive Receive Command and read the Return Parameters Data Structure to check if the Streams Directive is supported. If the Streams Directive is not supported, then this test is not applicable.
3. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is set to 1h, disable the feature, then re-enable the feature.
4. Perform a Get Log Page command to LID= 01h (Error Information Log).
5. For each Namespace in the chosen Endurance Group:
    a. Perform a Directive Send command specifying the Streams Directive in the DTYPE field and the Directive Operation field is set to 01h
    b. Perform a Get Log Page command to LID= 01h (Error Information Log)

Observable Results:

1. Verify all commands up to and including step 4 complete with Status Code ‘Successful Completion’ (00h).
2. Verify that the Directive Send command in step 5.a is aborted with the status code of ‘Invalid Field in Command’ (02h).
3. Verify that the Get Log Page in step 5.b completes successfully and if it contains a new entry that the new entry sets the PEL to indicate the Directive Operation field.

© 2026 UNH–IOL

Test 9.12 – FDP Events Feature Identifier (1Eh) (OF-FYI, FYI)

Case 1: FDP Events Feature – Events Enabled (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify command with CNS=01h (Identify Controller) and check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0h, then skip this test.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
3. Perform a Get Feature command for Flexible Data Placement Events (FID 1Eh) to determine the list of supported FDP Events. If there is no supported event, this test is not applicable.
4. Perform a Set Feature command specifying the Flexible Data Placement Event Feature (FID=1Eh), with Number of FDP Event Types (NOET) set to the number of supported events returned by the Get Features command, and the FDP Event Enable bit set to 1h.
5. Perform a Get Feature command specifying the Flexible Data Placement Event Feature (FID=1Eh).

Observable Results:

1. Verify all commands complete with Status Code ‘Successful Completion’ (00h).
2. Verify that both Get Feature commands return the Events Types for each event listed in the log and Number of Events.
3. Verify that all Supported FDP Event Descriptors returned by the Get Feature command in step 5, have FDP Event Enable (bit 0) set to 1h.

Case 2: FDP Events Feature – Invalid Namespace (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify command with CNS=01h (Identify Controller) and check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller data structure. If it is cleared to 0h, then skip this test.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
3. Perform a Set Feature specifying Flexible Data Placement Events (FID=1Eh) with a Namespace Identifier set to FFFFFFFFh.
4. Configure the NVMe Host to issue a Get Features command specifying Flexible Data Placement Events (FID 1Eh) with the Namespace Identifier (NSID) field set to FFFFFFFFh.

Observable Results:

1. Verify all commands prior to step 3 complete with Status Code ‘Successful Completion’ (00h).
2. Verify the Set Feature aborts with Status Code ‘Invalid Field in Command’ (02h).
3. Verify the Get Features command in step 4 is aborted with a status code of ‘Invalid Field in Command’ (02h).

Case 3: FDP Events Feature – FDP Disabled (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify command with CNS=01h (Identify Controller) and check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0h, then skip this test.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is not cleared to 0h, then disable the feature for this case.
3. Perform a Set Feature specifying Flexible Data Placement Events (FID=1Eh).

© 2026 UNH–IOL

Observable Results:

1. Verify all commands prior to step 3 complete with Status Code ‘Successful Completion’ (00h).
2. Verify the Set Feature aborts with Status Code ‘FDP Disabled’ (29h).

Case 4: FDP Events Feature – Events Enabled (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify command with CNS=01h (Identify Controller) and check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0h, then skip this test.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
3. Perform a Set Feature specifying Flexible Data Placement Events (FID=1Eh) with an Invalid Placement Handle (PHNDL).

Observable Results:

1. Verify all commands prior to step 3 complete with Status Code ‘Successful Completion’ (00h).
2. Verify that the Set Features command aborts with Status Code ‘Invalid Field in Command’ (02h).

© 2026 UNH–IOL

Test 9.13 – FDP Data Placement Directive (OF-FYI, FYI)

Case 1: FDP Data Placement Directive (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify command with CNS=01h (Identify Controller) and check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0, then skip this test.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is set to 1, disable the feature, then re-enable the feature.
3. Perform a Directive Send command specifying:
    1. The Directive Operation field set to Enable Directive (1h),
    2. The NSID of a namespace inside the FDP Enabled endurance group,
    3. The DTYPE field in Command Dword 11 set to Identify (00h),
    4. The DTYPE field in Command Dword 12 set to Data Placement (02h),
    5. The Enable Directive bit is set to 1.

Observable Results:

1. Verify all commands complete with Status Code ‘Successful Completion’.

Case 2: Data Placement Directive, Persistent Across Reset (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify command with CNS=01h (Identify Controller) and check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0h, then skip this test.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
3. Perform a Directive Send command specifying:
    a. The Directive Operation field set to Enable Directive (1h),
    b. The NSID of a namespace inside the FDP Enabled endurance group,
    c. The DTYPE field in Command Dword 11 set to Identify (00h),
    d. The DTYPE field in Command Dword 12 set to Data Placement (02h),
    e. The Enable Directive bit set to 1h.
4. Perform a Controller Level Reset.
5. Perform a Directive Receive command with the following parameters:
    a. Directive Operation (DO) Field set to Return Parameters (01h),
    b. Directive Type (DTYPE) in Command Dword 11 set to Identify (00h).

Observable Results:

1. Verify all commands complete with Status Code ‘Successful Completion’ (00h).
2. Verify the directive Receive command from step 5 has the Data Placement Directive enabled in the return structure.

Case 3: Data Placement Directive, NSID=FFFFFFFFh (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify command with CNS=01h (Identify Controller) and check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0, then skip this test.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is set to 1, disable the feature, then re-enable the feature. If the field is cleared to 0, then enable the feature.
3. Perform a Directive Send command with the following parameters:

© 2026 UNH–IOL

1. Directive Operation (DO) Field set to ‘Enable Directive’ (01h)
2. Directive Type (DTYPE) in CDW11 to ‘Identify’ (00h)
3. Directive Type (DTYPE) in CDW12 to ‘Data Placement’ (02h)
4. Enable Directive bit set to ‘1’
5. Namespace Identifier (NSID) set to FFFFFFFFh

Observable Results:

1. Verify all commands up to step 3 complete with Status Code ‘Successful Completion’.
2. Verify the Directive Send command aborts with Status Code ‘Invalid NSID’.

Case 4: Data Placement Directive, FDP Disabled (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify command with CNS=01h (Identify Controller) and check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0h, then skip this test.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is set to 1h, then disable the feature for this case.
3. Perform a Directive Send command with the following parameters:
    a. Directive Operation (DO) Field set to ‘Enable Directive’ (01h)
    b. Directive Type (DTYPE) in CDW11 to ‘Identify’ (00h)
    c. Directive Type (DTYPE) in CDW12 to ‘Data Placement’ (02h)
    d. Enable Directive bit set to ‘1h’.
    e. Namespace Identifier (NSID) set to a Namespace where FDP is disabled.

Observable Results:

1. Verify all commands up to step 3 complete with Status Code ‘Successful Completion’ (00h).
2. Verify the Directive Send command aborts with Status ‘FDP Disabled’ (29h).

Case 5: Data Placement Directive, Invalid Operation (OF-FYI, FYI)

Test Procedure:

1. Perform an Identify command with CNS=01h (Identify Controller) and check the Flexible Data Placement Support bit in the CTRATT field of the Identify Controller Data Structure. If it is cleared to 0h, then skip this test.
2. Perform a Get Feature with FID=1Dh (Flexible Data Placement) and check the FDPE (Flexible Data Placement Enable) field. If the field is set to 1h, disable the feature, then re-enable the feature. If the field is cleared to 0h, then enable the feature.
3. Perform a Directive Send command with the following parameters:
    a. Directive Operation (DO) Field set to ‘Enable Directive’ (01h)
    b. Directive Type (DTYPE) in CDW11 to ‘Data Placement’ (02h)
    c. Directive Type (DTYPE) in CDW12 to ‘Data Placement’ (02h)
    d. Enable Directive bit set to ‘1h’.
    e. Namespace Identifier (NSID) set to a Namespace where FDP is Enabled

Observable Results:

1. Verify all commands up to step 3 complete with Status ‘Successful Completion’ (00h).
2. Verify the Directive Send command fails with Status ‘Invalid Field in Command’ (02h).

© 2026 UNH–IOL

Test 9.14 – RUH Scope Reporting (OF-FYI, FYI)

Purpose: To verify that a controller supporting Flexible Data Placement (FDP) correctly reports the Reclaim Unit Handle Scope (RUHS) bit in the FID Supported and Effects data structure for applicable feature identifiers, per TP4215.

References:

NVM Express Base Specification: 5.2.12.1.18, 5.2.26.1.21
Technical Proposal 4215: Adding Reclaim Unit Handle Scope Reporting

Resource Requirements:

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

Last Modification: July 24, 2026

Discussion: TP4215 introduced the Reclaim Unit Handle (RUH) scope for features that impact specific reclaim unit handles within an Endurance Group. For a controller supporting the Flexible Data Placement capability, the Flexible Data Placement Events (FID 1Eh) feature is defined as having RUH scope.

Testing verifies that the controller accurately reports this scope in the Feature Identifiers Supported and Effects log page (LID 12h). Specifically, the RUHS bit (bit 7 of the FID Scope field) must be set to '1h' in the entry corresponding to FID 1Eh to indicate that modifying the attributes of this feature impacts Reclaim Unit Handles.

Test Setup: See Appendix A.

Test Procedure:

1. Issue an Identify command (CNS 01h) and check the Flexible Data Placement Support (FDPS) bit in the Controller Attributes (CTRATT) field of the Identify Controller data structure (Byte 99, bit 19). If it is cleared to '0h', this test is not applicable.
2. Issue a Get Log Page command for the Feature Identifiers Supported and Effects log page (LID 12h). If this LID is not supported, this test is not applicable.
3. Locate the FID Supported and Effects data structure entry for FID 1Eh (Flexible Data Placement Events).

Observable Results:

1. Verify the FID Supported (FSUPP) bit (bit 0 of the entry's first dword) for FID 1Eh is set to '1h'.
2. Verify the RUHS bit (bit 7 of the FID Scope (FSP) field, which corresponds to bit 27 of the entry's first dword) is set to '1h', indicating the controller reports the Reclaim Unit Handle scope.

© 2026 UNH–IOL

Group 10: Reachability Groups

Overview:
This section describes a method for performing conformance verification for NVMe products implementing NVMe Reachability Groups.

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).

© 2026 UNH–IOL

Test 10.1 – Reachability Groups (OF-FYI, FYI)

Case 1: Reachability Groups Log with RGO = 1h (OF-FYI, FYI)

Purpose: To verify proper implementation of the Reachability Groups Log Page in TP4156a.

Test Procedure:

1. Configure the NVMe Host to issue an Identify Command (CNS 01h) to the DUT. If RRSUP is set to 0 in the identify controller data structure, skip this test.
2. Configure the NVMe Host to issue a Get Log Page command with LID equal to 1Ah and RGO set to 1h in the Reachability Groups Log Specific Parameter Field.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that NNID is cleared to 0h in each of the Reachability Group Descriptors returned by the Get Log Page command in step 2.

Case 2: Reachability Groups Log Change Count (OF-FYI, FYI)

Purpose: To verify proper implementation of the Reachability Groups Log Page in TP4156a.

Test Procedure:

1. Configure the NVMe Host to issue an Identify Command (CNS 01h) to the DUT. If RRSUP or OACS bit 3 is cleared to 0h in the identify controller data structure, skip this test.
2. Perform a Controller Level Reset.
3. Configure the NVMe Host to issue a Get Log Page command with LID equal to 1Ah.
4. Repeat the following 5 times:
    a. Detach a namespace attached to the controller.
    b. Attach the namespace detached in step 4a to the controller.
5. Configure the NVMe Host to issue a Get Log Page command with LID equal to 1Ah.
6. Perform a Controller Level Reset.
7. Configure the NVMe Host to issue a Get Log Page command with LID equal to 1Ah.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify the Get Log Page in step 3 returns a Change Count (Bytes 07:00 of Reachability Groups Log Page) that is cleared to 0h.
3. Verify the Get Log Page in step 5 returns a Change Count (Bytes 07:00 of Reachability Groups Log Page) that is equal to Ah.
4. Verify the Get Log Page in step 7 returns a Change Count (Bytes 07:00 of Reachability Groups Log Page) that is cleared to 0h.

Case 3: Reachability Groups Log no Namespaces Attached (OF-FYI, FYI)

Purpose: To verify proper implementation of the Reachability Groups Log Page in TP4156a.

Test Procedure:

1. Configure the NVMe Host to issue an Identify Command (CNS 01h) to the DUT. If RRSUP or OACS bit 3 is cleared to 0h in the identify controller data structure, skip this test.
2. Detach all namespace attached to the controller.
3. Configure the NVMe Host to issue a Get Log Page command with LID equal to 1Ah.
4. Attach all namespace detached in step 2 to the controller.

Observable Results:

© 2026 UNH–IOL

1. Verify that all commands complete successfully.
2. Verify the Get Log Page in step 3 returns with NRGD cleared to 0h.

Case 4: Reachability Association Log with RGO = 1h (OF-FYI, FYI)

Purpose: To verify proper implementation of the Reachability Association Log Page in TP4156a.

Test Procedure:

1. Configure the NVMe Host to issue an Identify Command (CNS 01h) to the DUT. If RRSUP is set to 0 in the identify controller data structure, skip this test.
2. Configure the NVMe Host to issue a Get Log Page command with LID equal to 1Bh and RGO set to 1h in the Reachability Association Log Specific Parameter Field.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that NRID is cleared to 0h in each of the Reachability Association Descriptors returned by the Get Log Page command in step 2.

Case 5: Reachability Association Log no Namespaces Attached (OF-FYI, FYI)

Purpose: To verify proper implementation of the Reachability Association Log Page in TP4156a.

Test Procedure:

1. Configure the NVMe Host to issue an Identify Command (CNS 01h) to the DUT. If RRSUP is set to 0 or OACS bit 3 is cleared to 0h in the identify controller data structure, skip this test.
2. Detach all namespace attached to the controller.
3. Configure the NVMe Host to issue a Get Log Page command with LID equal to 1Bh.
4. Attach all namespace detached in step 2 to the controller.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify the Get Log Page in step 3 returns with NRAD cleared to 0h.

Case 6: Reachability Association Log Change Count (OF-FYI, FYI)

Purpose: To verify proper implementation of the Reachability Association Log Page in TP4156a.

Test Procedure:

1. Configure the NVMe Host to issue an Identify Command (CNS 01h) to the DUT. If RRSUP or OACS bit 3 is cleared to 0h in the identify controller data structure, skip this test.
2. Perform a Controller Level Reset.
3. Configure the NVMe Host to issue a Get Log Page command with LID equal to 1Bh.
4. Repeat the following 5 times:
    a. Detach a namespace attached to the controller.
    b. Attach the namespace detached in step 4a to the controller.
5. Configure the NVMe Host to issue a Get Log Page command with LID equal to 1Bh.
6. Perform a Controller Level Reset.
7. Configure the NVMe Host to issue a Get Log Page command with LID equal to 1Bh.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify the Get Log Page in step 3 returns a Change Count (Bytes 07:00 of Reachability Association Log Page) that is cleared to 0h.
3. Verify the Get Log Page in step 5 returns a Change Count (Bytes 07:00 of Reachability Association Log Page) that is equal to Ah.

© 2026 UNH–IOL

4. Verify the Get Log Page in step 7 returns a Change Count (Bytes 07:00 of Reachability Association Log Page) that is cleared to 0h.

Case 7: Reachability Group Identifier When Reachability Group Reporting is Not Supported (OF-FYI, M)

Purpose: To verify proper implementation of the additionally defined bytes in Identify CNS 08h in TP4156a.

