



LEGAL NOTICE:

© **Copyright 2008 to 2022 NVM Express®, Inc. ALL RIGHTS RESERVED.**

This Technical Proposal is proprietary to the NVM Express, Inc. (also referred to as “Company”) and/or its successors and assigns.

NOTICE TO USERS WHO ARE NVM EXPRESS, INC. MEMBERS: Members of NVM Express, Inc. have the right to use and implement this Technical Proposal subject, however, to the Member’s continued compliance with the Company’s Intellectual Property Policy and Bylaws and the Member’s Participation Agreement.

NOTICE TO NON-MEMBERS OF NVM EXPRESS, INC.: If you are not a Member of NVM Express, Inc. and you have obtained a copy of this document, you only have a right to review this document or make reference to or cite this document. Any such references or citations to this document must acknowledge NVM Express, Inc. copyright ownership of this document. The proper copyright citation or reference is as follows: “© 2008 to 2022 NVM Express, Inc. ALL RIGHTS RESERVED.” When making any such citations or references to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of NVM Express, Inc. Nothing contained in this document shall be deemed as granting you any kind of license to implement or use this document or the specification described therein, or any of its contents, either expressly or impliedly, or to any intellectual property owned or controlled by NVM Express, Inc., including, without limitation, any trademarks of NVM Express, Inc.

LEGAL DISCLAIMER:

THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN “**AS IS**” BASIS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, NVM EXPRESS, INC. (ALONG WITH THE CONTRIBUTORS TO THIS DOCUMENT) HEREBY DISCLAIM ALL REPRESENTATIONS, WARRANTIES AND/OR COVENANTS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, VALIDITY, AND/OR NONINFRINGEMENT.

All product names, trademarks, registered trademarks, and/or servicemarks may be claimed as the property of their respective owners.

The NVM Express® design mark is a registered trademark of NVM Express, Inc.

NVM Express Workgroup
c/o VTM, Inc.
3855 SW 153rd Drive
Beaverton, OR 97003
USA
info@nvmexpress.org

NVM Express® Technical Proposal (TP)

Technical Proposal ID	TP4136 Should to Shall Conversion Catch-Up
Revision Date	2022-11-27
Builds on Specification(s)	NVM Express Base Specification revision 2.0b NVM Command Set Specification revision 1.0b NVM Express Management Interface Specification revision 1.2b
References	NVMe-oF 1.0 ECN 005 NVMe 1.1 ECN 005 NVMe 1.2 ECN 002 NVMe 1.2 ECN 003 NVMe 1.2 ECN 004 NVMe 1.2 ECN 007 NVMe 1.2 ECN 008 NVMe 1.2.1 ECN 001 NVMe 1.3 ECN 002 NVMe 1.3 ECN 003 NVMe 1.3 ECN 004a NVMe 1.3 ECN 005 NVMe 1.3 ECN 006 NVMe 1.4 ECN 001 NVMe-MI 1.1 ECN 002

Technical Proposal Author(s)

Name	Company
Matt Goepfert, Curtis Ballard	HPE
Mike Allison	Samsung

Technical Proposal Overview

This technical proposal identifies instances of “should” statements that were introduced into the various NVMe specifications via ECNs and changes them to “shall” statements where appropriate. These “should” statements that are changing to “shall” statements are for behaviors that were strong recommendations during their initial incorporation and were intended to be converted into hard requirements at a later point.

Revision History

Revision Date	Change Description
2022.03.02	<ul style="list-style-type: none"> Initial draft
2022.03.09	<ul style="list-style-type: none"> Proposed change to SMART / Health Information log page text Added more placeholder discussion items to the document
2022.03.29	<ul style="list-style-type: none"> Added all remaining placeholder discussion items to the document
2022.04.14	<ul style="list-style-type: none"> Includes recommended changes from 4/14 TWG meeting with comments for all completed items marked as resolved
2022.05.12	<ul style="list-style-type: none"> Includes recommended changes from 5/12 TWG meeting with comments for all completed items marked as resolved Includes comments from Austin Bolen for all MI-related items to be discussed during next NVMe-MI meeting
2022.05.16	<ul style="list-style-type: none"> Completed walkthrough of all discussion items Includes changes based upon 5/16 NVMe-MI meeting discussion
2022.05.19	<ul style="list-style-type: none"> Added editorial fixes to impacted text in section 5.16.1 (added two missing words and a missing comma) Added editorial fix to impacted text in section 5.27.1.27 (capitalization correction) Added changes summary section Removed all items that required no change Removed all resolved comments Changed document structure to align with conventional TP format
2022.05.26	<ul style="list-style-type: none"> Made small correction in Revision History section (editorial fix is in section 5.27.1.27 – not section 5.27.1.26) Added editor's note to section 5.16.1 to make added comma stand out
2022.06.08	<ul style="list-style-type: none"> Changed document title to indicate Phase 3
2022.07.12	<ul style="list-style-type: none"> Includes editorial changes found by Samsung during member review
2022.07.14	<ul style="list-style-type: none"> Added TP name to Technical Proposal ID section Made small stylistic correction to last sentence of Figure 156
2022-11-27	<ul style="list-style-type: none"> Integrated

