LEGAL NOTICE:

© Copyright 2007 to 2020 NVM Express™, Inc. ALL RIGHTS RESERVED.

This NVM Express revision 1.4 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 NVM Express revision 1.4 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: **"© 2007 to 2020 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

Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant's agreement. Copyright © 2014 to 2020 NVMe™ Corporation.

NVM Express Technical Proposal for New Feature

Technical Proposal ID	**4060 Multiple Controller Firmware Update**
Change Date	**2020-07-09**
Builds on Specification	**NVM Express 1.4**
References Specification	**NVMe-MI 1.1** **TP 4009 Domains and Partitions**

Technical Proposal Author(s)

Name	**Company**
Mike Allison, Jonathan Hughes, Nick Adams, Karl Heichelheim, Piotr Kwidzinski	Intel
Fred Knight	NetApp
Curtis Ballard	HPE
James C. Hatfield	Seagate

This proposal intends to update the process of a firmware update to a domain that contains multiple controllers by updating the specification to:

- State that overlapping downloads results in undefined behavior.
- Allow a domain to optionally report the detection of overlapping downloads to a host.

Revision History

Revision Date	**Change Description**
2020-003-03	Initial version
2020-03-31	Due to the Technical WG discussion and the agreement that simultaneous image downloads are just not needed in the ecosystem, decided a much simpler approach to just state it is undefined and allow a domain to optionally report the detection of a simultaneous image download on a Firmware Image Download command and Firmware Commit command.
2020-04-03	Using David Black suggestion – especially replacing the word “simultaneous” with “overlapping” as that is more appropriate.
2020-04-16	Adjusted table formatting.
2020-04-17	David Black provided a definition for the sequence of commands to download and commit a firmware image of Boot Partition. Added the definition to section 1.6 and used the definition to text.
2020-04-20	Editorial changes to clarify overlap is about the command sequence. Updated highlighting to new section references.
2020-05-21	MUD field to be cleared to 00b when detection not supported. Editorial changes. Approved for 30 day member review.
2020-05-26	Austin Bolen review: Fixed spelling and missing “detect” word in Figure TBD. Fixed capitalization, bolding of text, and use TBD in the bit assignment for the FRMW field.
2020-07-08	Integrated into the NVM Express Base Specification.
2020-07-09	Made sure new text used same shade of blue.

Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant’s agreement. Copyright © 2014 to 2020 NVMe™ Corporation.

Description for NVMe 1.4 Changes Document

This technical proposal adds the process of dealing with firmware downloads that overlap in time to:

- state that these result in undefined behavior;
- clarify the firmware download process utilizes the same controller for all Firmware Image Download commands and Firmware Commit command; and
- update the all Firmware Image Download commands and Firmware Commit command Dword 0 to return an optional status if the domain detects a host that overlap firmware downloads at the same time.

Description of Specification Changes

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

Modify portions of NVMe 1.4 as shown below:

Add a new definition to section 1.6 as defined below:

1.6.TBD firmware/boot partition image update command sequence

The sequence of one or more Firmware Image Download commands that download a firmware image or a boot partition image followed by a Firmware Commit command that commits that downloaded image to a firmware slot or a boot partition.

Modify a portion of section 5.11 as defined below:

5.11 Firmware Commit command

Note: This command was known in NVM Express revisions prior to revision 1.2 ~~1.0 and 1.1~~ as “Firmware Activate.”

The Firmware Commit command is used to modify the firmware image or Boot Partitions.

Modify a portion of section 5.11.1 as defined below:

5.11.1 Command Completion

Upon completion of the Firmware Commit command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.

For Firmware Commit commands that specify activation of a new firmware image at the next Controller Level Reset (i.e., the CA field was set to 001b or 010b) and complete with a status code value of 0h (i.e.,

Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant’s agreement. Copyright © 2014 to 2020 NVMe™ Corporation.

Success Completion), a Controller Level Reset initiated by any of the methods defined in section 7.3.2 activates the specified firmware.