Test Procedure:

1. Configure the NVMe Host to issue an Identify Command (CNS 01h) to the DUT. If RRSUP, isn't cleared to 0h in the identify controller data structure, this test is not applicable.
2. For each namespace attached:
3. Configure the NVMe Host to issue an Identify Command (CNS 08h) to the DUT record the RGRPID.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that RGRPID is cleared to 0h in the I/O Command Set Independent Identify Namespace Data Structure.

© 2026 UNH–IOL

Group 11: Boot Partition

Overview:
This section describes a method for performing conformance verification for NVMe products implementing Boot Partitions. These tests are not applicable to devices that do not claim to support Boot Partitions.

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).

Possible Problem: Tests that enable RPMB will prevent the DUT from being able to test Set Features Boot Partition Write Protection tests, as RPMB Boot Partition Write Protection cannot be disabled once enabled.

© 2026 UNH–IOL

Test 11.1 – Boot Partition Write Protection (OF-FYI, FYI)

Purpose: To determine if an NVMe Controller properly implements Boot Partition Write Protection mechanisms.

References:

NVM Express Base Specification 2.0d : 8.2

Resource Requirements:

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

Last Modification: August 1st, 2024

Discussion: Boot Partitions provide an optional area of NVM storage that may be read by the host without the host initializing queues or enabling the controller. The simplified interface to access Boot Partitions may be used for platform initialization code (e.g., a bootloader that is executed from host ROM) to boot to a pre-OS environment (e.g., UEFI) instead of storing the image on another non-volatile storage medium (e.g., SPI flash)

A controller that supports Boot Partitions has two Boot Partitions of equal size using Boot Partition identifiers 0h and 1h. The two Boot Partitions allow the host to update one and verify the contents before marking the Boot Partition active. Controllers in the NVM subsystem may share the same Boot Partitions.

The contents of Boot Partitions are only modified using the Firmware Image Download and Firmware Commit commands and may be secured using Replay Protected Memory Block to prevent unauthorized modifications

Test Setup: See Appendix A.

Case 1: Valid Values (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to read the CAP.BPS register field. If the value is cleared to zero then this test is not applicable.
2. Configure the NVMe Host to issue Get Log Page command for the Feature Identifiers Supported and Effects Log Page (LID 12h). If FSUPP is cleared to 0h for FID 85h, this test is not applicable.
3. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure to read the BPCAP field. If SFBPWPS is cleared to 0b, this test is not applicable.
4. Configure the NVMe Host to issue a Get Feature command to the NVMe Controller using FID 85h. If BP0WPS and BP1WPS are set to 0b100 (Write Protection controlled by RPMB), this test is not applicable.
5. Configure the NVMe Host to issue a Get Log Page command to LID 03h and record AFI (Active Firmware Info).
6. Configure the NVMe Host to issue a Set Feature command to the NVMe Controller using FID 85h, with BP0WPS and BP1WPS fields in CDW11 set to 000b.
7. Configure the NVMe Host to issue a Get Feature command to the NVMe Controller using FID 85h.
8. Repeat steps 6 and 7 using 001b for BP0WPS and BP1WPS fields in CDW11 of the Set Feature.
9. Repeat steps 6 and 7 using 010b for BP0WPS and BP1WPS fields in CDW11 of the Set Feature.
10. Repeat steps 6 and 7 using 011b for BP0WPS and BP1WPS fields in CDW11 of the Set Feature.
11. Configure the NVMe Host to issue a Set Feature command to the NVMe Controller with a Feature Identifier of 85h, with BP0WPS and BP1WPS fields in CDW11 set to 001b.
12. Power cycle the NVMe controller. Wait for the controller to become available to the NVMe Host before checking the Observable Results.
13. Configure the NVMe Host to issue a Get Feature command to the NVMe Controller using FID 85h.

Observable Results:

1. Verify that the Set Feature in step 6 is successful.
2. Verify that the Get Feature in step 7 is successful and reports 010b for both BP0WPS and BP1WPS fields.
3. Verify the Set Features command in step 12 aborts with the status code of ‘Feature Not Changeable’ (0Eh).

© 2026 UNH–IOL

4. Verify that the Get Feature in step 13 is successful and reports 010b for both BP0WPS and BP1WPS fields.

Case 2: Set Feature, RPMB Enabled (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe host to read the CAP.BPS register field, if the value is set to zero this test is not applicable.
2. Configure the NVMe Host to issue Get Log Page command for the Feature Identifiers Supported and Effects Log Page (LID 12h). If FSUPP is cleared to 0h for FID 85h, this test is not applicable.
3. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure to read the BPCAP field. If SFBPWPS is cleared to 0b or RPMBBPWPS is not set to 10b, and the DUT supports NVMe v2.0 or higher, this test is not applicable.
    a. If the DUT is less than NVMe v2.0, and Boot Partitions or RPMB are not supported, this test is not applicable.
4. Configure the NVMe Host to issue an Authenticated Data Write using a Security Send Command to the RPMB Device Configuration Block data structure to enable RPMB Boot Partition Write Protection.
5. Configure the NVMe Host to issue a Get Feature command to the NVMe Controller using FID 85h.
6. Configure the NVMe Host to issue a Set Feature command to the NVMe Controller using FID 85h, with BP0WPS and BP1WPS fields in CDW11 set to 100b.

Observable Results:

1. Verify that the Get Feature command returns a value of 100b for either Boot Partition 0 Write Protection State Field or the Boot Partition 1 Write Protection State Fields in CDW0 of the returned completion queue entry, indicating that RPMB was successfully enabled.
2. Verify that the Get Feature in step 7 is successful and reports 010b for both BP0WPS and BP1WPS fields.
3. Verify that the Firmware Commit command in step 8 aborts with the status of ‘Boot Partition Write Protected’ (1Eh).

Case 3: Set Feature, Feature Not Changeable (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe host to read the CAP.BPS register field. If the value is cleared to zero then this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command for the Feature Identifiers Supported and Effects Log Page (LID 12h). If FSUPP is cleared to 0h for FID 85h then this test is not applicable.
3. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. Read the BPCAP field. If SFBPWPS is cleared to 0b or RPMBBPWPS is not set to 10b, and the DUT supports NVMe v2.0 or higher, this test is not applicable.
    a. If the DUT is less than NVMe v2.0, and Boot Partitions or RPMB are not supported, this test is not applicable.
4. Configure the NVMe Host to issue an Authenticated Data Write using a Security Send Command to the RPMB Device Configuration Block data structure to enable RPMB Boot Partition Write Protection.
5. Configure the NVMe Host to issue a Set Feature command to the NVMe Controller using FID 85h, with BP0WPS and BP1WPS fields in CDW11 set to 001b.

Observable Results:

1. Verify that the Authenticated Data Write completes successfully. Verify that the Get Feature in step 7 is successful and reports 010b for both BP0WPS and BP1WPS fields.
2. Verify that the Set Feature command in step 5 aborts with the status code of ‘Feature Not Changeable’ (0Eh).

Case 4: Get Feature, RPMB Enabled (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe host to read the CAP.BPS register field, if the value is set to zero this test is not applicable.

© 2026 UNH–IOL

2. Configure the NVMe Host to issue Get Log Page command for the Feature Identifiers Supported and Effects Log Page (LID 12h). If FSUPP is cleared to 0h for FID 85h, this test is not applicable.
3. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure to read the BPCAP field. If RPMBBPWPS is not set to 10b, and the DUT supports NVMe v2.0 or higher, this test is not applicable.
    a. If the DUT is less than NVMe v2.0, and Boot Partitions or RPMB are not supported, this test is not applicable.
4. Configure the NVMe Host to issue an Authenticated Data Write using a Security Send Command to the RPMB Device Configuration Block data structure to enable RPMB Boot Partition Write Protection.
5. Configure the NVMe Host to issue a Get Feature command with SEL = 000b to the NVMe Controller using FID 85h.

Observable Results:

1. Verify that the Get Feature command returns a value of 100b for both the Boot Partition 0 Write Protection State Field and the Boot Partition 1 Write Protection State Fields in CDW0 of the returned completion queue entry.
2. Verify all commands completed successfully.

Case 5: Boot Partitions & RPMB Not Supported (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe host to read the CAP.BPS register field, if the value is cleared to zero this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure to read the RPMBS field. If the Number of RPMB Units is non-zero, this test is not applicable.
3. Configure the NVMe Host to issue Get Log Page command for the Feature Identifiers Supported and Effects Log Page (LID 12h).

Observable Results:

1. Verify that all commands completed successfully.
2. Verify that FID 85h is supported, which is required to configure the Boot Partition Write Protection if RPMBS is not supported.

Case 6: Boot Partitions & RPMB Supported (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe host to read the CAP.BPS register field, if the value is set to zero this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure to read the RPMBS field. If the Number of RPMB Units is cleared to zero, then this test is not applicable.
3. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure to read the BPCAP field.

Observable Results:

1. Verify that either FID 85h or RPMB Boot Partition Write Protection is supported by checking the BPCAP field in the Identify Controller Data Structure. If the DUT is v2.0 or earlier, RPMBBPWPS can be cleared to 0h.

Case 7: RPMB Boot Partition Write Protection Not Supported (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe host to read the CAP.BPS register field, if the value is set to zero this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure to read the RPMBS field and BPCAP field. For DUTs that support NVMe v2.0 or higher, if the Number of RPMB Units (RPMBS) is cleared to 0 or RPMBBPWPS is set to 10b, this test is not applicable

© 2026 UNH–IOL

3. Configure the NVMe Host to issue an Authenticated Data Write using a Security Send Command to the RPMB Device Configuration Block data structure to enable RPMB Boot Partition Write Protection.

Observable Results:

1. Verify that the Authenticated Data Write fails with status 'Write failure in the RPMB Operation Status field' (05h).

Case 8: Enable While in Write Locked Until Power Cycle State (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe host to read the CAP.BPS register field. If the value is cleared to zero then this test is not applicable.
2. Configure the NVMe Host to issue Get Log Page command for the Feature Identifiers Supported and Effects Log Page (LID 12h). If FSUPP is cleared to 0h for FID 85h then this test is not applicable.
3. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure to read the BPCAP and RPMBS field. For DUTs that support NVMe v2.0 or higher, if SFBPWPS is cleared to 0b, or RPMBBPWPS is not set to 10b, or the Number of RPMB Units is cleared to 0h, this test is not applicable.
4. Configure the NVMe Host to issue a Set Feature command to the NVMe Controller using FID 85h, with BP0WPS and BP1WPS fields in CDW11 set to 011b (Write Locked Until Power Cycle).
5. Configure the NVMe Host to issue an Authenticated Data Write using a Security Send Command to the RPMB Device Configuration Block data structure to enable RPMB Boot Partition Write Protection.
6. Power cycle the NVMe controller. Wait for the controller to become available to the NVMe Host before checking the Observable Results.

Observable Results:

1. Verify that the Authenticated Data Write fails with status 'Write failure in the RPMB Operation Status field' (05h).
2. Verify all commands completed successfully.

Case 9: Get Feature While RPMB Enabled (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe host to read the CAP.BPS register field, if the value is set to zero this test is not applicable.
2. Configure the NVMe Host to issue Get Log Page command for the Feature Identifiers Supported and Effects Log Page (LID 12h). If FSUPP is cleared to 0h for FID 85h, this test is not applicable.
3. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure to read the BPCAP and RPMBS field. For DUTs that support NVMe v2.0 or higher, if SFBPWPS is cleared to 0b, or RPMBBPWPS is not set to 10b, or the Number of RPMB Units is cleared to 0h, this test is not applicable.
4. Configure the NVMe Host to issue an Authenticated Data Write using a Security Send Command to the RPMB Device Configuration Block data structure to enable RPMB Boot Partition Write Protection.
5. Configure the NVMe Host to issue a Get Feature command with SEL = 000b to the NVMe Controller using FID 85h.

Observable Results:

1. Verify that the RPMB Authenticated Data Write command is successful.
2. Verify that the Boot Partition 0 Write Protection State and Boot Partition 1 Write Protection State values returned by the Get Feature command are set to 100b.

© 2026 UNH–IOL

Group 12: PCIe Infrastructure for Live Migration

Overview:
This section describes a method for performing conformance verification for NVMe products implementing the optional Track Send, Track Receive, Migration Receive, Migration Send, Controller Data Queue commands, and Controller Data Queue Feature. These tests are not applicable to devices that do not claim to support Host Managed Live Migration Features.

Note 1: 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).

Note 2: Many of the following test cases require multiple controllers in the same NVM Subsystem to test the full functionality of Live Migration, Controller Data Queues, and the Track Send/Receive commands. These test cases will utilize the terms “Controller A” to denote a Migration Management Controller (MMC), and “Controller B” to denote a migratable controller. These tests assume that Controller A is a valid Migration Management Controller (See NVM Express Base Specification, Revision 2.1, Section 8.1.12 “Host Managed Live Migration”) that sets the HMLMS bit to ‘1.’ Controller A is required to support the Migration Send command with the Suspend (0h), Resume (1h) and Set Controller State (2h) Management Operations, the Migration Receive command with the Get Controller State (0h) Management Operation, and the CNS value of 20h in the Identify command (Supported Controller State Formats data structure). In order for the host to differentiate between the two controllers, the following steps will be performed, and the Identify Controller Data Structures will be recorded and associated with their respective controller name for the duration of testing:

1. This test requires the NVMe Host to be aware of at least 2 different controllers in the same NVM subsystem. If this is not the case, then this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure for each of the controllers. If HMLMS is set to 1b and Controller A has yet to be assigned, then assign this controller as Controller A. If HMLMS is cleared to 0b and Controller B has yet to be assigned, then assign this controller as Controller B. Record the value of the CNTLID field for each controller.
3. If Controller A and Controller B are not assigned, then this test is not applicable.

© 2026 UNH–IOL

Test 12.1 – Host Managed Live Migration Support (FYI)

Purpose: To determine if an NVMe Controller that sets the HMLMS bit to '1' supports the following:

Migration Send command with the Management Operations:
- Suspend
- Resume
- Set Controller State

The Migration Receive command with Management Operations
- Controller State (i.e., 1h)
- The CNS value of 21h20h (typo in spec) in the Identify command

References:

NVM Express Base Specification 2.1d: 8.1.12

Resource Requirements:
- Tools capable of monitoring and decoding traffic on the NVMe interface.

Last Modification: July 31st, 2025

Test Setup: See Appendix A.

Case 1: HMLMS Support Valid Values (OF-FYI, FYI)

Test Procedure:
1. Configure the NVMe Host to issue an Identify command to Controller A to retrieve the Supported Controller State Formats (CNS 20h) data structure for Controller A.
2. Configure the NVMe Host to issue a Migration Receive command to Controller A, specifying the Get Controller State Management Operation (SEL=0h), CNTLID=CNTLID specifying Controller B, and CSVI=0 to retrieve an empty Controller State Data Structure.
3. Configure the NVMe Host to issue a Migration Send command to Controller A, specifying the Suspend Management Operation (SEL=0h), CNTLID=CNTLID specifying Controller B, and STYPE=1h.
4. Configure the NVMe Host to issue a Migration Send command to Controller A, specifying the Set Controller State Management Operation (SEL=2h), CNTLID=CNTLID specifying Controller B, CSVI=0, and SEQIND=11b.
5. Configure the NVMe Host to issue a Migration Send command to Controller A, specifying the Resume Management Operation (SEL=1h) and CNTLID=CNTLID specifying Controller B.

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

Case 2: CNS=01h Host Managed Live Migration Support (OF-FYI, M)

Test Procedure:
1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure.

Observable Results:
1. Verify that if the HMLMS bit is set to 1b, then HMPRE is cleared to 0h.
2. Verify that if the THMCS bit is cleared to 0b, then CMMRTD is cleared to 0h.
3. Verify that if the THMCS bit is cleared to 0b, then NMMRTD, MINMRTG, and MAXMRTG are cleared to 0h.
4. Verify that if the TUDCS bit is cleared to 0b, then MCUDMQ and MNSUDMQ are cleared to 0h.
5. Verify that NMCMR is greater than or equal to the value of MCMR.