Description for Changes Document for NVMe 2.0b Base Specification

New Features/Feature Enhancements/Required Changes:

- **New requirement / incompatible change** in section 3.1.3.5
 - Attempting to create an I/O queue before initializing the I/O Completion Queue Entry Size (CC.IOCQES) and the I/O Submission Queue Entry Size (CC.IOSQES) shall cause a controller to abort a Create I/O Completion Queue command or a Create I/O Submission Queue command with a status code of Invalid Queue Size.
- **New requirement / incompatible change** in section 3.3.2.1.1
 - If the Fabrics Command Type (FCTYPE) field in a Fabrics command capsule is set to a reserved value, the command shall be aborted with a status code of Invalid Field in Command.
- **New requirement / incompatible change** in section 3.3.3.1
 - If the namespace identifier is used for the command, the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller shall abort the command with a status code of Invalid Field in Command, unless otherwise specified.
 - If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller shall abort the command with a status code of Invalid Field in Command, unless otherwise specified.
- **New requirement / incompatible change** in section 3.11
 - If the host submits a Firmware Commit command on the controller with a Commit Action of 011b, the downloaded firmware image shall replace the firmware image in the firmware slot.
- **New requirement / incompatible change** in section 5.2
 - If Asynchronous Event Request commands are outstanding when the controller is reset, then each of those commands is aborted and shall not return a CQE.
- **New requirement / incompatible change** in section 5.4
 - If the Physically Contiguous bit in Command Dword 11 of a Create I/O Completion Queue command is cleared to '0' and CAP.CQR is set to '1', then the controller shall return an error of Invalid Field in Command.
- **New requirement / incompatible change** in section 5.5
 - If the value specified in the Completion Queue Identifier (CQID) field in Command Dword 11 of a Create I/O Submission Queue command is 0h (i.e., the Admin Completion Queue), then the controller shall return an error of Invalid Queue Identifier.
 - If the value specified in the Completion Queue Identifier (CQID) field in Command Dword 11 of a Create I/O Submission Queue command is outside the range supported by the controller, then the controller shall return an error of Invalid Queue Identifier.
 - If the value specified in the Completion Queue Identifier (CQID) field in Command Dword 11 of a Create I/O Submission Queue command is within the range supported by the

controller and does not identify an I/O Completion Queue that has been created, then the controller shall return an error of Completion Queue Invalid

- **New requirement / incompatible change** in section 5.14
 - If there are I/O commands being processed for a namespace, then a Format NVM command that is submitted affecting that namespace may be aborted; if aborted, then a status code of Command Sequence Error shall be returned.
 - If a Format NVM command is in progress, then an I/O command that is submitted for any namespace affected by that Format NVM command may be aborted; if aborted, then a status code of Format in Progress shall be returned.
- **New requirement / incompatible change** in section 5.16.1
 - For log pages with a scope of NVM subsystem or controller, the controller shall abort commands that specify namespace identifiers other than 0h or FFFFFFFFh with a status code of Invalid Field in Command.
- **New requirement / incompatible change** in section 5.16.1.3
 - If the SMART / Health Information log page is not supported on a per namespace basis and if a namespace identifier other than 0h or FFFFFFFFh is specified by the host, then the controller shall abort the command with a status code of Invalid Field in Command.
- **New requirement / incompatible change** in section 5.17.2.2
 - For the Active Namespace ID list, the controller shall abort the Identify (CNS 02h) command with a status code of Invalid Namespace or Format if the NSID field is set to FFFFFFFEh or FFFFFFFFh.
- **New requirement / incompatible change** in section 5.17.2.9
 - For the Allocated Namespace ID list, the controller shall abort the Identify (CNS 10h) command with a status code of Invalid Namespace or Format if the NSID field is set to FFFFFFFEh or FFFFFFFFh.
- **New requirement / incompatible change** in section 5.17.2.10
 - For the Identify Namespace data structure for an Allocated Namespace ID, if the NSID field is set to FFFFFFFFh, then the controller shall abort the Identify (CNS 11h) command with a status code of Invalid Namespace or Format.
- **New requirement / incompatible change** in section 5.17.2.11
 - For the Namespace Attached Controller list, if the NSID field is set to FFFFFFFFh, then the controller shall abort the Identify (CNS 12h) command with a status code of Invalid Field in Command.
- **New requirement / incompatible change** in section 5.24
 - If the Firmware Commit command that established a pending firmware activation with reset condition returned a status code of: Firmware Activation Requires Controller Level Reset, Firmware Activation Requires Conventional Reset, or Firmware Activation Requires NVM Subsystem Reset, then the controller shall abort the Sanitize command with that same status code.
 - If the Firmware Commit command that established a pending firmware activation with reset condition completed successfully or returned a status code other than: Firmware Activation Requires Controller Level Reset, Firmware Activation Requires Conventional Reset, or Firmware Activation Requires NVM Subsystem Reset, then the controller shall abort the