If the controller detects overlapping firmware/boot partition image update command sequences (refer to section 1.6.TBD) of more than one firmware image and/or Boot Partition or the use of more than one controller and/or Management Endpoint to update a single firmware image, then the results of that detection are reported in Dword 0 of the completion queue entry as defined in Figure TBD. Refer to section 8.1 and section 8.13.2.

Figure TBD: Firmware Commit – Completion Queue Entry Dword 0

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31:02</td><td>Reserved</td></tr>
<tr><td>01:00</td><td>Multiple Update Detected (MUD): This field indicates if a controller detected overlapping firmware/boot partition image update command sequences of Boot Partitions and/or firmware images (refer to section 8.1 and section 8.13.2). If the SMUD bit in the Firmware Update field of the Identify Controller data structure is cleared to ‘0’, then this field shall be cleared to 00b.

This field is valid if the command is successful or aborted.

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>1</td><td>If set to ‘1’, then the controller detected an overlapping firmware/boot partition image update command sequence due to processing a command from a Management Endpoint. If cleared to ‘0’, then the controller did not detect an overlapping firmware/boot partition image update command sequence due to processing a command from a Management Endpoint.</td></tr>
<tr><td>0</td><td>If set to ‘1’, then the controller detected an overlapping firmware/boot partition image update command sequence due to processing a command from an Admin Submission Queue on a controller. If cleared to ‘0’, then the controller did not detect an overlapping firmware/boot partition image update command sequence due to processing a command from an Admin Submission Queue on a controller.</td></tr>
</table>
</td></tr>
</table>

Firmware Commit command specific status values are defined in Figure 176.

Figure 176: Firmware Commit – Command Specific Status Values

Value	Description
06h	**Invalid Firmware Slot:** The firmware slot indicated is invalid or read only. This error is indicated if the firmware slot exceeds the number supported.
07h	**Invalid Firmware Image:** The firmware image specified for activation is invalid and not loaded by the controller.
0Bh	**Firmware Activation Requires Conventional Reset:** The firmware commit was successful, however, activation of the firmware image requires a Conventional Reset. If an FLR or Controller Reset occurs prior to a Conventional Reset, the controller shall continue operation with the currently executing firmware image.
10h	**Firmware Activation Requires NVM Subsystem Reset:** The firmware commit was successful, however, activation of the firmware image requires an NVM Subsystem Reset. If any other type of Controller Level Reset occurs prior to an NVM Subsystem Reset, the controller shall continue operation with the currently executing firmware image.
11h	**Firmware Activation Requires Controller Level Reset:** The firmware commit was successful; however, the image specified does not support being activated without a Controller Level Reset. The image shall be activated at the next Controller Level Reset. This status code should be returned only if the Commit Action field in the Firmware Commit command is set to 011b (i.e., activate immediately).
12h	**Firmware Activation Requires Maximum Time Violation:** The image specified if activated immediately would exceed the Maximum Time for Firmware Activation (MTFA) value reported in the Identify Controller data structure (refer to Figure 247). To activate the firmware, the Firmware Commit command needs to be re-issued and the image activated using a reset.
13h	**Firmware Activation Prohibited:** The image specified is being prohibited from activation by the controller for vendor specific reasons (e.g., controller does not support down revision firmware).
14h	**Overlapping Range**: This error is indicated if the firmware image has overlapping ranges.

Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant’s agreement. Copyright © 2014 to 2020 NVMe™ Corporation.

Figure 176: Firmware Commit – Command Specific Status Values

Value	Description
1Eh	**Boot Partition Write Prohibited:** This error is indicated if a command attempts to modify a Boot Partition while locked (refer to section 8.13.3).

Modify a portion of section 5.12 as defined below:

5.12 Firmware Image Download command

The Firmware Image Download command is used to download all or a portion of an image for a future update to the controller. The Firmware Image Download command may be submitted while other commands on the Admin Submission Queue or I/O Submission Queues are outstanding. The Firmware Image Download command downloads a new image (in whole or in part) to the controller.