© 2026 UNH–IOL

Test 12.2 – Host Managed Live Migration: Migration Receive (FYI)

Purpose: To test the controller capabilities of the Migration Receive command

References:

NVM Express Base Specification 2.1d: 5.1.16

Resource Requirements:

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

Last Modification: January 23rd, 2025

Test Setup: See Appendix A.

Case 1: Migration Receive command - Management Operation=0h (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Migration Receive command to Controller A with SEL=0h (Get Controller State) and the CNTLID of Controller B.

Observable Results:

1. Verify that all commands completed successfully.

Case 2: Migration Receive command - Offset Lower Dword Aligned (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Migration Receive command with SEL=0h (Get Controller State), a CNTLID of a Migratable Controller, and an OL=1h.

Observable Results:

1. Verify that the Migration Receive command completed with a status of ‘Invalid Field in Command’ (02h).

Case 3: Migration Receive command - Management Operation=0h, CSUP=0h, CSUP=1h (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Migration Receive command with SEL=0h (Get Controller State) and CNTLID of Controller B.
2. Configure the NVMe host to issue a Migration Send command to Controller A with SEL=0h (Suspend),STYPE=1h and CNTLID of Controller B.
3. Configure the NVMe host to issue a Migration Receive command to Controller A with SEL=0h (Get Controller State) and CNTLID of Controller B.
4. Configure the NVMe Host to issue a Migration Send command with Select (SEL) set to 1h for the Resume Management Operation before ending the test.

Observable Results:

1. Verify that all commands completed successfully.
2. Verify that the Migration Receive command step 1 completed successfully and has DW0 with CSUP cleared to 0h.
3. Verify that the Migration Receive command step 3 completed successfully and has DW0 with CSUP set to 1h.

Case 4: Migration Receive command - Management Operation=0h, Invalid CSVI (FYI)

Test Procedure:

© 2026 UNH–IOL

1. Configure the NVMe host to issue an Identify command to Controller A with CNS=20h (Supported Controller State Formats), and record the NV value.
2. Configure the NVMe host to issue a Migration Receive command to Controller A with SEL=0h (Get Controller State), a CNTLID of Controller B, and a CSVI value equal to NV + 1.

Observable Results:

1. Verify that the Migration Receive command step 2 completed with a status of 'Invalid Field in Command' (02h).
2. Verify all other commands completed successfully.

Case 5: Migration Receive command - Management Operation=0h, CSVI=0h (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Migration Receive command to a Controller A with SEL=0h (Get Controller State), CNTLID of Controller B, and a CSVI cleared to 0h.

Observable Results:

1. Verify that all commands completed successfully.
2. Verify that the NVMECSS and VSS fields in the data buffer returned by the Migration Receive command are cleared to 0h.

Case 6: Migration Receive command -Management Operation=0h, Invalid CSUUIDI (FYI)

Test Procedure:

1. Configure the NVMe host to issue an Identify command to Controller A with CNS=20h (Supported Controller State Formats), and record the NUUID value.
2. Configure the NVMe host to issue a Migration Receive command to Controller A with SEL=0h (Get Controller State), CNTLID of Controller B, and a CSUUIDI value equal to NUUID + 1.

Observable Results:

1. Verify that the Migration Receive command step 2 completed with a status of 'Invalid Field in Command' (02h).
2. Verify all other commands completed successfully.

Case 7: Controller State Data Structure Version Support and Check (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 20h Supported Controller State Formats. Read the value of Byte 0, Number of Versions (NV). If the value of NV is 0 this test is not applicable.
2. Configure the NVMe Host to issue a Migration Receive command to Controller A with the Management Operation Specific (MOS) field set to the lowest ordered Controller State Version Index (CSVI) number returned in the previous step, CNTLID of Controller B, and the Select (SEL) Field set to 0h (Get Controller State).

Observable Results:

1. Verify that all commands completed with a status code of "Success".
2. Verify that the Version (VER) field returned in step 2 of the test procedure is cleared to 0h.

Case 8: Controller State Data Structure NVMe Controller State Size Verification (FYI)

Test Procedure:

© 2026 UNH–IOL

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 20h Supported Controller State Formats. Read the value of Byte 0, Number of Versions (NV). If the value of NV is 0 this test is not applicable.
2. Configure the NVMe Host to issue a Migration Receive command to Controller A with the Management Operation Specific (MOS) field set to the lowest ordered Controller State Version Index (CSVI) number returned in the previous step, CNTLID of Controller B, and the Select (SEL) Field set to 0h (Get Controller State).
3. Read the NVMe Controller State Size (NVMECSS) field. If the value in this field is cleared to 0h, this test is not applicable. Otherwise, store the value of NVMECSS.
4. Calculate the total number of DWORDs the NVMe Controller State NVMECS field is for the Controller State Data entry returned in the Controller State Data Structure from step 1.

Observable Results:

1. Verify that all commands completed with a status code of “Success”.
2. Verify that the NVMe Controller State Size (NVMECSS) field and the value calculated in step 4 are equal.

© 2026 UNH–IOL

Test 12.3 – Host Managed Live Migration: Migration Send (FYI)

Purpose: To test the controller capabilities of the Migration Send command

References:

NVM Express Base Specification 2.1d: 5.1.17

Resource Requirements:

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

Last Modification: January 23rd, 2025

Test Setup: See Appendix A.

Case 1: Migration Send Command - Management Operation=0h, DUDMQ=1h (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Get Log command with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable. If the Controller Data Queue feature is not supported, this test is not applicable.
2. Configure the NVMe host to issue a Controller Data Queue command with SEL=0h (Create), QT=0h, CNTLID set to Controller B, and a PRP Entry specifying a portion of host memory to be used as a User Data Migration Queue, and set the PC bit to 1h.
3. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B with SEL=0h (Suspend), STYPE=1h and DUDMQ=1h.
4. Configure the NVMe host to issue a Get Feature command with FID set to 21h (Controller Data Queue) with the CDQID field in CDW11 set to the value returned by the Controller Data Queue command in step 2.

Observable Results:

1. Verify that the Get Feature command step 4 completed with a status of ‘Invalid Controller Data Queue’ (37h).
2. Verify all other commands completed successfully.

Case 2: Migration Send Command - Management Operation=0h, Command Processing (FYI)

Test Procedure:

1. Configure the NVMe host to issue an Asynchronous Event Request command to Controller B.
2. Configure the NVMe host to issue a number of write commands equal to the size of the I/O Submission Queue to Controller B.
3. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B with SEL=0h (Suspend) and STYPE=1h.
4. Configure the NVMe host to issue a Write command to Controller B.

Observable Results:

1. Verify that after steps 3 and 4, there are no Completion Queue Entries posted for the Asynchronous Event Request command in step 1 and the Write command in step 4.
2. Verify that all other commands completed successfully.

Case 3: Migration Send command - Management Operation=0h, Controller Already Suspended (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B, with SEL=0h (Suspend) and STYPE=1h.

© 2026 UNH–IOL

2. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B, with SEL=0h (Suspend) and STYPE=1h.
3. Configure the NVMe Host to issue a Migration Send command to Controller A specifying Controller B, with Select (SEL) set to 1h for the Resume Management Operation before ending the test.

Observable Results:

1. Verify all commands completed successfully and the Migration Send in step 3 did not cause an error status to be returned.

Case 4: Migration Send command - Management Operation=1h, Controller Not Suspended (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B with SEL=0h (Suspend) and STYPE=1h.
2. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B with SEL=1h (Resume).
3. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B with SEL=1h (Resume).

Observable Results:

1. Verify the Migration Send command in step 3 completed with a status of 'Controller Not Suspended' (3Ah).
2. Verify all other commands completed successfully.

Case 5: Migration Send command - Management Operation=2h, Controller State Not Verified or Committed (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B with SEL=0h (Suspend) and STYPE=1h.
2. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B with SEL=2h (Set Controller State) with CSUUIDI=0h, CSVI=0h, NUMD = Ch SEQIND=01b, and a zero filled data buffer.
3. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B with SEL=1h (Resume).

Observable Results:

1. Verify the Migration Send command in step 3 completed with a status of 'Command Sequence Error' (0Ch).
2. Verify all other commands completed successfully.

Case 6: Migration Send command - Management Operation=2h, Controller Not Suspended or Enabled (FYI)

Test Procedure:

1. Configure the NVMe host to set CC.EN to 0b (i.e. disable the controller).
2. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B with SEL=2h (Set Controller State), CSUUIDI=0h, CSVI=0h, NUMD=01h, SEQIND=11b, and a zero filled data buffer.

Observable Results:

1. Verify the Migration Send command in step 2 completed with a status of 'Invalid Controller Identifier' (1Fh).
2. Verify all other commands completed successfully.

© 2026 UNH–IOL

Case 7: Migration Send command - Management Operation=2h, SEQIND (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B with SEL=0h (Suspend) and STYPE=1h.
2. Configure the NVMe host to issue a 3 Migration Send commands to Controller A specifying Controller B all with SEL=2h (Set Controller State) with CSUUIDI=0h, CSVI=0h, NUMD = Ch, a zero filled data buffer, and with the values for SEQIND in the following order:
    a. 01b
    b. 00b
    c. 10b

Observable Results:

1. Verify all commands completed successfully.

Case 8: Migration Send command - Management Operation=2h, SEQIND=02h (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B with SEL=0h (Suspend) and STYPE=1h.
2. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B with SEL=2h (Set Controller State) with CSUUIDI=0h, CSVI=0h, SEQIND=02h, NUMD=Ch and a zero filled data buffer.

Observable Results:

1. Verify the Migration Send command in step 2 completed with a status of ‘Command Sequence Error’ (0Ch).

Case 9: Migration Send command - Management Operation=2h, CSOL not Dword Aligned (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B with SEL=2h (Set Controller State) with CSUUIDI=0h, CSVI=0h, NUMD=Ch, SEQIND=11h, CSOL=03h, and a zero filled data buffer.

Observable Results:

1. Verify the Migration Send command completed with a status of ‘Invalid Field in Command’ (02h).

Case 10: Migration Send command - Management Operation=2h, NUMD=0h (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B with SEL=2h (Set Controller State) with CSUUIDI=0h, CSVI=0h, NUMD=0h, SEQIND=11b, and CSOL=03h and a zero filled data buffer.

Observable Results:

1. Verify the Migration Send command completed with a status of ‘Invalid Field in Command’ (02h).

Case 11: Migration Send command - Management Operation=2h, Controller State Offset > NUMD (FYI)

Test Procedure:

© 2026 UNH–IOL

1. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B, with SEL=2h (Set Controller State), CSUUIDI=0h, CSVI=0h, NUMD=Ch, SEQIND=11b, and CSOU=04h, and a zero filled data buffer.

Observable Results:

1. Verify the Migration Send command completed with a status of 'Invalid Field in Command' (02h).

Case 12: Migration Send with Invalid Controller State Version Index (CSVI) in CDW11 (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 20h Supported Controller State Formats. Read the value of Byte 0, Number of Versions (NV). If the value of NV is 0, this test is not applicable.
2. Configure the NVMe Host to issue a Migration Send command to Controller A specifying Controller B, with Select (SEL) set to 0h for the Suspend Management Operation, and DWORD 11 configured with Delete User Data Migration Queue (DUDMQ) set to 0, and the Suspend Type (STYPE) Set to 1h (Suspend).
3. Configure the NVMe Host to issue a Migration Send command with Select (SEL) set to 2h (Set Controller State), with the Management Operation Specific Field Sequence Indicator (SEQIND) set to 11b, and Controller State Version Index (CSVI) in CDW11 set to a non-zero value that is greater than the value of NV returned in step 1.
4. Configure the NVMe Host to issue a Migration Send command with Select (SEL) set to 1h for the Resume Management Operation.

Observable Results:

1. Verify that the Migration Send command in step 3 was aborted by the controller with a status code of 'Invalid Field in Command'.
2. Verify that all other commands completed with a status code of Success.

Case 13: Migration Send Command with Mismatched Controller State Version Index (CSVI) and NVMe Controller State Size (NVMECSS) (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 20h Supported Controller State Formats. Read the value of Byte 0, Number of Versions (NV). If the value of NV is 0, or CNS 20h is not supported by the controller, this test is not applicable.
2. Configure the NVMe Host to issue a Migration Send command to Controller A specifying Controller B, with Select (SEL) set to 0h for the Suspend Management Operation, and DWORD 11 configured with Delete User Data Migration Queue (DUDMQ) set to 0, and the Suspend Type (STYPE) Set to 1h (Suspend).
3. Configure the NVMe Host to issue a Migration Send command to Controller A specifying Controller B, with Select (SEL) set to 2h (Set Controller State), with the Management Operation Specific Field Sequence Indicator (SEQIND) set to 11b, and Controller State Version Index (CSVI) in CDW11 set to 0, and NVMe Controller State Size (NVMECSS) set to 1.
4. Configure the NVMe Host to issue a Migration Send command to Controller A specifying Controller B, with Select (SEL) set to 1h for the Resume Management Operation.

Observable Results:

1. Verify that the Migration Send command in step 4 was aborted by the controller with a status code of Invalid Field in Command.
2. Verify that all other commands completed with a status code of Success.

Case 14: Migration Send with Invalid CSUUIDI (FYI)

Test Procedure:

© 2026 UNH–IOL

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 20h Supported Controller State Formats. Read the value of Byte 1, Number of UUIDs (NUUID) . If the value of NUUID is 0, this test is not applicable
2. Configure the NVMe Host to issue a Migration Send command to Controller A specifying Controller B, with Select (SEL) set to 0h for the Suspend Management Operation, and DWORD 11 configured with Delete User Data Migration Queue (DUDMQ) set to 0, and the Suspend Type (STYPE) Set to 1h (Suspend).
3. Configure the NVMe Host to issue a Migration Send command to Controller A specifying Controller B, with Select (SEL) set to 2h (Set Controller State), with the Management Operation Specific Field Sequence Indicator (SEQIND) set to 11b, and Controller State UUID Index (CSUUIDI) in CDW11 set to a value greater than the value of NUUID returned in step 2.
4. Configure the NVMe Host to issue a Migration Send command to Controller A specifying Controller B, with Select (SEL) set to 1h for the Resume Management Operation.

Observable Results:

1. Verify that the Migration Send command in step 3 was aborted by the controller with a status code of Invalid Field in Command.
2. Verify that all other commands completed with a status code of Success.

Case 15: Set Controller State via Migration Send Command when NVMECSS != 0 and IOSQs/IOCQs Exist (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Create I/O Completion Queue command to the controller
2. Configure the NVMe Host to issue a second Create I/O Submission Queue command to the controller.
3. Configure the NVMe Host to issue an Identify command to Controller A with CNS 20h Supported Controller State Formats. Read the value of Byte 0, Number of Versions (NV). If the value of NV is 0, this test is not applicable.
4. Configure the NVMe Host to issue a Migration Receive command to Controller A specifying Controller B, with the Management Operation Specific (MOS) field set to the lowest ordered Controller State Version Index (CSVI) number returned in the previous step, and the Select (SEL) Field set to 0h (Get Controller State). If the returned Controller State Data structure shows the NVMECSS field set to 0, this test is not applicable.
5. Configure the NVMe Host to issue a Migration Send command to Controller A specifying Controller B, with Select (SEL) set to 0h for the Suspend Management Operation, and DWORD 11 configured with Delete User Data Migration Queue (DUDMQ) set to 0, and the Suspend Type (STYPE) Set to 1h (Suspend).
6. Configure the NVMe Host to issue a Migration Send command to Controller A specifying Controller B, with Select (SEL) set to 1h for the Resume Management Operation.

Observable Results:

1. Verify that the Management Send command in step 5 was aborted by the controller with a status code of Invalid Field in Command.
2. Verify that all other commands completed with a status code of Success.