Sanitize command with a status code of Firmware Activation Requires Controller Level Reset.

- **New requirement / incompatible change** in section 5.27.1.4
 - If a volatile write cache is not present, then a Get Features command specifying the Volatile Write Cache feature identifier shall abort with a status code of Invalid Field in Command.
- **New requirement / incompatible change** in section 5.27.1.26
 - For the Reservation Notification Mask feature, a Get Features (FID 82h) command that uses a namespace ID of FFFFFFFFh shall be aborted with status code of Invalid Field in Command.
- **New requirement / incompatible change** in section 5.27.1.27
 - For the Reservation Persistence feature, a Get Features (FID 83h) command that uses a namespace ID of FFFFFFFFh shall be aborted with a status code of Invalid Field in Command.
 - The Reservation Persistence feature shall not be saveable.
- **New requirement / incompatible change** in section 8.11
 - If the Namespace Management capability is supported, then the controller shall support the Namespace Attribute Changed asynchronous event.
- **New requirement / incompatible change** in section 8.19.7
 - If the Preempt Reservation Key (PRKEY) value is non-zero and there are no registrants whose reservation key matches the value of the PRKEY field, the controller shall return an error of Reservation Conflict.
- References
 - TP4136

Description for Changes Document for NVM Command Set 1.0b Specification

New Features/Feature Enhancements/Required Changes:

- **New requirement / incompatible change** in section 4.1.3.1
 - For the LBA Range Type feature, if the controller checks for LBA range overlap and the controller detects an LBA range overlap, then the controller shall return an error of Overlapping Range.
 - The default value for the LBA Range Type feature shall clear the Number of LBA Ranges field to 0h (i.e., one LBA Range is present).
- **New requirement / incompatible change** in section 4.1.5.5
 - For the Identify Namespace data structure for an Allocated Namespace ID, if the NSID field is set to FFFFFFFFh, then the controller shall abort the Identify (CNS 11h) command with a status code of Invalid Namespace or Format.
- **New requirement / incompatible change** in section 5.2.3
 - If the Reference Tag Check bit of the Protection Information Check (PRCHK) field is set to '1' and the namespace is formatted for Type 3 protection, then the controller may ignore the Initial Logical Block Reference Tag (ILBRT) and Expected Initial Logical Block Reference Tag (EILBRT) fields. If a command is aborted as a result of the Reference Tag

Check bit of the PRCHK field being set to '1', then that command shall be aborted with a status code of Invalid Protection Information.

- References
 - TP4136

Description for Changes Document for NVMe-MI 1.2b Specification

New Features/Feature Enhancements/Required Changes:

- **New requirement / incompatible change** in section 4.2.1.1
 - For the Pause Control Primitive, if the Command Slot Identifier (CSI) bit is set to '1', then the Management Endpoint shall transmit an Invalid Parameter Error Response with the Parameter Error Location (PEL) field indicating the CSI bit.
- References
 - TP4136

Markup Conventions:

- Black: Unchanged (however, hot links are removed)
- ~~Red Strikethrough~~: Deleted
- Blue: New
- Blue Highlighted: TBD values, anchors, and links to be inserted in new text.
- <Green Bracketed>: Notes to editor

Description of Specification Changes for NVMe 2.0b Base Specification

3 NVM Express Architecture

...

3.1.3.5 Offset 14h: CC – Controller Configuration

This property modifies settings for the controller. Host software shall set the Arbitration Mechanism Selected (CC.AMS), the Memory Page Size (CC.MPS), and the I/O Command Set Selected (CC.CSS) to valid values prior to enabling the controller by setting CC.EN to '1'. Attempting to create an I/O queue before initializing the I/O Completion Queue Entry Size (CC.IOCQES) and the I/O Submission Queue Entry Size (CC.IOSQES) ~~should~~ shall cause a controller to abort a Create I/O Completion Queue command or a Create I/O Submission Queue command with a status code of Invalid Queue Size.