Case 16: Migration Send Command Set Controller State Operation with Unexpected Vendor Specific Size (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B with the SEL = 0h for the Suspend Management Operation with STYPE = 1h.
2. Configure the NVMe Host to issue a Migration Send command with Select (SEL) set to 2h for the Set Controller State Management Operation, the Controller State UUID Index (CSUUIDI) in Set Controller State Command Dword 11 to 0, and the Vendor Specific Size (VSS) in the Controller State structure set to a non-zero value.

Observable Results:

© 2026 UNH–IOL

1. Verify that the Migration Send command in step 2 was aborted by the controller with a status code of Invalid Field in Command.
2. Verify that all other commands completed with a status code of Success.

Case 17: Migration Send Command Set Controller State with No Format Indices Specified (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Migration Send command to Controller A specifying Controller B with the SEL = 0h for the Suspend Management Operation with STYPE = 1h.
2. Configure the NVMe Host to issue a Migration Send command to Controller A specifying Controller B, with the Select (SEL) to set 2h for the Set Controller State Management Operation. Configure the command to set the Controller State UUID Index (CSUUIDI) and the Controller State Version Index (CSVI) in Set Controller State Command Dword 11 to 0.

Observable Results:

1. Verify that the Migration Send command in step 2 was aborted by the controller with a status code of Invalid Field in Command.

© 2026 UNH–IOL

Test 12.4 – Controller Data Queue Command (FYI)

Purpose: To test the controller capabilities of the Controller Data Queue command

References:

NVM Express Base Specification 2.1d: 5.1.4

Resource Requirements:

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

Last Modification: January 23rd, 2025

Discussion: The Controller Data Queue command is used to manage queues in host memory that a controller posts a specific type of data. The Controller Data Queue command uses the Command Dword 10 field. The use of the PRP1 field, Command Dword 11 field, and Command Dword 12 field is specific to the management operation specified by the Select field.

Test Setup: See Appendix A.

Case 1: Controller Data Queue command - MCMR Exceeded (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. Record the value stored in MCMR. If MCMR is cleared to 0h, this test is not applicable.
2. Configure the NVMe host to issue a Get Log command with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.
3. Configure the NVMe Host to issue a Controller Data Queue command to Controller A with Management Operation 0h, PC bit cleared to 0b, CDQSIZE set to 8h, CNTLID=CNTLID specifying Controller B, and PRP List Pointer that contains a number of memory ranges greater than MCMR, aligned with CC.MPS.

Observable Results:

1. Verify that the Controller Data Queue command aborts with status of ‘Invalid Field in Command’ (02h).

Case 2: Controller Data Queue Command - NMCMR Exceeded (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. Record the value stored in NMCMR. If NMCMR is cleared to 0h, this test is not applicable.
2. Configure the NVMe host to issue a Get Log command with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.
3. Configure the NVMe Host to issue a Controller Data Queue command to Controller A with Management Operation 0h, PC bit cleared to 0b, CDQSIZE set to 8h, CNTLID=CNTLID specifying Controller B, and PRP List Pointer that contains a number of memory ranges greater than NMCMR, aligned with CC.MPS.

Observable Results:

1. Verify that the Controller Data Queue command aborts with status ‘Not Enough Resources’ (38h).

Case 3: Controller Data Queue Command – MNSUDMQ Exceeded (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 1 of TRATTR (Track User Data Changes Support) is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. Record the value stored in MNSUDMQ.

© 2026 UNH–IOL

3. Configure the NVMe host to issue a Get Log command with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.
4. Configure the NVMe Host to issue MNSUDMQ+1 Controller Data Queue commands to Controller A with Management Operation 0h, PC bit set to 1b, CDQSIZE set to 8h, CNTLID=CNTLID specifying Controller B, and a valid PRP Entry aligned with CC.MPS.

Observable Results:

1. Verify that the last Controller Data Queue command aborts with status 'Not Enough Resources' (38h).
2. Verify all other commands completed successfully.

Case 4: Controller Data Queue command - Invalid PRP Offset (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Get Log command with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Controller Data Queue command to Controller A with Management Operation 0h, PC bit set to 1b, CNTLID=CNTLID specifying Controller B, and PRP Entry 1 with bits 1:0 not cleared to 00b.

Observable Results:

1. Verify that the Controller Data Queue command aborts with status 'PRP Offset Invalid' (13h).

Case 5: Controller Data Queue command - PRP Entry 1 Invalid Memory (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Get Log command with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.
2. Check CAP.CMBS. If CAP.CMBS is cleared to 0, then this test is not applicable.
3. Configure the NVMe host to read the CMBMSC.CBA field.
4. Configure the NVMe Host to issue a Controller Data Queue command to Controller A with Management Operation 0h, PC bit set to 1b, CDQSIZE set to 8h, CNTLID=CNTLID specifying Controller B, and PRP Entry 1 specifying CMBMSC.CBA as the base address.

Observable Results:

1. Verify that the Controller Data Queue command aborts with status 'Invalid Field in Command' (02h).

Case 6: Controller Data Queue command - Invalid CDQSIZE (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Get Log command with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Controller Data Queue command to Controller A with Management Operation 0h, PC bit set to 1b, CNTLID=CNTLID specifying Controller B, a valid PRP Entry 1, and CDQSIZE cleared to 1h.

Observable Results:

1. Verify that the Controller Data Queue command aborts with status 'Invalid Field in Command' (02h).

Case 7: Controller Data Queue command - Equal CQS and CNTLID (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Get Log command with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.

© 2026 UNH–IOL

2. Configure the NVMe Host to issue a Controller Data Queue command to Controller A with Management Operation 0h, PC bit set to 1b, CDQSIZE set to 8h, valid PRP Entry 1 aligned with CC.MPS, and the CQS field set to the same CNTLID as the controller in use.

Observable Results:

1. Verify that the Controller Data Queue command aborts with status 'Invalid Field in Command' (02h).

Case 8: Controller Data Queue command - Exceed MCUDMQ (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Get Log command with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue MCUDMQ +1 Controller Data Queue commands to Controller A with Management Operation 0h, PC bit set to 1b, CNTLID=CNTLID specifying Controller B, CDQSIZE set to 8h, and a valid PRP Entry aligned with CC.MPS.

Observable Results:

1. Verify that the final Controller Data Queue command aborts with status 'Not Enough Resources' (38h).
2. Verify all other commands completed successfully.

Case 9: Controller Data Queue command - CDQ ID does not exist (FYI)

Test Procedure:

1. Configure the NVMe host to perform a Controller Level Reset.
2. Configure the NVMe host to issue a Get Log command with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.
3. Configure the NVMe Host to issue a Controller Data Queue command to Controller A with Management Operation 1h, CNTLID=CNTLID specifying Controller B, and CDW11 set to 1h.

Observable Results:

1. Verify that the Controller Data Queue command aborts with the status 'Invalid Controller Data Queue' (37h).

Case 10: Log User Data Changes - Controller Data Queue Entry Phase Tag (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Get Log command to Controller A with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 1 of TRATTR (Track User Data Changes Support) is cleared to 0, this test is not applicable.
3. Configure the NVMe Host to issue a Controller Data Queue command to Controller A with SEL=0h (Create), QT=0h, CNTLID=CNTLID specifying Controller B, CDQSIZE=8h, PC=1b, and a PRP Entry aligned with CC.MPS specifying a portion of host memory to be used as a User Data Migration Queue. Record the value of CDQID from CQE Dword 0.
4. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=1h, and CDQID=CDQID from step 1 to perform a Log User Data Changes management operation to start logging.
5. Configure the NVMe Host to issue a Write command to any LBA in any namespace associated with Controller B to populate the User Data Migration Queue in host memory.
6. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=0h, and CDQID=CDQID from step 1 to perform a Log User Data Changes management operation to stop logging.

Observable Results:

1. Verify that all commands complete successfully

© 2026 UNH–IOL

2. Verify that the LBA Migration Queue Entry located in host memory contains a CDQP value of 0 after step 3.
3. Verify that the LBA Migration Queue Entry located in host memory contains a CDQP value of 1 after step 5.

Case 11: Log User Data Changes - Controller Reset Stops Logging (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Get Log command to Controller A with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 1 of TRATTR (Track User Data Changes Support) is cleared to 0, this test is not applicable.
3. Configure the NVMe Host to issue a Controller Data Queue command to Controller A with SEL=0h (Create), QT=0h, CNTLID=CNTLID specifying Controller B, CDQSIZE=8h, PC=1b, and a PRP Entry aligned with CC.MPS specifying a portion of host memory to be used as a User Data Migration Queue. Record the value of CDQID from CQE Dword 0.
4. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=1h, and CDQID=CDQID from step 1.
5. Configure the NVMe Host to issue a Write command to any LBA on any namespace associated with Controller B.
6. Configure the NVMe Host to perform a Controller Level Reset on Controller A.
7. Configure the NVMe Host to issue a Track Send command with SEL=0h, LACT=0h, and CDQID=CDQID from step 1.

Observable Results:

1. Verify that the Controller Data Queue contains a single entry after step 5.
2. Verify that the Track Send command in step 7 aborts with the status ‘Invalid Controller Data Queue’ (37h).
3. Verify that all other commands complete successfully.

Case 12: Log User Data Changes - ESA = 000b, LBACIR = 000b (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Get Log command to Controller A with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 1 of TRATTR (Track User Data Changes Support) is cleared to 0, this test is not applicable.
3. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=1h, and CDQID=CDQID from step 1.
4. Wait for Controller A to log the first entry in the Controller Data Queue.
5. Configure the NVMe Host to issue a Write command to any LBA on a single namespace associated with Controller B.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the LBA Migration Queue Entry posted after step 6 reports ESA=000b and LBACIR=000b.

Case 13: ESA Values 000-011b, 111b (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Get Log command to Controller A with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.

© 2026 UNH–IOL

2. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 1 of TRATTR (Track User Data Changes Support) is cleared to 0, this test is not applicable.
3. Configure the NVMe Host to issue a Controller Data Queue command to Controller A with SEL=0h (Create), QT=0h, CNTLID=CNTLID specifying Controller B, CDQSIZE=100h, PC=1b, and a PRP Entry aligned with CC.MPS specifying a portion of host memory to be used as a User Data Migration Queue. Record the value of CDQID from CQE Dword 0.
4. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=1h, and CDQID=CDQID from step 1.
5. Wait for Controller A to log the first entry in the Controller Data Queue.
6. Configure the NVMe Host to issue a Write command to any LBA on any namespace associated with Controller B.
7. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=0h, and CDQID=CDQID from step 1.
8. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=1h, and CDQID=CDQID from step 1.
9. Configure the NVMe Host to issue a Migration Send command to Controller A with SEL=0h (Suspend), STYPE=1h, and CNTLID=CNTLID specifying Controller B.
10. Configure the NVMe Host to issue a Migration Send command to Controller A with SEL=1h (Resume) and CNTLID=CNTLID specifying Controller B.
11. Configure the NVMe Host to issue Write commands to any LBA on any namespace associated with Controller B until the LBA Migration Queue is full.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the LBA Migration Queue Entry generated after step 4 contains an ESA of 001b.
3. Verify that the LBA Migration Queue Entry generated after step 6 contains an ESA of 000b.
4. Verify that the LBA Migration Queue Entry generated after step 7 contains an ESA of 010b.
5. Verify that the LBA Migration Queue Entry generated after step 9 contains an ESA of 011b.
6. Verify that the last LBA Migration Queue Entry generated after step 11 contains an ESA of 111b.

Case 14: Empty CDQ with Invalid Head Pointer Field (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Get Log command with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable. If the Controller Data Queue feature is not supported, this test is not applicable.
2. Configure the NVMe Host to issue a Controller Data Queue Command to Controller A with the Select (SEL) set to Create Controller Data Queue (Management Operation 0h), and the Management Operation Specific (MOS) set to 0h (User Data Migration Queue), the Create Queue Specific field set to Controller B.
3. Issue a Get Features command with FID 21h, record the value of Head Pointer field in the returned data structure.
4. Configure the NVMe Host to issue a Set Features command specifying feature identifier 21h (Controller Data Queue), and set the Enable Tail Pointer Trigger (ETPT) to 0 and Tail Pointer Trigger (TPT) set to 0, and Head Pointer (HP) set to any value other than 0 or the value returned and recorded in the previous step.

Observable Results:

1. Verify that the Set Features command in step 4 completes with a status code of ‘Invalid Field in Command’ (02h).
2. Verify that all other commands complete with a status code of ‘Success’.

Case 15: Tail Pointer Not Associated with the Controller Data Queue (FYI)

Test Procedure:

© 2026 UNH–IOL

1. Configure the NVMe host to issue a Get Log command with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable. If the Controller Data Queue feature is not supported, this test is not applicable.
2. Configure the NVMe Host to issue a Controller Data Queue Command to Controller A with the Select (SEL) set to Create Controller Data Queue (Management Operation 0h), and the Management Operation Specific (MOS) set to 0h (User Data Migration Queue), a CDQSIZE to 10h, and the Create Queue Specific field set to Controller B.
3. Configure the NVMe Host to issue a Set Features command specifying feature identifier 21h (Controller Data Queue), Head Pointer set to 0, Enable Tail Pointer Trigger (ETPT) set to 1h, and Tail Pointer Trigger (TPT) set to 1h.

Observable Results:

1. Verify that the Set Features command in step 3 completes with a status code of Invalid Field in Command.
2. Verify that all other commands complete with a status code of Success.

Case 16: Specifying a Non-Existent Controller Data Queue (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Controller Level Reset to clear all Controller Data Queues.
2. Configure the NVMe Host to issue a Set Features command specifying feature identifier 21h (Controller Data Queue), with CDQID cleared to 0.

Observable Results:

1. Verify that the Set Features command in step 3 completes with a status code of Invalid Controller Data Queue.

© 2026 UNH–IOL

Test 12.5 – Host Managed Live Migration: Track Send (FYI)

Purpose: To test the controller capabilities of the Track Send command

References:

NVM Express Base Specification 2.1d: 5.1.27

Resource Requirements:

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

Last Modification: January 23rd, 2025

Discussion: The Track Send command is used to manage the tracking of information by a controller (e.g., what to track during the migration of a controller in an NVM subsystem). The Track Send command uses the Command Dword 10 field. The use of the Data Pointer field and Command Dword 11 field is specific to the management operation specified by the Select field.

Test Setup: See Appendix A.

Case 1: Tracked Memory Changes - Reported Memory Range Granularity Alignment (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 0 of TRATTR (Track Host Memory Changes Support) is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=1h, TACT=1b, CNTLID=CNTLID specifying Controller B. Populate the data buffer specifying host memory ranges that Controller B will modify in the next step.
3. Configure the NVMe Host to issue an Identify Command to obtain the Identify Controller (CNS 01h) data structure for Controller B. Note the location in host memory where the data structure is stored.
4. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=1h, TACT=0b, CNTLID=CNTLID specifying Controller B.
5. Configure the NVMe Host to issue multiple Track Receive commands to Controller A with SEL=0h, CNTLID=CNTLID specifying Controller B to retrieve all tracked memory changes through multiple Tracked Memory Change Data Structures. Stop issuing new commands when NTMCD = 0h. Record the value of RPMPG for each data structure.

Observable Results:

1. Verify that all commands complete successfully.
2. For each Tracked Memory Changed Descriptor in each Tracked Memory Change Data Structure received in step 5, verify the value of SADDR in each descriptor is aligned to the granularity specified by the RPMPG field from the associated Data Structure.

Case 2: Log User Data Changes - Tracking After LACT=1b, No Tracking After LACT=0b (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Get Log command to Controller A with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 1 of TRATTR (Track User Data Changes Support) is cleared to 0, this test is not applicable.

© 2026 UNH–IOL

3. Configure the NVMe Host to issue a Controller Data Queue command to Controller A with SEL=0h (Create), QT=0h, CNTLID=CNTLID specifying Controller B, CDQSIZE=10h, PC=1b, and a PRP Entry aligned with CC.MPS specifying a portion of host memory to be used as a User Data Migration Queue. Record the value of CDQID from CQE Dword 0.
4. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=1h, and CDQID=CDQID from step 3 to perform a Log User Data Changes management operation to start logging.
5. Configure the NVMe Host to issue a Write command to any LBA in any namespace associated with Controller B to populate the User Data Migration Queue in host memory.
6. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=0h, and CDQID=CDQID from step 3 to perform a Log User Data Changes management operation to stop logging.
7. Configure the NVMe Host to issue a Write command to a different LBA in the same namespace associated with Controller B as step 3.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the User Data Migration Queue contains different data after step 4 and step 5.
3. Verify that the User Data Migration Queue contains the same data after step 6 and step 7.

Case 3: Log User Data Changes - Already Logging Controller Data Queue (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Get Log command to Controller A with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 1 of TRATTR (Track User Data Changes Support) is cleared to 0h, this test is not applicable.
3. Configure the NVMe Host to issue a Controller Data Queue command to Controller A with SEL=0h (Create), QT=0h, CNTLID=CNTLID specifying Controller B, PC=1b, and a PRP Entry aligned with CC.MPS specifying a portion of host memory to be used as a User Data Migration Queue. Record the value of CDQID from CQE Dword 0.
4. Configure the NVMe Host to issue two Track Send commands to Controller A with SEL=0h, LACT=1h, and CDQID=CDQID from step 3 to perform a Log User Data Changes management operation to start logging.
5. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=0h, and CDQID=CDQID from step 3 to perform a Log User Data Changes management operation to stop logging.

Observable Results:

1. Verify that the second Track Send command fails with the status code of ‘Invalid Controller Data Queue’ (37h).
2. Verify that all other commands complete successfully.

Case 4: Log User Data Changes - CDQID Associated with Suspended Controller (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Get Log command to Controller A with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 1 of TRATTR (Track User Data Changes Support) is cleared to 0, this test is not applicable.
3. Configure the NVMe Host to issue a Controller Data Queue command to Controller A with SEL=0h (Create), QT=0h, CNTLID=CNTLID specifying Controller B, PC=1b, CDQSIZE=10h, and a PRP Entry aligned with CC.MPS specifying a portion of host memory to be used as a User Data Migration Queue. Record the value of CDQID from CQE Dword 0.

© 2026 UNH–IOL

4. Configure the NVMe Host to issue a Migration Send command to Controller A with SEL=0h (Suspend), STYPE=1h, and CNTLID=CNTLID specifying Controller B.
5. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=1h, and CDQID=CDQID from step 3.
6. Configure the NVMe Host to issue a Migration Send command to Controller A with SEL=1h (Resume) and CNTLID=CNTLID specifying Controller B.

Observable Results:

1. Verify that the Track Send command returns a status code of 'Controller Suspended'.

Case 5: Tracked Memory Changes - Memory Range Tracking Granularity Bounds (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 0 of TRATTR (Track Host Memory Changes Support) is cleared to 0h, this test is not applicable
2. Record the values of MAXMRTG and MINMRTG from the Identify Controller (CNS 01h) Data Structure of Controller A.
3. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=1h, TACT=1b, CNTLID=CNTLID specifying Controller B, and populate the Track Memory Changes Data Structure with RMRTG < MINMRTG
4. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=1h, TACT=1b, CNTLID=CNTLID specifying Controller B, and populate the Track Memory Changes Data Structure with RMRTG > MAXMRTG.

Observable Results:

1. Verify that each Track Send command returns a status code of 'Invalid Field in Command' (02h).

Case 6: Tracked Memory Changes - RNMRTD > CMMRTD (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 0 of TRATTR (Track Host Memory Changes Support) is cleared to 0h, this test is not applicable
2. Record the value of CMMRTD from the Identify Controller (CNS 01h) Data Structure of Controller A.
3. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=1h, TACT=1b, CNTLID=CNTLID specifying Controller B, and populate the Track Memory Changes Data Structure with RNMRTD > CMMRTD from Controller A.

Observable Results:

1. Verify that the Track Send command aborts with status 'Invalid Field in Command' (02h).

Case 7: Tracked Memory Changes - RNMRTD > NMMRTD (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 0 of TRATTR (Track Host Memory Changes Support) is cleared to 0h, this test is not applicable.
2. Record the value of NMMRTD from the Identify Controller (CNS 01h) Data Structure of Controller A.
3. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=1h, TACT=1b, CNTLID=CNTLID specifying Controller B, and populate the Track Memory Changes Data Structure with RNMRTD > NMMRTD from Controller A.

Observable Results:

© 2026 UNH–IOL

1. Verify that the Track Send command aborts with status ‘Invalid Field in Command’ (02h).

Case 8: Tracked Memory Changes - Already Tracking Controller (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 0 of TRATTR (Track Host Memory Changes Support) is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=1h, TACT=1b, CNTLID=CNTLID specifying Controller B.
3. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=1h, TACT=1b, CNTLID=CNTLID specifying Controller B.
4. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=1h, TACT=0b, CNTLID=CNTLID specifying Controller B.

Observable Results:

1. Verify that the Track Send commands in steps 2 and 4 complete successfully.
2. Verify that the Track Send command in step 2 fails with status ‘Invalid Controller Identifier’.

Case 9: Track Memory Changes - MRTLL=1b, Descriptor LEN is a power of 2 (FYI)

Test Procedure:

1. Record the value of MRTLL and THMCS from the Identify Controller (CNS 01h) Data Structure of Controller A. If MRTLL or THMCS is cleared to 0b, then this test is not applicable.
2. Configure the NVMe host to issue a Track Send command to Controller A with SEL=1h, TACT=1b, CNTLID=CNTLID specifying Controller B. Populate the Track Memory Changes Data Structure with one Memory Range Tracking Descriptor with LEN=3h.

Observable Results:

1. Verify that the Track Send command fails with status ‘Invalid Field in Command’ (02h).

Case 10: Track Memory Changes - TACT=0h, Track Memory Changes Data Structure Ignored (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 0 of TRATTR (Track Host Memory Changes Support) is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=1h, TACT=1b, CNTLID=CNTLID specifying Controller B.
3. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=1h, TACT=0b, CNTLID=CNTLID specifying Controller B. Populate the Track Memory Changes Data Structure with VER=1h.

Observable Results:

1. Verify that all commands complete successfully.

Case 11: Track Memory Changes - Tracking After TACT=1b, No Tracking After TACT=0b (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 0 of TRATTR (Track Host Memory Changes Support) is cleared to 0h, this test is not applicable.

© 2026 UNH–IOL

2. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=1h, TACT=1b, CNTLID=CNTLID specifying Controller B. Populate the data buffer specifying host memory ranges that Controller B will modify in the next step.
3. Configure the NVMe Host to issue an Identify Command to obtain the Identify Controller (CNS 01h) data structure for Controller B. Note the location in host memory where the data structure is stored.
4. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=1h, TACT=0b, CNTLID=CNTLID specifying Controller B.
5. Configure the NVMe Host to issue multiple Track Receive commands to Controller A with SEL=0h, CNTLID=CNTLID specifying Controller B to retrieve all tracked memory changes through multiple Tracked Memory Change Data Structures. Stop issuing new commands when NTMCD = 0h.
6. Configure the NVMe Host to issue an Identify Command to obtain the Identify Controller (CNS 01h) data structure for Controller B. Note the location in host memory where the data structure is stored.
7. Configure the NVMe Host to issue a Track Receive command to Controller A with SEL=0h, and CNTLID=CNTLID specifying Controller B.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that at least one Tracked Memory Change Data Structure from step 5 reports NTMCD > 0h.
3. Verify that the Tracked Memory Change Data Structure from step 7 reports NTMCD = 0h.

Case 12: Track Memory Changes - Controller is Suspended (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 0 of TRATTR (Track Host Memory Changes Support) is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Migration Send command to Controller A with SEL=0h (Suspend), STYPE=1h, and CNTLID=CNTLID specifying Controller B.
3. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=1h, TACT=1h, and CNTLID=CNTLID specifying Controller B.
4. Configure the NVMe Host to issue a Migration Send command to Controller A with SEL=1h (Resume) and CNTLID=CNTLID specifying Controller B.

Observable Results:

1. Verify that the Track Send command aborts with status code ‘Controller Suspended’.
2. Verify that all other commands complete successfully.

Case 13: Track Memory Changes - Incorrect Version (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 0 of TRATTR (Track Host Memory Changes Support) is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=1h, TACT=1b, CNTLID=CNTLID specifying Controller B. Populate the Track Memory Changes Data Structure with VER set to a non-zero value.

Observable Results:

1. Verify that the Track Send command fails with status code ‘Invalid Field in Command’ (02h).

Case 14: Track Memory Changes - Descriptor SADDR not aligned to RMRTG (FYI)

Test Procedure:

© 2026 UNH–IOL

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 0 of TRATTR (Track Host Memory Changes Support) is cleared to 0h, this test is not applicable.
2. Configure the NVMe host to issue a Track Send command to Controller A with SEL=1h, TACT=1b, and CNTLID=CNTLID specifying Controller B. Populate the Track Memory Changes Data Structure with one Memory Range Tracking Descriptor with SADDR and RMRTG configured such that SADDR is not aligned to the granularity specified by the RMRTG field.

Observable Results:

1. Verify that the Track Send command fails with status code 'Invalid Field in Command' (02h).

Case 15: Log User Data Changes - Track Send Clears Controller Data Queue Full Error Event (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 1 of TRATTR (Track User Data Changes Support) is cleared to 0, this test is not applicable.
2. Configure the NVMe host to issue a Get Log command to Controller A with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.
3. Configure the NVMe Host to issue a Controller Data Queue command to Controller A with SEL=0h (Create), QT=0h, CNTLID=CNTLID specifying Controller B, CDQSIZE=8h, PC=1b, and a PRP Entry aligned with CC.MPS specifying a portion of host memory to be used as a User Data Migration Queue. Record the value of CDQID from CQE Dword 0.
4. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=1h, and CDQID=CDQID from step 3.
5. Configure the NVMe Host to issue multiple Write commands to different LBAs on any namespace(s) associated with Controller B until the Controller Data Queue (LBA Migration Queue) in host memory is full.
6. Configure the NVMe Host to issue an Asynchronous Event Request to Controller A.
7. Configure the NVMe Host to perform a Controller Level Reset on Controller A.
8. Configure the NVMe Host to issue a Controller Data Queue command to Controller A with SEL=0h (Create), QT=0h, CNTLID=CNTLID specifying Controller B, CDQSIZE=10h, PC=1b, and a PRP Entry aligned with CC.MPS specifying a portion of host memory to be used as a User Data Migration Queue. Record the value of CDQID from CQE Dword 0.
9. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=1h, and CDQID=CDQID from step 8.
10. Configure the NVMe Host to issue multiple Write commands to different LBAs on any namespace(s) associated with Controller B until the Controller Data Queue (LBA Migration Queue) in host memory is full.
11. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=0h, and CDQID=CDQID from step 8.
12. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=1h, and CDQID=CDQID from step 8.
13. Configure the NVMe Host to issue an Asynchronous Event Request to Controller A.

Observable Results:

1. Verify the Asynchronous Event Request from step 6 posts a Completion Queue Entry reporting a Controller Data Queue Full Error Event.
2. If the Track Send command in step 12 aborts with status code 'Controller Data Queue Full', then this test is not applicable.
3. Verify that no completion queue entry is posted for the Asynchronous Event Request in step 13.
4. Verify that all other commands complete successfully.

© 2026 UNH–IOL

Case 16: Log User Data Changes - LBACIR=10b, NSID & NLB Cleared to 0h (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A with CNS 01h Identify Controller Data Structure, I/O Command Set Independent. If bit 1 of TRATTR (Track User Data Changes Support) is cleared to 0, this test is not applicable.
2. Configure the NVMe host to issue a Get Log command to Controller A with LID=05h (Commands Supported and Effects). If the CSUPP field for the Controller Data Queue command (ACS45) is cleared to 0h, this test is not applicable.
3. Configure the NVMe Host to issue a Controller Data Queue command to Controller A with SEL=0h (Create), QT=0h, CNTLID=CNTLID specifying Controller B, CDQSIZE=10h, PC=1b, and a PRP Entry aligned with CC.MPS specifying a portion of host memory to be used as a User Data Migration Queue. Record the value of CDQID from CQE Dword 0.
4. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=1h, and CDQID=CDQID from step 3.
5. Configure the NVMe Host to issue a Track Send command to Controller A with SEL=0h, LACT=0h, and CDQID=CDQID from step 3.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that LBA Migration Queue Entry posted after step 4 reports LBACIR=10b, NSID=0h, SLBA = 0h, and NLB=0h.
3. Verify that LBA Migration Queue Entry posted after step 5 reports LBACIR=10b, NSID=0h, SLBA = 0h, and NLB=0h.

Case 17: Log User Data Changes - Controller Data Queue Doesn’t Exist (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A to retrieve the Identify Controller (CNS 01h) Data Structure. If TUDCS is cleared to 0b, then this test is not applicable. Record the value of CNTLID.
2. Configure the NVMe Host to issue a Track Send command Controller A to with SEL=0h, LACT=1h, and CDQID=1h.

Observable Results:

1. Verify that the Identify command completes successfully.
2. Verify that the Track Send command fails with status ‘Invalid Controller Data Queue’ (37h).

© 2026 UNH–IOL

Test 12.6 – Host Managed Live Migration: Track Receive (FYI)

Purpose: To test the controller capabilities of the Track Receive command

References:

NVM Express Base Specification 2.1d: 5.1.26

Resource Requirements:

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

Last Modification: January 23rd, 2025

Discussion: The Track Receive command is used to obtain the information being tracked by the controller that was enabled by a Track Send command. The Track Receive command uses the Data Pointer field, Command Dword 10 field, and Command Dword 12 field. The use of the Command Dword 11 field is specific to the management operation specified by the Select field.

Test Setup: See Appendix A.

Case 1: Track Receive Command VER=0h (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command Controller A to obtain the Identify Controller (CNS 01h) data structure. If THMCS is cleared to 0b, then this test is not applicable.
2. Configure the NVMe Host to issue a Track Receive command to Controller A specifying Controller B, with SEL=0h to perform a Tracked Memory Changes management operation. Record the value of the VER field in the Tracked Memory Change Data Structure.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the VER field is cleared to 0h.

Case 2: Track Receive Command MTR=0b (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to Controller A specifying Controller B, obtain the Identify Controller (CNS 01h) data structure. If THMCS is cleared to 0b, then this test is not applicable.
2. Configure the NVMe Host to issue a Track Receive command to Controller A specifying Controller B, with SEL=0h to perform a Tracked Memory Changes Management Operation.
3. If the Number of Tracked Memory Changed Descriptors (NTMCD) in the received Tracked Memory Change Data Structure contains a value other than 0h, repeat the previous step until it contains a value of 0h.
4. Record the value of the More To Report (MTR) field in the latest Tracked Memory Change Data Structure.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify the MTR field from step 4 is cleared to 0b.

Case 3: Track Receive Operations Clear Previously Tracked Memory (FYI)

Test Procedure:

1. Configure the NVMe Host to read the Track Host Memory Changes Support (THMCS) within the Tracking Attributes (TRATTR) in the Identify Controller Data Structure, I/O Command Set Independent data structure for Controller A. If cleared to 0, this test is not applicable.

© 2026 UNH–IOL

2. Configure the NVMe Host to issue a Track Send command to Controller A specifying Controller B, with the Track Memory Changes Management Operation specified in the Select field, and the Management Operation Specific Field Tracking Action set to 1, and a data buffer that specifies host memory ranges that the controller will write to in the next step.
3. Configure the NVMe Host to issue an Identify Command to Controller B, specifying CNS 01h (Identify Controller data structure). Note the location in host memory that the data buffer returned from this command is stored.
4. Configure the NVMe Host to issue a Track Receive command to Controller A specifying Controller B, with the Tracked Memory Changes Management Operation.
5. Configure the NVMe Host to issue an Identify command with CNS 00h to Controller B.
6. Configure the NVMe Host to issue a second Track Receive command to Controller A specifying Controller B, with the Tracked Memory Changes Management Operation.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the returned data in step 6 does not contain the data from the returned by the Track Receive command in step 4.