<The above "should" statement was added with NVMe 1.2 ECN 007.>

...

3.3.2.1.1 Command Capsules

...

Figure 80: Fabrics Command Capsule – Submission Queue Entry Format

Bytes	Description
...	...

Figure 80: Fabrics Command Capsule – Submission Queue Entry Format

Bytes	Description
03:02	Command Identifier (CID): This field specifies a unique identifier for the command. The identifier shall be unique among all outstanding commands associated with a particular queue.
04	Fabrics Command Type (FCTYPE): This field specifies the Fabrics command transferred in the capsule. The Fabrics command types are defined in Figure 375. If this field is set to a reserved value, the command should shall be aborted with a status code of Invalid Field in Command.
39:05	Reserved
...	...

<The above “should” statement was added with NVMe-oF 1.0 ECN 005.>

...

3.3.3.1 Submission Queue Entry

...

Figure 87: Common Command Format

Bytes	Description
03:00	Command Dword 0 (CDW0): This field is common to all commands and is defined in Figure 86.
07:04	<p>Namespace Identifier (NSID): This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 138 and Figure 390 for commands that support the use of the value FFFFFFFFh in this field.</p> <p>Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified.</p> <p>If the namespace identifier is used for the command (refer to Figure 138), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should shall abort the command with a status code of Invalid Field in Command, unless otherwise specified.</p> <p>If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should shall abort the command with a status code of Invalid Field in Command, unless otherwise specified.</p>
11:08	Command Dword 2 (CDW2): This field is command specific Dword2.
...	...

<The above “should” statements were added with NVMe 1.3 ECN 005.>

...

3.11 Firmware Update Process

...

The process for a firmware update to be activated on a domain without a reset is:

1. The host issues a Firmware Image Download command to download the firmware image to a controller. There may be multiple portions of the firmware image to download, thus the offset for each portion of the firmware image being downloaded on that controller is specified in the Firmware Image Download command. The data provided in the Firmware Image Download command should conform to the Firmware Update Granularity indicated in the Identify Controller data structure or the firmware update may fail;

2. The host submits a Firmware Commit command on that controller with a Commit Action of 011b which specifies that the firmware image should be activated immediately without reset. The downloaded firmware image ~~should~~ shall replace the firmware image in the firmware slot. If no firmware image was downloaded since the last reset or Firmware Commit command, (i.e., the first step was skipped), then that controller shall verify and activate the firmware image in the specified slot. If that controller starts to activate the firmware image, any controllers affected by the new firmware image send a Firmware Activation Starting asynchronous event to the host if Firmware Activation Notices are enabled (refer to Figure 326):
 - a. The Firmware Commit command may also be used to activate a firmware image associated with a previously committed firmware slot;
3. The controller completes the Firmware Commit command. The following actions are taken in certain error scenarios:
 - a. If the firmware image is invalid, then the controller reports the appropriate error (e.g., Invalid Firmware Image);
 - b. If the firmware activation was not successful because a Controller Level Reset is required to activate this firmware, then the controller reports an error of Firmware Activation Requires Controller Level Reset and the firmware image is applied at the next Controller Level Reset;
 - c. If the firmware activation was not successful because an NVM Subsystem Reset is required to activate this firmware image, then the controller reports an error of Firmware Activation Requires NVM Subsystem Reset and the image is applied at the next NVM Subsystem Reset;
 - d. If the firmware activation was not successful because a Conventional Reset is required to activate this firmware, then the controller reports an error of Firmware Activation Requires Conventional Reset and the firmware image is applied at the next Conventional Reset; and
 - e. If the firmware activation was not successful because the firmware activation time would exceed the MTF value reported in the Identify Controller data structure, then the controller reports an error of Firmware Activation Requires Maximum Time Violation. In this case, to activate the firmware, the Firmware Commit command needs to be re-issued and the firmware image activated using a reset.

<The above “should” statement was added with NVMe 1.2 ECN 003.>

...

5 Admin Command Set

...

5.2 Asynchronous Event Request command

...

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 report to the host. If Asynchronous Event Request commands are outstanding when the controller is reset, then each of those commands is aborted and ~~should~~ shall not return a CQE.

<The above “should” statement was added with NVMe 1.3 ECN 006.>

...

5.4 Create I/O Completion Queue command

...

Figure 156: Create I/O Completion Queue – Command Dword 11

Bits	Description
...	...