© 2026 UNH–IOL

Group 13: Configurable Device Personality

Overview:
This section describes a method for performing conformance verification for NVMe products implementing support for Configurable Device Personality. These tests assess correct reporting, selection, and runtime behavior of supported device personalities through applicable Identify data structures, Feature settings, and command handling. These tests are not applicable to devices that do not claim support for configurable or multi-personality operation.

Note 1: These tests have the following requirements:

1. Authenticated Unfreeze Support must not be cleared to 0h. If there is no mechanism to unfreeze the personality, the device will be put into an untestable state.
2. If any of the personalities are frozen, there must be a pre-shared key supplied.
3. If there are vendor defined personalities, each personality will need to be tested to ensure that the various supported features between personalities conform to the NVMe specifications.

© 2026 UNH–IOL

Test 13.1 – Personality Freeze and Unfreeze Behavior (FYI)

Purpose: The purpose of the Personality Freeze and Unfreeze Behavior tests is to verify that the controller correctly enforces administrative control and security boundaries around Configurable Device Personalities as defined in NVMe 2.3. These tests ensure that once a personality is placed into a frozen state, the device prevents unauthorized or unintended modifications, and that any supported unfreeze mechanisms operate strictly according to specification.

References:

NVM Express Base Specification 2.3: 8.1.6

Resource Requirements:

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

Last Modification: February 5th, 2026

Test Setup: See Appendix A.

Case 1: Changing a Frozen Personality Maintains Values (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). Record the PERID value of a Personality Properties entry with a Programmable Key Authentication Support (PKAS) set to 01h. If the Device Personalities Log is not supported, or the returned log page does not contain a Personality Properties entry with the PKAS field set to 1h, this test is not applicable.
2. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to 1h and record the Personality Freeze State (PERFS).
3. If PERFS is cleared to 0, configure the NVMe Host to issue a Set Features command with FID=22h (Configurable Device Personality), a PERID set to 1h, and PERFS set to 1h.
4. Configure the NVMe Host to issue a Get Features command with FID=22h (Configurable Device Personality), SEL = 00b, and a PERID value identified in step 1. Record the values returned.
5. Configure the NVMe Host to issue a Set Features command for FID=22h (Configurable Device Personality) with CHPS=0h, PERID=01h, and PERFS=01h.
6. Configure the NVMe Host to issue a Get Features command with FID=22h (Configurable Device Personality), SEL = 00b, and a PERID value identified in step 1. Record the values returned.

Observable Results:

1. Verify that the Set Features command in step 5 completes with command status ‘Feature not Changeable’ (0Eh).
2. Verify that the values returned by the Get Features command in step 4 match the values returned by the Get Features command in step 6.
3. Verify that all other commands completed successfully.

Case 2: Feature Not Changeable When Personality Requirements Not Met (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). If the Device Personalities Log is not supported, or the returned log page does not contain a Personality Properties entry with the PERID field set to a non-vendor-specific value that is modifiable by the host, this test is not applicable.
2. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to the value identified in step 1, SEL = 00b, and record the values. If the Personality Manufacturing Default Settings State (PMDSS) field is cleared to 0h, or the PERFS field is set to 1h, this test is not applicable.

© 2026 UNH–IOL

3. Configure the NVMe Host to issue a Set Features command for FID=22h (Configurable Device Personality) with CHPS=01h, PERID to the value identified in step 1, and a data pointer specifying different values than recorded in step 2.
4. Configure the NVMe Host to issue a Set Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to the value identified in step 1 and Change Personality Settings (CHPS) set to 1h.

Observable Results:

1. Verify that the Set Features command in step 4 completes with status code 'Feature not Changeable' (0Eh).

Case 3: Set Features with PERFS = 1 and CHPS = 1 Returns Feature Not Changeable (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). Record the PERID value of a Personality Properties entry with a Programmable Key Authentication Support (PKAS) set to 01h. If the Device Personalities Log is not supported, or the returned log page does not contain a Personality Properties entry with the PKAS field set to 1, this test is not applicable.
2. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to the value determined in step 1 and record the Personality Freeze State (PERFS).
3. If PERFS is cleared to 0, configure the NVMe Host to issue a Set Features command with FID=22h (Configurable Device Personality), a PERID set to the value determined in step 1, and PERFS set to 1h.
4. Configure the NVMe Host to issue a Set Features command for FID=22h (Configurable Device Personality) with CHPS=1h, PERID = the value determined in step 1, and PERFS=01h.

Observable Results:

1. Verify that the Set Features command in step 4 completes with status code 'Feature not Changeable' (0Eh).
2. Verify that all other commands completed successfully.

Case 4: Freeze and Unfreeze All Personalities With PERID=FFh (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). Record the PERID value of all Personality Properties entries with a Programmable Key Authentication Support (PKAS) set to 01h. If the Device Personalities Log is not supported, or the returned log page does not contain a Personality Properties entry with the PKAS field set to 1h, this test is not applicable.
2. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to each PERID recorded from the previous step and record the Personality Freeze State (PERFS). Record each Personality Identifier with a PERFS value set to 0h.
3. If PERFS is cleared to 0h, configure the NVMe Host to issue a Set Features command with FID=22h (Configurable Device Personality), a PERID set to FFh, and PERFS set to 1h.
4. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to each PERID recorded from step 2 and record the Personality Freeze State (PERFS).
5. Configure the NVMe Host to issue a Set Features command with FID=22h (Configurable Device Personality), a PERID set to FFh, PERFS set to 1h, and a supported unfreeze authentication method for each personality recorded in step 2.
6. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to each PERID recorded from step 2 and record the Personality Freeze State (PERFS).

Observable Results:

1. Verify that the PERFS of every Personality Identifier is set to 1h in step 4.

© 2026 UNH–IOL

2. Verify that all commands completed successfully.
3. Verify that the PERFS of every Personality Identifier is set to 0h in step 6.

Possible Problem: Not all personalities are required to have the same authentication method or unfreezing requirements. Therefore, it is possible that some personalities are left in a frozen state and will need to be manually unfrozen.

Case 5: Unfreezing Personality with Unsupported Authentication Method Remains Frozen (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). Record the PERID value of a Personality Properties entry with only one authentication method supported (e.g. Programmable Key Authentication Support (PKAS) set to 01h and PCAS set to 0h). If the Device Personalities Log is not supported, or the returned log page does not contain a Personality Properties entry with only one supported authentication mechanism, this test is not applicable.
2. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to the value determined in step 1 and record the Personality Freeze State (PERFS).
3. If PERFS is cleared to 0h, configure the NVMe Host to issue a Set Features command with FID=22h (Configurable Device Personality), a PERID set to the value determined in step 1, and PERFS set to 1h.
4. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, SPSP=02h, and a data pointer specifying an unsupported authentication method.
5. Configure the NVMe Host to issue a Get Features command with FID=22h (Configurable Device Personality), and a PERID to the value determined in step 1, to record the PERFS bit.
6. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, SPSP=02h.

Observable Results:

1. Verify that the PERFS in step 5 is set to 1h.
2. Verify that the Security Receive command returns a successful completion and a CDP Response Data Frame containing the Result Code value of ‘Authentication Method Not Supported’ (04h).

Case 6: Security Personality – Attribute Change Logged in Persistent Event Log (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). If the Device Personalities Log is not supported or the returned log page does not contain a Personality Properties entry with the PERID field set to 1h, this test is not applicable.
2. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to 1h and record the Security Protocol Status (SPS) of a security protocol which reports a value of 1h in the Supported Security Protocol (SSP) field. If the Personality Freeze State (PERFS) in the completion queue entry is set to 1h or if the Supported Security Protocol (SSP) field is cleared to 0h, this test is not applicable.
3. Configure the NVMe Host to issue a Set Features command for FID=22h (Configurable Device Personality) with Change Personality Settings (CHPS) set to 1h, Personality Identifier (PERID) set to 1h, and a data pointer specifying a Security Personality Attributes Data Structure with the Allow Security Protocol set to the opposite of the value recorded in step 2, and the bit corresponding to the relevant Security Protocol set to 1h.
4. Configure the NVMe Host to issue a Get Log Page command with LID=0Dh (Persistent Event Log).

Observable Results:

1. Verify all commands completed successfully.
2. Verify that the Persistent Event Log returned by the Get Log Page command in step 4 contains a new entry describing a Configurable Device Personality Change Event.

© 2026 UNH–IOL

Case 7: CDP Programmable Key – Pre-Existing Key (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). If the Device Personalities Log is not supported or the returned log page does not contain a Personality Properties entry with the Programmable Key Authentication Support (PKAS) field set to 1h, this test is not applicable.
2. Configure the NVMe Host to issue an Identify Controller command. If the HMAC-SHA-384 bit in the Configurable Device Personality Attributes field is cleared to 0h this test is not applicable.
3. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF value of 0002h for CDP Authentication, and a data pointer specifying a valid CDP Derived HMAC Key Programming Request Data Frame.
4. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication
5. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF value of 0002h for CDP Authentication, and a data pointer specifying a valid CDP Derived HMAC Key Programming Request Data Frame.
6. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication.

Observable Results:

1. Verify all commands completed successfully.
2. Verify the Data Pointer returned by the Security Receive command in step 4 contains a Result Code field set to ‘Operation Successful’ (00h).
3. Verify the Data Pointer returned by the Security Receive command in step 6 contains a Result Code field set to a non-zero value.

Case 8: CDP Programmable Key – Key Not Programmed After Personality Switch (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). Record the PERID value of a Personality Properties entry with a Programmable Key Authentication Support (PKAS) set to 01h. If the Device Personalities Log is not supported, or the returned log page does not contain a Personality Properties entry with the PKAS field set to 1h, this test is not applicable.
2. Configure the NVMe Host to issue an Identify Controller command. If the HMAC-SHA-384 bit in the Configurable Device Personality Attributes field is cleared to 0h this test is not applicable.
3. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF value of 0002h for CDP Authentication, and a data pointer specifying a valid CDP Derived HMAC Key Programming Request Data Frame.
4. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication.
5. Configure the NVMe Host to issue a Set Feature command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to 0h.
6. Configure the NVMe Host to issue a Set Feature command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to the PERID recorded in step 1 and a PERFS set to 01h.
7. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF value of 0002h for CDP Authentication, and a data pointer specifying a valid CDP Programmable Key Authentication Unfreeze Request Data Frame with a PERID field set the PERID recorded in step 1 and an HMAC Value calculated using the Host Programmed Key specified in the Security Send command in step 3.
8. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication

Observable Results:

1. Verify all commands completed successfully.

© 2026 UNH–IOL

2. Verify the Data Pointer returned by the Security Receive command in step 4 contains a Result Code field set to ‘Operation Successful’ (00h).
3. Verify the Data Pointer returned by the Security Receive command in step 6 contains a Result Code field set to ‘DHK Not Programmed’ (03h).

Case 9: CDP Programmable Key – Host Key Not Purged After Reset or Sanitize (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). Record the PERID value of a Personality Properties entry with a Programmable Key Authentication Support (PKAS) set to 01h. If the Device Personalities Log is not supported, or the returned log page does not contain a Personality Properties entry with the PKAS field set to 1h, this test is not applicable.
2. Configure the NVMe Host to issue an Identify Controller command. If the HMAC-SHA-384 bit in the Configurable Device Personality Attributes field is cleared to 0h this test is not applicable.
3. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF value of 0002h for CDP Authentication, and a data pointer specifying a valid CDP Derived HMAC Key Programming Request Data Frame.
4. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication.
5. Configure the NVMe Host to trigger a Controller Level Reset.
6. Configure the NVMe Host to issue a Sanitize command.
7. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF value of 0002h for CDP Authentication, and a data pointer specifying a valid CDP Derived HMAC Key Programming Request Data Frame.
8. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication.

Observable Results:

1. Verify all commands completed successfully.
2. Verify the Data Pointer returned by the Security Receive command in step 4 contains a Result Code field set to ‘Operation Successful’ (00h).
3. Verify the Data Pointer returned by the Security Receive command in step 6 contains a Result Code field set to a non-zero value.

Case 10: CDP Programmable Key – Improper HMAC Size (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). Record the PERID value of a Personality Properties entry with a Programmable Key Authentication Support (PKAS) set to 01h. If the Device Personalities Log is not supported, or the returned log page does not contain a Personality Properties entry with the PKAS field set to 1, this test is not applicable.
2. Configure the NVMe Host to issue an Identify Controller command. If the HMAC-SHA-384 bit in the Configurable Device Personality Attributes field is cleared to 0h this test is not applicable
3. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication and a Request Message Type of 05h for a Random Nonce Request
4. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication.
5. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF value of 0002h for CDP Authentication, and a data pointer specifying a valid CDP Derived HMAC Key Programming Request Data Frame.
6. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication.

© 2026 UNH–IOL

7. Configure the NVMe Host to issue a Set Features command for FID=22h (Configurable Device Personality) with CHPS=0h, PERID=01h, and PERFS=01h.
8. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF value of 0002h for CDP Authentication, and a data pointer specifying a valid CDP Programmable Key Authentication Unfreeze Request with PERID=01h, a Random Nonce value and size equal to the one returned in the CDP Random Nonce Read Response Data Frame in step 5, an HMAC value calculated using the Host Program Key specified in step 3, an HMAC size which does not correspond to the calculated HMAC value.
9. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication.
10. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to 1h.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that the Security Receive command in step 9 returned a data pointer with a CDP Response Data Frame containing a Result Code field set to 'Authentication Failure' (02h).
3. Verify that all other Security Receive commands returned a data pointer with a CDP Response Data Frame containing a Result Code field cleared to 'Operation Successful' (0h).
4. Verify that the Get Features command in step 10 returns a Dword 0 with a PERFS field set to 01h.

Case 11: CDP Programmable Key – Personality Unfreeze Request Purges Nonce (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). Record the PERID value of a Personality Properties entry with a Programmable Key Authentication Support (PKAS) set to 01h. If the Device Personalities Log is not supported, or the returned log page does not contain a Personality Properties entry with the PKAS field set to 1h, this test is not applicable.
2. Configure the NVMe Host to issue an Identify Controller command. If the HMAC-SHA-384 bit in the Configurable Device Personality Attributes field is cleared to 0h this test is not applicable
3. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication and a Request Message Type of 05h for a Random Nonce Request
4. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication
5. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF value of 0002h for CDP Authentication, and a data pointer specifying a valid CDP Derived HMAC Key Programming Request Data Frame.
6. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication
7. Configure the NVMe Host to issue a Set Features command for FID=22h (Configurable Device Personality) with CHPS=0h, PERID=01h, and PERFS=01h.
8. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF value of 0002h for CDP Authentication, and a data pointer specifying a valid CDP Programmable Key Authentication Unfreeze Request with PERID=01h, and an HMAC value and size calculated using the Host Program Key specified in step 3 and a Random Nonce value and size equal to the one returned in the CDP Random Nonce Read Response Data Frame in step 7.
9. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication
10. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to 1h.
11. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication and a Request Message Type of 05h for a Random Nonce Request.

© 2026 UNH–IOL

12. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that all Security Receive commands return a data pointer with a Result Code field cleared to 'Operation Successful' (0h).
3. Verify that the Get Features command in step 10 returns a Dword 0 with a PERFS field cleared to 0h.
4. Verify that the Security Receive command in step 12 returns a data pointer with a Random Nonce field which does not equal the Random Nonce field in the data pointer returned by the Security Receive command in step 5.

Case 12: Security Personality – Security Protocol Status (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). If the Device Personalities Log is not supported or the returned log page does not contain a Personality Properties entry with the PERID field set to 1h, this test is not applicable.
2. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to 1h and record a Security Protocol which reports a value of 1h in the Supported Security Protocol (SSP) and Security Protocol Status (SPS) fields. If the Personality Manufacturing Default Settings State (PMDSS) field is cleared to 0h, the PERFS field is set to 1h, the SSP field is cleared to 0h, or the SPS field is cleared to 0h, this test is not applicable.
3. Configure the NVMe Host to issue a Set Features command for FID=22h (Configurable Device Personality) with CHPS=01h, PERID=01h, and a data pointer specifying a Security Personality Data Structure with an Allow Security Protocol (ASP) cleared to 0h and a value of 01h for the bit corresponding the security protocol recorded in step 2.
4. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to 1h.

Observable Results:

1. Verify that all commands completed successfully.
2. Verify that the Completion Queue Entry of the Get Features command in step 4 contains a Personality Freeze State field set to 01h.

Case 13: Security Personality – SSP=0h (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). If the Device Personalities Log is not supported or the returned log page does not contain a Personality Properties entry with the PERID field set to 1h, this test is not applicable. If all Security Protocols are supported as indicated in the SSP field, this test is not applicable.
2. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to 1h and record a Security Protocol which reports a value of 0h in the Supported Security Protocol (SSP) fields.
3. Configure the NVMe Host to issue a Set Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to 1h and a data pointer specifying a Security Personality Data Structure with an Allow Security Protocol (ASP) set to 1 and a value of 1 for the bit corresponding the security protocol recorded in step 2.

Observable Results:

1. Verify that the Set Feature command in step 3 completed with a status of 'Invalid Field in Command' (02h).
2. Verify all other commands completed successfully.

© 2026 UNH–IOL

Case 14: Security Receive – Security Receive without Security Send (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication and a Request Message Type of 05h for a Random Nonce Request.
2. Configure the NVMe Host to trigger a Controller Level Reset.
3. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication.

Observable Results:

1. Verify that the Security Receive command in step 3 completed with a status of 'Command Sequence Error' (0Ch).

Case 15: CDP Authentication – CDP Request REQDFV=0h, DFHL=4h (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication with a REQDFV field set to a value other than 0h.
2. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication with a DFHL field set to a value other than 04h.

Observable Results:

1. Verify that both Security Send commands complete with a status code of 'Invalid Field in Command' (02h).

Case 16: CDP Authentication – CDP Response RESPDFV=0h, DFHL=4h, RESPDFS=07h (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication and a Request Message Type of 05h for a Random Nonce Request.
2. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication.

Observable Results:

1. Verify that all commands completed successfully.
2. Verify that the CDP Response Data Frame returned by the Security Receive command has a RESPDFV field cleared to 0h, a DFHL field set to 4h, a RESPDFS field set to 07h, and the Result Code is 'Operation Successful' (00h).

Case 17: CDP Authentication – Derived HMAC Key Programming Responses (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). Record the PERID value of a Personality Properties entry with a Programmable Key Authentication Support (PKAS) set to 01h. If the Device Personalities Log is not supported, or the returned log page does not contain a Personality Properties entry with the PKAS field set to 1h, this test is not applicable.
2. Configure the NVMe Host to issue an Identify Controller command. If the HMAC-SHA-384 bit in the Configurable Device Personality Attributes field is cleared to 0h this test is not applicable.
3. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF value of 0002h for CDP Authentication, and a data pointer specifying a valid CDP Derived HMAC Key Programming Request Data Frame.

© 2026 UNH–IOL

4. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication.

Observable Results:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). Record the PERID value of a Personality Properties.
2. Verify the Data Pointer returned by the Security Receive command in step 4 contains a CDP Response Message Type field set to 04h, a RESPDFS field cleared to 0h, and a Result Code field status indicating 'Operation Successful' (00h).

Case 18: CDP Authentication – CLR Purges Nonce (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication and a Request Message Type of 05h for a Random Nonce Request.
2. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication.
3. Configure the NVMe Host to initiate a Controller Level Reset.
4. Configure the NVMe Host to issue a Security Send command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication and a Request Message Type of 05h for a Random Nonce Request.
5. Configure the NVMe Host to issue a Security Receive command specifying Security Protocol EAh, an NSSF Specific value of 0002h for CDP Authentication.

Observable Results:

1. Verify that all commands completed successfully.
2. Verify that the CDP Response Data Frame returned by the Security Receive commands in steps 2 and 5 have a Result Code field cleared to 'Operation Successful' (0h).
3. Verify that the Security Receive command in step 5 returns a data pointer with a Random Nonce field which does not equal the Random Nonce field in the data pointer returned by the Security Receive command in step 2.

© 2026 UNH–IOL

Test 13.2 – Device Personalities Log Page Behavior (FYI)

Purpose: The Device Personalities Log Page provides a host-readable description of the personalities that are implemented in an NVM subsystem and their current administrative state (e.g., frozen vs. unfrozen) as introduced with the Configurable Device Personality (CDP) capability in NVMe 2.3. These tests verify that the log page is present when CDP is supported, that all defined fields are populated with valid/consistent values, and that state transitions and security preconditions are accurately reflected to software.

References:

NVM Express Base Specification 2.3: 8.1.6

Resource Requirements:

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

Last Modification: February 5th, 2026

Test Setup: See Appendix A.

Case 1: PEL entry creation with PERID set to 00h (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). If the returned log page does not contain a Personality Properties entry with the Programmable Key Authentication Support (PKAS) field set to 1h, this test is not applicable.
2. Configure the NVMe Host to issue a Set Features command for FID=22h (Configurable Device Personality) Personality Identifier (PERID) set to 00h.
3. Configure the NVMe Host to issue a Get Log Page command with LID=0Dh (Persistent Event Log).

Observable Results:

1. Verify that the Persistent Event Log returned by the Get Log Page command in step 3 contains a new entry describing a Configurable Device Personality Change Event.
2. Verify that PERID in the CDP Change Event Data is set to PERID 00h.
3. Verify all commands completed successfully.

Case 2: Personality 0 is Manufacturing Default (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). If the Device Personalities Log is not supported this test is not applicable.

Observable Results:

1. Verify that PERID for Personality 0 in the CDP Personality list is set to 00h.
2. Verify all commands completed successfully.

Case 3: Version 0h Device Personality 1Dh (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). If the Device Personalities Log is not supported this test is not applicable.

Observable Results:

1. Verify that CDP Log Page Version is cleared to 0h and DPLPHL is set to 6h.
2. Verify that the Log is cleared 0h past byte DPLPHL + PPS * (Personality NUMP)-1.
3. Verify all commands completed successfully.

© 2026 UNH–IOL

Case 4: Personality changed effects PSCUDE and UDCC (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). If the Device Personalities Log is not supported or the returned log page does not contain a Personality Properties entry with the PSCUDE field set to 1h, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities).
3. Configure the NVMe Host to issue a Get Log Page command with LID=12h (Feature Identifiers Supported and Effects).

Observable Results:

1. Verify all commands completed successfully.
2. Verify that Personality 0 has PSCUDE set to 1h.
3. Verify that User Data Content Change (UDCC) is set to 1h.

Case 5: PMDSS Clears and Sets (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). If the Device Personalities Log is not supported this test is not applicable. Record a supported non vendor specific PERID from the CDP Personality listed.
2. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with the recorded Personality Identifier (PERID) and record the returned data structure.
3. Configure the NVMe Host to issue a Set Features command for FID=22h (Configurable Device Personality) with Change Personality Settings (CHPS) set to 1h, Personality Identifier (PERID) set to the recorded PERID.
4. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to the recorded PERID and record PMDSS.
5. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to 0h and record PMDSS.
6. Configure the NVMe Host to issue a Security Send command with CDP Request Data configured to initiate an unfreeze request for the Security Personality.
7. Configure the NVMe Host to issue a Set Features command for FID=22h (Configurable Device Personality) with Change Personality Settings (CHPS) set to 1h, and the Personality Identifier (PERID) set to 0h.
8. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to the recorded PERID and record PMDSS.
9. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to FFh and record PMDSS.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that PMDSS is cleared to 0h in Dword 0 returned by the Get Features command in step 4.
3. Verify that PMDSS is cleared to 0h in Dword 0 returned by the Get Features command in step 5.
4. Verify that PMDSS is set to 1h in Dword 0 returned by the Get Features command in step 8.
5. Verify that PMDSS is set to 1h in Dword 0 returned by the Get Features command in step 9.

Sub-Cases: For each of the following subcases, unless otherwise noted, perform the same test procedure for case 13.2.5 and expect the same observable results.

1. PERID 01h

Sub-case Procedure:

1. Configure the NVMe Host to issue a Get Log Page command with LID=1Dh (Device Personalities). If the Device Personalities Log is not supported or if LID=1h contains no Personality Properties with PERID set to 01h, this test is not appliable.
2. Configure the NVMe Host to issue a Get Features command for FID=22h (Configurable Device Personality) with a Personality Identifier (PERID) set to 1h and record the Security Protocol Status (SPS) of

© 2026 UNH–IOL

a security protocol which reports a value of 1h in the Supported Security Protocol (SSP) field. If the Supported Security Protocol (SSP) field is cleared to 0h, this test is not applicable.

3. Configure the NVMe Host to issue a Set Features command for FID=22h (Configurable Device Personality) with Change Personality Settings (CHPS) set to 1h, Personality Identifier (PERID) set to 1h, and a data pointer specifying a Security Personality Attributes Data Structure with the Allow Security Protocol set to the opposite of the value recorded in step 2, and the bit corresponding to the relevant Security Protocol set to 1h.

© 2026 UNH–IOL

Test 13.3 – Pending State Verifications (FYI)

Purpose: The Pending State Verification tests validate controller behavior when a personality change is pending and has not yet taken effect. In NVMe 2.3, certain personality modifications—initiated via Set Features or other administrative mechanisms—may require a reset or additional action before becoming active. During this interval, the controller is required to enter a well-defined pending state and to enforce specific behavioral restrictions until the transition is completed or abandoned.

References:

NVM Express Base Specification 2.3: 8.1.6

Resource Requirements:

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

Last Modification: February 5th, 2026

Test Setup: See Appendix A.

Case 1: Sanitize Prohibited During Pending Personality Change (OF-FYI, FYI)

Test Procedure:

1. Verify the controller supports the CDP capability by checking bit 0 (FSUPP) for FID 22h in the Feature Identifiers Supported and Effects log page (LID 12h). If CDP is not supported, this test is not applicable.
2. Issue a Get Log Page command for the Device Personalities log page (LID 1Dh) to identify a supported personality that has an MRSTT value of 01h (Controller Level Reset) or higher. If no personalities have MRSTT with value >= 1, this test is not applicable. Ensure there is a Supported Personality in the Device Personalities Log Page within the CDP Personality List that is not vendor specific. If there are no non-vendor specific personalities listed as supported, this test is not applicable because there is no specification defined behavior.
3. Confirm the controller supports at least one sanitize operation (Block Erase, Crypto Erase, or Overwrite) by checking the SANICAP field in the Identify Controller data structure. If no sanitize operations are supported, this test is not applicable.
4. Configure the NVMe Host to issue a Set Features command for FID 22h (CDP) targeting the personality identified in step 2. Set the Change Personality Settings (CHPS) bit to '1h' and provide a valid configuration in the data buffer that differs from the current setting.
5. Configure the host to issue a supported Sanitize command.
6. Perform the Minimum Required Reset (indicated by the personality's MRSTT) to clear the pending state and return the DUT to the pre-test state.

Observable Results:

1. Verify all commands except the Sanitize command complete with a status of 'Successful Completion' (00h).
2. Verify that the PPSC bit was reported as '1h' in step 4.
3. Verify that the sanitize command in Step 5 is aborted by the controller with a status code of 'Access Denied' (86h).

Case 2: CDP Modification Restrictions during Pending Change (OF-FYI, FYI)

Test Procedure:

1. Verify the controller supports CDP (FID 22h) via the Feature Identifiers Supported and Effects log page (LID 12h).
2. Issue a Get Log Page for the Device Personalities log (LID 1Dh). Identify and record two supported personalities: Personality A (which must have an MRSTT > 00h) and Personality B. If there is only one personality supported, or neither have MRSTT > 00h, this test is not applicable.

© 2026 UNH–IOL

3. Configure the NVMe Host to issue a Set Features command for FID 22h to change the settings of Personality A (CHPS=1, a valid data buffer that does not contain any changes). This Personality must have MRSTT > 00h such that the change will be pending until the minimum required reset occurs.
4. Configure the host to issue a Set Features command for FID 22h to change the settings of Personality B (CHPS=1).
5. Configure the host to issue a Set Features command for FID 22h to change Personality A to a configuration value different from the one currently pending in Step 3.

Observable Results:

1. Verify the Set Features command in Step 3 completes with status code 'Successful Completion' (00h).
2. Verify that the PPSC bit is reported as '1' in Step 3.
3. Verify that the Set Features command in Step 4 is aborted with a status of 'Feature Not Changeable' (0Eh).
4. Verify that the Set Features command in Step 5 is aborted with a status of 'Feature Not Changeable' (0Eh).

Case 3: CDP Get Features All Personalities with Pending Change (OF-FYI, FYI)

Test Procedure:

1. Verify the controller supports CDP (FID 22h) via the Feature Identifiers Supported and Effects Log Page (LID 12h).
2. Issue a Get Log Page command for the Device Personalities Log Page (LID 1Dh). Identify a personality (e.g., Security Personality 01h) that has a Minimum Required Reset Type (MRSTT) greater than 00h (No Reset).
3. Configure the host to issue a Set Features command for FID 22h. Set CHPS='1' and provide a valid configuration data buffer for the identified personality (e.g., PERID=01h).
4. Issue a Get Features command for FID 22h with the specific PERID used in Step 3 (e.g., 01h).
5. Configure the host to issue a Get Features command for FID 22h with the PERID set to FFh (All Personalities) in Command Dword 13.
6. Perform the required reset to apply the settings and clear the pending state.

Observable Results:

1. Verify all commands complete with a Successful Completion (00h).
2. In the CQE Dword 0 for Step 5, verify the Pending Personality Settings Change (PPSC) bit (bit 09) is set to '1h'. In the same CQE Dword 0, verify the Personality Identifier (PERID) field (bits 07:00) matches the identifier of the specific personality targeted in Step 3 (e.g., returns 01h, not FFh).

© 2026 UNH–IOL

Group 14: Sanitize Per Namespace

Overview:
This section describes a method for performing conformance verification for NVMe products implementing the Sanitize Per Namespace capability introduced in Technical Proposal (TP) 4153. These tests assess a controller's ability to perform, report, and manage background sanitize operations directed at an individual namespace via the Sanitize Namespace command (opcode 8Ch), rather than the entire NVM subsystem.

© 2026 UNH–IOL

Test 14.1 – Sanitize Namespace Command (OF-FYI, FYI)

Purpose: To verify that an NVMe Controller correctly implements the capability to perform background sanitize operations on individual namespaces as introduced in Technical Proposal 4153.

References:

NVM Express Base Specification 2.3: 1.5, 3.2.1.6, 5.1.2, 5.1.12.1.33, 5.1, 8.1.24
Technical Proposal (TP) 4153: Sanitize Per Namespace

Resource Requirements:

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

Last Modification: July 30, 2026

Discussion: The Sanitize Per Namespace capability enables sanitize operations to be performed on a specific sanitization target (i.e., a single namespace) rather than requiring a sanitize of the entire NVM subsystem. These operations are initiated via the Sanitize Namespace command (opcode 8Ch) and are performed in the background.

Testing verifies that the controller accurately reports the progress and results of these operations through the Sanitize Namespace Status List log page (LID 7Fh) and namespace-specific updates to the Sanitize Status log page (LID 81h). Key reporting requirements include the use of the Namespace Data Erased (NDE) bit to indicate a namespace has been erased since its creation or last successful sanitize, and the Sanitization Target NSID (STNSID) field to identify the specific target.