Figure 156: Create I/O Completion Queue – Command Dword 11

Bits	Description
01	Interrupts Enabled (IEN): If set to '1', then interrupts are enabled for this Completion Queue. If cleared to '0', then interrupts are disabled for this Completion Queue.
00	<p>Physically Contiguous (PC): If set to '1', then the Completion Queue is physically contiguous and PRP Entry 1 (PRP1) is the address of a contiguous physical buffer. If cleared to '0', then the Completion Queue is not physically contiguous and PRP Entry 1 (PRP1) is a PRP List pointer. If this bit is cleared to '0' and CAP.CQR is set to '1', then the controller should shall return an error of Invalid Field in Command abort the command with a status code of Invalid Field in Command.</p> <p>If the:</p> <ul style="list-style-type: none"> • queue is located in the Controller Memory Buffer; • PC is cleared to '0'; and • CMBLOC.CQPDS is cleared to '0', <p>then the controller shall abort the command with a status code of Invalid Use of Controller Memory Buffer status.</p>

<The above “should” statement was added with NVMe 1.4 ECN 001.>

...

5.5 Create I/O Submission Queue command

...

Figure 160: Create I/O Submission Queue – Command Dword 11

Bits	Description
31:16	<p>Completion Queue Identifier (CQID): This field indicates the identifier of the I/O Completion Queue to utilize for any command completions entries associated with this Submission Queue. If the value specified:</p> <ol style="list-style-type: none"> is 0h (i.e., the Admin Completion Queue), then the controller should shall return an error of Invalid Queue Identifier abort the command with a status code of Invalid Queue Identifier; is outside the range supported by the controller, then the controller should shall return an error of Invalid Queue Identifier abort the command with a status code of Invalid Queue Identifier; or is within the range supported by the controller and does not identify an I/O Completion Queue that has been created, then the controller should shall return an error of Completion Queue Invalid abort the command with a status code of Completion Queue Invalid.
15:03	Reserved
...	...

<The first above “should” statement was existing black text. The second two above “should” statements were added with NVMe 1.3 ECN 005.>

...

5.14 Format NVM command

...

The Format NVM command may be aborted with a status code defined in this specification under circumstances defined by a security specification (e.g., invalid security state as specified in TCG Storage Interface Interactions specification). If there are I/O commands being processed for a namespace, then a Format NVM command that is submitted affecting that namespace may be aborted; if aborted, then a status code of Command Sequence Error ~~should~~ **shall** be returned. If a Format NVM command is in progress, then an I/O command that is submitted for any namespace affected by that Format NVM command may be

aborted; if aborted, then a status code of Format in Progress ~~should~~ shall be returned. Refer to section 5 for further information about restrictions on Admin Commands during Format NVM.

<The first above “should” statement was added with NVMe 1.4 ECN 001. The second above “should” statement was added with NVMe 1.3 ECN 006.>

...

5.16.1 Log Specific Information

...

For log pages with a scope of NVM subsystem or controller (as shown in Figure 202), the controller ~~should~~ shall abort commands that specify namespace identifiers other than 0h or FFFFFFFFh with a status code of Invalid Field in Command. Otherwise, ~~and~~ the rules for namespace identifier usage in Figure 87 apply.

<The above “should” statement was added with NVMe 1.3 ECN 004a.>

...

5.16.1.3 SMART / Health Information (Log Identifier 02h)

...

If the log page is not supported on a per namespace basis, ~~specifying~~ and:

- if a namespace identifier other than 0h or FFFFFFFFh is specified by the host, then the controller shall ~~should~~ abort the command with a status code of Invalid Field in Command; and
- ~~if a namespace identifier of 0h or FFFFFFFFh is specified by the host the controller does not abort the command,~~ then the controller returns the controller log page.

There is no namespace specific information defined in the SMART / Health Information log page in this revision of the specification, thus the controller log page and namespaces specific log page contain identical information.

<The above “should” statement was added with NVMe 1.1 ECN 005.>

...

5.16.1.7 Device Self-test (Log Identifier 06h)

...