The tests also validate the controller's enforcement of command processing restrictions. Prohibited Admin and I/O commands targeting a namespace undergoing sanitization must be aborted with the status Sanitize Namespace In Progress, while commands targeting a namespace that has failed a sanitize operation must be aborted with Sanitize Namespace Failed. Furthermore, the controller is verified for its management of concurrency, ensuring it correctly reports and enforces the Maximum Namespace Sanitize Operations In Progress (MNSOIP) limit. Finally, the testing ensures that Asynchronous Event Notifications (AENs) and Persistent Event Log (PEL) entries correctly identify the specific namespace target by including its NSID in the relevant fields.

Test Setup: See Appendix A.

Case 1: Delete Namespace Aborted with Sanitize Namespace in Progress (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects log page (LID 05h). If bit 0 of the Commands Supported and Effects data structure for the Sanitize Namespace command (opcode 8Ch) is cleared to 0b, this test is not applicable.
2. Check the OACS field to determine if the DUT supports Namespace Management. If the DUT does not support Namespace Management, this test is not applicable.
3. Perform a Sanitize Namespace command with the Sanitize Action (SANACT) field set to 100b (Crypto Erase).
4. Before command completion arrives for the Sanitize Namespace command, configure the NVMe Host to issue a Namespace Management command with the Select (SEL) field set to 1h (Delete) and an NSID to the namespace used in the Sanitize Namespace command in step 3.

Observable Results:

1. Verify that all commands in steps 1, 2, and 3 completed successfully.
2. Verify that the Namespace Management command in step 4 is aborted with 'Sanitize Namespace in Progress' (01h).

Case 2: Namespace Data Erased (NDE) (OF-FYI, FYI)

Test Procedure:

© 2026 UNH–IOL

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects log page (LID 05h). If bit 0 of the Commands Supported and Effects data structure for the Sanitize Namespace command (opcode 8Ch) is cleared to 0b, this test is not applicable.
2. Check the OACS field to determine that the DUT supports Namespace Management. If the DUT does not support Namespace Management, this test is not applicable.
3. Configure the NVMe Host to issue a Namespace Management command with the Select (SEL) field set to 0h (Create) and valid values for the attached data structure to the DUT to create a new namespace.
4. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status log page (LID 81h) specifying the NSID of the namespace created in step 3.
5. Configure the NVMe Host to write a known data pattern to the first LBA on the namespace created in step 3.
6. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status log page (LID 81h) specifying the NSID of the namespace created in step 3.

Observable Results:

1. Verify that Namespace Data Erased (NDE) bit (Bit 10 of the Sanitize Status (SSTAT) field, Bytes 03:02) is set to 1h in the Sanitize Status field returned from the Get Log Page command in step 4.
2. Verify that Namespace Data Erased (NDE) bit (Bit 10 of the Sanitize Status (SSTAT) field, Bytes 03:02) is cleared to 0h in the Sanitize Status field returned from the Get Log Page command in step 6.

Case 3: Sanitize Namespace - Overwrite Passes Completed (OPC) (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects log page (LID 05h). If bit 0 of the Commands Supported and Effects data structure for the Sanitize Namespace command (opcode 8Ch) is cleared to 0b, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status log page (LID 81h).

Observable Results:

1. Verify that all commands completed successfully.
2. Verify that the Sanitization Target NSID (STNSID) from the Get Log Page command in step 2 is set to the NSID of the namespace issued in the command.
3. Verify that the Overwrite Passes Completed (OPC) field returned from the Get Log Page command in step 2 is cleared to 0h.

Case 4: Sanitize Namespace - Sanitize Never Started (SOS 000b) (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects log page (LID 05h). If bit 0 of the Commands Supported and Effects data structure for the Sanitize Namespace command (opcode 8Ch) is cleared to 0b, this test is not applicable.
2. Configure the NVMe Host to issue a Namespace Management command with the Select (SEL) field set to 0h (Create) and valid values for the attached data structure to the DUT to create a new namespace.
3. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status log page (LID 81h) specifying the NSID of the namespace created in step 2.

Observable Results:

1. Verify that all the commands completed successfully.
2. Verify that the Sanitization Target NSID (STNSID) from the Get Log Page command in step 3 is set to the NSID of the namespace issued in the command.
3. Verify that the Sanitize Operation Status (SOS) field from the Get Log Page command in step 3 indicates status ‘Sanitize Never Started’ (0h).

© 2026 UNH–IOL

Case 5: Sanitize Namespace Concurrent Operations Limit Validation (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects log page (LID 05h). If bit 0 of the Commands Supported and Effects data structure for the Sanitize Namespace command (opcode 8Ch) is cleared to 0b, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. Record the Maximum Namespace Sanitize Operations In Progress (MNSOIP) field to determine the maximum concurrent operation limit. If MNSOIP is cleared to 0h, this test is not applicable.
3. Configure the NVMe Host to simultaneously issue valid Sanitize Namespace commands with the Sanitize Action (SANACT) field set to 100b targeting N = MNSOIP individual distinct attached namespaces.
4. While the Sanitize commands from the previous step are in progress, configure the NVMe Host to issue a Sanitize Namespace command (opcode 8Ch) targeting an (NN+1) distinct active namespace.

Observable Results:

1. Verify that all initial Sanitize Namespace commands in step 3 are completed successfully.
2. Verify that the Sanitize Namespace command in step 4 is aborted by the controller with a status code of ‘Request Exceeds Maximum Namespace Sanitize Operations In Progress’ (3Ch).

Case 6: Sanitize Namespace NSID Target Validation and Rejection Constraints (OF-FYI, M)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects log page (LID 05h). If bit 0 of the Commands Supported and Effects data structure for the Sanitize Namespace command (opcode 8Ch) is cleared to 0b, this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize Namespace command (opcode 8Ch) with the NSID field set to 0h and a valid Sanitize Action (SANACT) code (Crypto Erase 100b).
3. Configure the NVMe Host to issue a Sanitize Namespace command (opcode 8Ch) with the NSID field set to FFFFFFFFh and a valid SANACT code (Crypto Erase 100b).
4. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. Check the Namespace Management Supported (NMS) bit. If NMS is cleared to 0b, this test is not applicable.
5. Configure the NVMe host to issue a Namespace Management command specifying a valid NSID, a Select (SEL) field value of 0h (Create), and valid values for the attached data structure to the controller under test to create a new namespace.
6. Configure the NVMe Host to issue a Sanitize Namespace command (opcode 8Ch) targeting the NSID created in step 5.

Observable Results:

1. Verify that the commands sent in steps 1, 2, and 6 are aborted by the controller with a status code of ‘Invalid Namespace or Format’ (0Bh).

Case 7: Sanitize Namespace command Option and Rejection Constraints (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Commands Supported and Effects Log Page (LID 05h). If bit 0 of the Commands Supported and Effects Data Structure for the Sanitize Namespace command (opcode 8Ch) is cleared to 0b, this test is not applicable.
2. Configure the NVMe Host to issue a Sanitize Namespace command targeting an attached namespace with the Sanitize Action (SANACT) field set to 001b (Exit Failure Mode).
3. Configure the NVMe Host to issue a Sanitize Namespace command targeting an attached namespace with the SANACT field set to 101b (Exit Media Verification State).
4. Configure the NVMe Host to issue a Sanitize Namespace command targeting an attached namespace with the SANACT field set to 100b (Crypto Erase) and the Allow Unrestricted Sanitize Exit (AUSE) bit set to 1b.

© 2026 UNH–IOL

Observable Results:

1. Verify that the Sanitize Namespace command in step 2 is aborted by the controller with status code 'Sanitize Namespace Failed' (2Bh).
2. Verify that the Sanitize Namespace command in step 3 is aborted by the controller with status code 'Invalid Field in Command' (02h).
3. Verify that the Sanitize Namespace command in step 4 is aborted by the controller with status code 'Sanitize Namespace Failed' (2Bh).

Case 8: Memory Namespace Sanitize Data Zeroing and Action Equivalence (OF-FYI, FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the I/O Command Set data structure (CNS 1Ch) data structure. If bit 3 of the data structure is cleared to 0b, this test is not applicable.
2. Configure the NVMe Host to issue an Identify command to obtain the Identify Controller (CNS 01h) data structure. If bits 2:0 of SANICAP are cleared to 0b, this test is not applicable.
3. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status Log Page (LID 81h) with the Retain Asynchronous Event (RAE) bit cleared to 0b to clear any previous Sanitize asynchronous events.
4. Configure the NVMe Host to issue an Identify command to obtain the I/O Command Set specific Identify Namespace data structure (CNS 05h, CSI 03h). Record the Namespace Size (NSZE) field.
5. For each Sanitize Action (SANACT) supported by the DUT as reported in the Sanitize Capabilities (SANICAP) field, execute the following steps:
    a. Configure the NVMe Host to issue a Memory Write command with Starting Byte set to 0h and Write Length set to NSZE to fill a memory namespace with a known non-zero data pattern.
    b. Configure the NVMe Host to issue an Asynchronous Event Request command to the NVMe Controller.
    c. Configure the NVMe Host to issue a Sanitize command with the current SANACT code. If the action is Overwrite (011b), configure a non-zero Overwrite Pattern parameter.
    d. Configure the NVMe Host to issue a Get Log Page command for the Sanitize Status log page (LID 81h) with the Retain Asynchronous Event (RAE) bit set to 1b. Repeat this step periodically until the sanitize operation has completed, indicated by Sanitize Progress (SPROG) set to FFFFh.
    e. Configure the NVMe Host to issue a Read command with SB set to 0h and RL set to NSZE to the same memory namespace in step 5.a.

Observable Results:

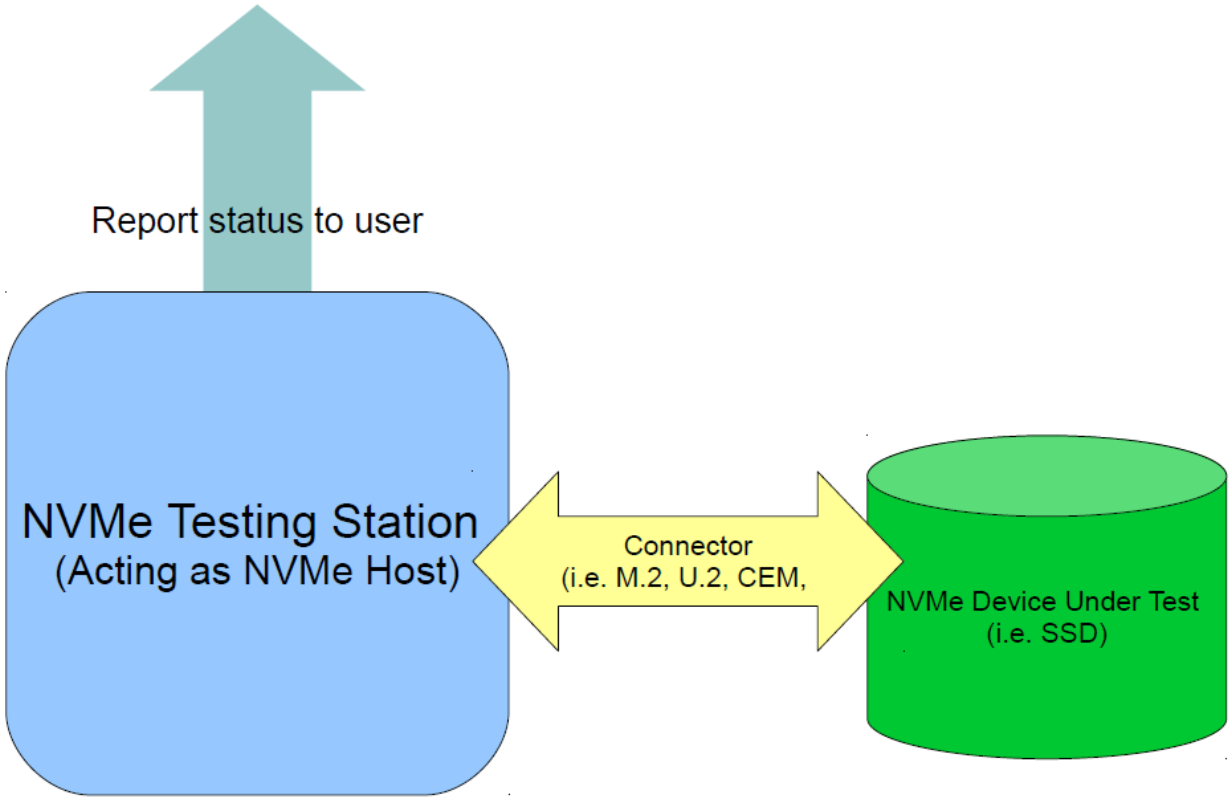
1. Verify all commands completed successfully.
2. Verify that for each supported sanitize action, there is a completion queue entry for the Asynchronous Event Request command following the sanitize operation's completion with Log Page Identifier set to 81h, Asynchronous Event Type set to 110b, and the Asynchronous Event Information indicates a status of 'Sanitize Operation Completed' (01h).
3. Verify that for every sanitize action, the data recorded in step 5.e consists entirely of 0h bytes for all logical blocks within the memory namespace.

© 2026 UNH–IOL

APPENDIX A: DEFAULT TEST SETUP

Except where otherwise specified, all tests will require the DUT to have one of the following default physical configuration at the beginning of each test case:

Test Setup for NVMe Device:



© 2026 UNH–IOL

Appendix B: Notes on Test Procedures

There are scenarios where in test procedures it is desirable to leave certain aspects of the testing procedure as general as possible. In these cases, the steps in the described test procedure may use placeholder values, or may intentionally use non–specific terminology, and the final determination of interpretation or choice of values is left to the discretion of the test technician. The following is an attempt to capture and describe all such instances used throughout the procedures.

Ports on Testing Station and Device Under Test	In general, any PCIe Port on the Testing Station or Device Under Test may be used as an interface with a test station or interoperability partner. There is *assumed* to be no difference in behavior, with respect to the protocols involved in this test suite, between any two PCIe ports on the Testing Station or Device Under Test. Hence, actual ports used may be chosen for convenience. However, it is recommended that the PCIe port used in the test configuration is recorded by the test technician.
Use of "various"	To maintain generality, some steps will specify that "various other values" (or the like) should be used in place of a given parameter. Ideally, all possible values would be tested in this case. However, limits on available time may constrain the ability of the test technician to attempt this. Given this, a subset of the set of applicable values must generally be used. When deciding how many values should be used, it should be noted that the more values that are tested, the greater the confidence of the results obtained (although there is a diminishing return on this). When deciding which specific values to use, it is generally recommended to choose them at pseudo–randomly yet deterministically. However, if there exist subsets of the applicable values with special significance, values from each subset should be attempted.

© 2026 UNH–IOL

Appendix C: Test Tools

The Tests described in this document can be performed using available IOL INTERACT NVMe Test Software available from UNH–IOL.

If using the PC Edition of the IOL INTERACT NVMe Test Software, UNH-IOL recommends using v24.0 or higher of the IOL INTERACT NVMe Test Software. This software is available at: https://www.iol.unh.edu/solutions/test-tools/interact.

If using the Teledyne-LeCroy edition of the IOL INTERACT NVMe Test Software, UNH-IOL recommends using v24.0 or higher of the IOL INTERACT NVMe Test Software. This software is available at: https://www.iol.unh.edu/solutions/test-tools/interact.
This should be used in conjunction with the latest Teledyne-LeCroy software available at: https://teledynelecroy.com

© 2026 UNH–IOL

APPENDIX D: NVME INTEGRATORS LIST REQUIREMENTS

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

References:

[1] NVMe Integrators List Policy Document

Resource Requirements:

- NVMe Host Platform and Device.

Last Modification: April 2, 2019

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

Test 1.1 Example Name (M) – Test is mandatory for all PCIe based products and does not apply to NVMe-oF products.

Test 2.1 Example Name (M, OF) – Test is mandatory for all products, including NVMe-oF products.

Test 3.1 Example Name (M, OF-IP) – Test is mandatory for all PCIe based products, and test is currently In Progress for NVMe-oF products.

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

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

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

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

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

© 2026 UNH–IOL