Figure 213: Get Log Page – Self-test Result Data Structure

Description																																									
<p>Device Self-test Status: This field indicates the device self-test code and the status of the operation.</p> <p>Bits 7:4 indicates the Self-test Code value that was specified in the Device Self-test command that started the device self-test operation that this Self-test Result Data Structure describes.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Reserved</td> </tr> <tr> <td>1h</td> <td>Short device self-test operation</td> </tr> <tr> <td>2h</td> <td>Extended device self-test operation</td> </tr> <tr> <td>3h to Dh</td> <td>Reserved</td> </tr> <tr> <td>Eh</td> <td>Vendor specific</td> </tr> <tr> <td>Fh</td> <td>Reserved</td> </tr> </tbody> </table> <p>Bits 3:0 indicates the result of the device self-test operation that this Self-test Result Data Structure describes.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Operation completed without error</td> </tr> <tr> <td>1h</td> <td>Operation was aborted by a Device Self-test command</td> </tr> <tr> <td>2h</td> <td>Operation was aborted by a Controller Level Reset</td> </tr> <tr> <td>3h</td> <td>Operation was aborted due to a removal of a namespace from the namespace inventory</td> </tr> <tr> <td>4h</td> <td>Operation was aborted due to the processing of a Format NVM command</td> </tr> <tr> <td>5h</td> <td>A fatal error or unknown test error occurred while the controller was executing the device self-test operation and the operation did not complete</td> </tr> <tr> <td>6h</td> <td>Operation completed with a segment that failed and the segment that failed is not known</td> </tr> <tr> <td>7h</td> <td>Operation completed with one or more failed segments and the first segment that failed is indicated in the Segment Number field</td> </tr> <tr> <td>8h</td> <td>Operation was aborted for unknown reason</td> </tr> <tr> <td>9h</td> <td>Operation was aborted due to a sanitize operation</td> </tr> <tr> <td>Ah to Eh</td> <td>Reserved</td> </tr> <tr> <td>Fh</td> <td>Entry not used (does not contain a test result)</td> </tr> </tbody> </table> <p>Segment Number: This field indicates the segment number (refer to section 8.6) where the first self-test failure occurred. If Device Self-test Status field bits [3:0] are not set to 7h, then this field should be ignored by the host.</p> <p>Valid Diagnostic Information: This field indicates the diagnostic failure information that is reported.</p> <p>Bits 7:4 are reserved.</p> <p>Bit 3 (SC Valid): If set to '1', then the contents of Status Code field are valid. If cleared to '0', then the contents of the Status Code field are invalid.</p> <p>Bit 2 (SCT Valid): If set to '1', then the contents of the Status Code Type field are valid. If cleared to '0', then the contents of the Status Code Type field are invalid.</p> <p>Bit 1 (FLBA Valid): If set to '1', then the contents of the Failing LBA field are valid. If cleared to '0', then the contents of the Failing LBA field are invalid.</p> <p>Bit 0 (NSID Valid): If set to '1', then the contents of the Namespace Identifier field are valid. If cleared to '0', then the contents of the Namespace Identifier field are invalid.</p> <p>...</p>		Value	Definition	0h	Reserved	1h	Short device self-test operation	2h	Extended device self-test operation	3h to Dh	Reserved	Eh	Vendor specific	Fh	Reserved	Value	Definition	0h	Operation completed without error	1h	Operation was aborted by a Device Self-test command	2h	Operation was aborted by a Controller Level Reset	3h	Operation was aborted due to a removal of a namespace from the namespace inventory	4h	Operation was aborted due to the processing of a Format NVM command	5h	A fatal error or unknown test error occurred while the controller was executing the device self-test operation and the operation did not complete	6h	Operation completed with a segment that failed and the segment that failed is not known	7h	Operation completed with one or more failed segments and the first segment that failed is indicated in the Segment Number field	8h	Operation was aborted for unknown reason	9h	Operation was aborted due to a sanitize operation	Ah to Eh	Reserved	Fh	Entry not used (does not contain a test result)
Value	Definition																																								
0h	Reserved																																								
1h	Short device self-test operation																																								
2h	Extended device self-test operation																																								
3h to Dh	Reserved																																								
Eh	Vendor specific																																								
Fh	Reserved																																								
Value	Definition																																								
0h	Operation completed without error																																								
1h	Operation was aborted by a Device Self-test command																																								
2h	Operation was aborted by a Controller Level Reset																																								
3h	Operation was aborted due to a removal of a namespace from the namespace inventory																																								
4h	Operation was aborted due to the processing of a Format NVM command																																								
5h	A fatal error or unknown test error occurred while the controller was executing the device self-test operation and the operation did not complete																																								
6h	Operation completed with a segment that failed and the segment that failed is not known																																								
7h	Operation completed with one or more failed segments and the first segment that failed is indicated in the Segment Number field																																								
8h	Operation was aborted for unknown reason																																								
9h	Operation was aborted due to a sanitize operation																																								
Ah to Eh	Reserved																																								
Fh	Entry not used (does not contain a test result)																																								

5.17.2.2 Active Namespace ID list (CNS 02h)

A list of 1,024 namespace IDs is returned to the host containing active NSIDs in increasing order that are greater than the value specified in the Namespace Identifier (NSID) field of the command. The controller

should shall abort the command with a status code of Invalid Namespace or Format if the NSID field is set to FFFFFFFEh or FFFFFFFFh. The NSID field may be cleared to 0h to retrieve a Namespace List including the namespace starting with NSID of 1h. The data structure returned is a Namespace List (refer to section 4.4.2).

<The above “should” statement was added with NVMe 1.2 ECN 007.>

...

5.17.2.9 Allocated Namespace ID list (CNS 10h)

...

The controller **should shall** abort the command with a status code of Invalid Namespace or Format if the NSID field is set to FFFFFFFEh or FFFFFFFFh. The NSID field may be cleared to 0h to retrieve a Namespace List including the namespace starting with NSID of 1h. The data structure returned is a Namespace List (refer to section 4.4.2).

<The above “should” statement was added with NVMe 1.2 ECN 008.>

5.17.2.10 Identify Namespace data structure for an Allocated Namespace ID (CNS 11h)

...

If the specified namespace is an invalid NSID, then the controller shall abort the command with a status code of Invalid Namespace or Format. If the NSID field is set to FFFFFFFFh, then the controller **should shall** abort the command with a status code of Invalid Namespace or Format.

<The above “should” statement was added with NVMe 1.2.1 ECN 001.>

5.17.2.11 Namespace Attached Controller list (CNS 12h)

A Controller List (refer to section 4.4.1) of up to 2,047 controller identifiers is returned containing a controller identifier greater than or equal to the value specified in the Controller Identifier (CDW10.CNTID) field. The list contains controller identifiers of controllers that are attached to the namespace specified in the Namespace Identifier (NSID) field. If the NSID field is set to FFFFFFFFh, then the controller **should shall** abort the command with a status code of Invalid Field in Command.

<The above “should” statement was added with NVMe 1.3 ECN 006.>

...

5.24 Sanitize command

...

If the Firmware Commit command that established the pending firmware activation with reset condition returned a status code of:

- a) Firmware Activation Requires Controller Level Reset;
- b) Firmware Activation Requires Conventional Reset; or
- c) Firmware Activation Requires NVM Subsystem Reset,

then the controller **should shall** abort the Sanitize command with that same status code.

If the Firmware Commit command that established the pending firmware activation with reset condition completed successfully or returned a status code other than:

- a) Firmware Activation Requires Controller Level Reset;
- b) Firmware Activation Requires Conventional Reset; or
- c) Firmware Activation Requires NVM Subsystem Reset,

then the controller **should shall** abort the Sanitize command with a status code of Firmware Activation Requires Controller Level Reset.

<The above “should” statements were added with NVMe 1.3 ECN 006.>

...

5.27.1.4 Volatile Write Cache (Feature Identifier 06h), (Optional)

...

If a volatile write cache is not present, then a Set Features command specifying the Volatile Write Cache feature identifier shall abort with a status code of Invalid Field in Command, and a Get Features command specifying the Volatile Write Cache feature identifier **should shall** abort with a status code of Invalid Field in Command.

<The above “should” statement was added with NVMe 1.3 ECN 004a.>

...

5.27.1.26 Reservation Notification Mask (Feature Identifier 82h), (Optional²)

...

A Set Features command that uses a namespace ID other than FFFFFFFFh modifies the reservation notification mask for the corresponding namespace only. A Set Features command that uses a namespace ID of FFFFFFFFh modifies the reservation notification mask of all namespaces that are attached to the controller and that support reservations. A Get Features command that uses a namespace ID other than FFFFFFFFh returns the reservation notification mask for the corresponding namespace. A Get Features command that uses a namespace ID of FFFFFFFFh **should shall** be aborted with status code of Invalid Field in Command. If a Set Features command or a Get Features command attempts to access the Reservation Notification Mask on a namespace that does not support reservations or is invalid, then that command is aborted with status code of Invalid Field in Command.

<The above “should” statement was added with NVMe 1.3 ECN 006.>

...

5.27.1.27 Reservation Persistence (Feature Identifier 83h), (Optional³)

...

The PTPL state is contained in the Reservation Persistence **F**eature that is namespace specific. A Set Features command that uses the namespace ID FFFFFFFFh modifies the PTPL state associated with all namespaces that are attached to the controller and that support PTPL (i.e., support reservations). A Set Features command that uses a valid namespace ID other than FFFFFFFFh and corresponds to a namespace that supports reservations, modifies the PTPL state for that namespace. A Get Features command that uses a namespace ID of FFFFFFFFh **should shall** be aborted with a status code of Invalid Field in Command. A Get Features command that uses a valid namespace ID other than FFFFFFFFh and corresponds to a namespace that supports PTPL, returns the PTPL state for that namespace. If a Set Features command or a Get Features command using a namespace ID other than FFFFFFFFh attempts to access the PTPL state for a namespace that does not support this Feature Identifier, then the command is aborted with status code of Invalid Field in Command.

This **F**eature **should shall** not be saveable (refer to Figure 195).~~If this Feature is saveable, then the host should set the current value and the saved value to the same value.~~

<The above “should” statements were added with NVMe 1.3 ECN 006.>

...

8 Extended Capabilities

...

8.11 Namespace Management

...

If the Namespace Management capability is supported, then the controller:

- a) shall support the Namespace Management command and the Namespace Attachment command;
- b) shall set bit 3 to '1' in the OACS field (refer to Figure 275);
- c) ~~should~~ shall support the Namespace Attribute Changed asynchronous event (refer to Figure 146 and section 5.27.1.8); and
- d) may support Namespace Granularity (refer to the NVM Command Set Specification).

<The above "should" statement was added with NVMe 1.3 ECN 003.>

...

8.19.7 Preempting a Reservation or Registration

...

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:

- a) If the PRKEY field value is 0h, then the following occurs as an atomic operation:
 - all registrants other than the host that issued the command are unregistered;
 - the reservation is released; and
 - a new reservation is created of the type specified by the Reservation Type (RTYPE) field in the command for the host that issued the command as the reservation key holder;

or

- b) If the PRKEY value is non-zero, then registrants whose reservation key matches the value of the PRKEY field are unregistered. If the PRKEY value is non-zero and there are no registrants whose reservation key matches the value of the PRKEY field, the controller ~~should~~ shall ~~return an error of Reservation Conflict~~ abort the command with a status code of Reservation Conflict.

<The above "should" statement was added with NVMe 1.2 ECN 004.>

...

Description of Specification Changes for NVM Command Set 1.0b Specification

4 Admin Commands for the NVM Command Set

...

4.1.3.1 LBA Range Type (Feature Identifier 03h), (Optional)

...

Each entry in the LBA Range Type data structure is defined in Figure 83. The LBA Range feature is a set of 64 byte entries; the number of entries is indicated as a command parameter, the maximum number of entries is 64. The controller is not required to perform validation checks on any of the fields in this data structure. The LBA ranges should not overlap and may be listed in any order (e.g., ordering by LBA is not required). If the controller checks for LBA range overlap and the controller detects an LBA range overlap, then the controller ~~should~~ shall ~~return an error of Overlapping Range~~ abort the command with a status code of Overlapping Range.

...

The default value for this feature ~~should~~ shall clear the Number of LBA Ranges field to 0h (i.e., one LBA Range is present) and initialize the LBA Range Type data structure to contain a single entry with the:

- Type field cleared to 0h;
- Attributes field set to 1h;
- Starting LBA field cleared to 0h;
- Number of Logical Blocks field set to indicate the number of LBAs in the namespace; and

- GUID field cleared to 0h, or set to a globally unique identifier.

<The first above “should” statement was existing black text. The second above “should” statement was added with NVMe 1.2 ECN 002.>

...

4.1.5.5 Identify Namespace data structure for an Allocated Namespace ID (CNS 11h)

...

If the specified namespace is an invalid NSID, then the controller shall abort the command with a status code of Invalid Namespace or Format. If the NSID field is set to FFFFFFFFh, then the controller ~~should~~ shall abort the command with a status code of Invalid Namespace or Format.

<The above “should” statement was added with NVMe 1.2.1 ECN 001.>

...

5.2.3 Control of Protection Information Checking - PRCHK

...

- If the Reference Tag Check bit of the PRCHK field is set to ‘1’ and the namespace is formatted for Type 3 protection, then the controller:
 - should not compare the protection Information Reference Tag field to the computed reference tag; and
 - may ignore the ILBRT and EILBRT fields. If a command is aborted as a result of the Reference Tag Check bit of the PRCHK field being set to ‘1’, then that command ~~should~~ shall be aborted with a status code of Invalid Protection Information, ~~but may be aborted with a status code of Invalid Field in Command.~~

<The above “should” statement was added with NVMe 1.3 ECN 002.>

...

Description of Specification Changes for NVMe-MI 1.2b Specification

4 Message Servicing Model

...

4.2.1.1 Pause

The Pause Control Primitive is used to suspend response transmission and suspend the timeout waiting for packet, as defined in the MCTP Base Specification, for both Command Slots in a Management Endpoint. The CSI bit in a Pause Control Primitive is not used and shall be cleared to 0h. If the CSI bit is set to ‘1’, then the Management Endpoint ~~should~~ shall transmit an Invalid Parameter Error Response with the PEL field indicating the CSI bit.

<The above “should” statement was added with NVMe-MI 1.1 ECN 002.>

...