The image may be constructed of multiple pieces that are individually downloaded with separate Firmware Image Download commands. Each Firmware Image Download command includes a Dword Offset and Number of Dwords that specify a dword range. The host software should ensure that image pieces do not have dword ranges that overlap and that the NUMD field and OFST field meet the alignment and granularity requirements indicated in the FWUG field (refer to Figure 247). Firmware portions may be submitted out of order to the controller. Host software shall submit image portions in order when updating a Boot Partition. If ranges overlap, the controller may return an error of Overlapping Range.

The new firmware image is not activated as part of the Firmware Image Download command. Refer to section 8.1 for details on the firmware update process. The firmware update process does not modify the contents of Boot Partitions. Refer to section 0 for details on the Boot Partition update process.

Host software should not overlap command sequences that update Boot Partitions and/or firmware images ~~simultaneously~~ (refer to section 8.1 and section 8.13.2).

After downloading an image, host software issues a Firmware Commit command before downloading another image. Processing of the first Firmware Image Download command after completion of a Firmware Commit command shall cause the controller to discard all remaining portion(s), if any, of downloaded images. If a reset occurs between a firmware download and completion of the Firmware Commit command, then the controller shall discard all portion(s), if any, of downloaded images.

The Firmware Image Download command uses the Data Pointer, Command Dword 10, and Command Dword 11 fields. All other command specific fields are reserved.

Figure 177: Firmware Image Download – Data Pointer

Bits	Description
127:00	**Data Pointer (DPTR):** This field specifies the location where data should be transferred from. Refer to Figure 105 for the definition of this field.

Figure 178: Firmware Image Download – Command Dword 10

Bits	Description
31:00	**Number of Dwords (NUMD):** This field specifies the number of dwords to transfer for this portion of the firmware. This is a 0's based value. If the value specified in this field does not meet the requirement indicated by the FWUG field (refer to Figure 247), the firmware update may fail with status of Invalid Field in Command.

Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant's agreement. Copyright © 2014 to 2020 NVMe™ Corporation.

Figure 179: Firmware Image Download – Command Dword 11

Bits	Description
31:00	**Offset (OFST):** This field specifies the number of dwords offset from the start of the firmware image being downloaded to the controller. The offset is used to construct the complete firmware image when the firmware is downloaded in multiple pieces. The piece corresponding to the start of the firmware image shall have an Offset of 0h. If the value specified in this field does not meet the requirement indicated by the FWUG field (refer to Figure 247), the firmware update may fail with status of Invalid Field in Command.

Modify a portion of section 5.12.1 as defined below:

5.12.1 Command Completion

Upon completion of the Firmware Image Download command, the controller posts a completion queue entry to the Admin Completion Queue. Firmware Image Download command specific status values are defined in Figure 180.

Figure 180: Firmware Image Download – Command Specific Status Values

Value	Description
14h	**Overlapping Range:** This error is indicated if the firmware image has overlapping ranges. This error is indicated if the granularity or alignment of the firmware image downloaded does not conform to the Firmware Update Granularity field indicated in the Identify Controller data structure.

If the controller detects overlapping firmware/boot partition image update command sequences (refer to section 1.6.TBD) of more than one firmware image and/or Boot Partition or the use of more than one controller and/or Management Endpoint to update a single firmware image, then the results of that detection are reported in Dword 0 of the completion queue entry as defined in Figure TBD. Refer to section 8.1 and section 8.13.2.

Modify a portion of figure 247 in section 5.15.2.2 as defined below:

5.15.2.2 Identify Controller data structure (CNS 01h)

The Identify Controller data structure (refer to Figure 247) is returned to the host for this controller.

Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant's agreement. Copyright © 2014 to 2020 NVMe™ Corporation.

Figure 247: Identify – Identify Controller Data Structure

<table>
<tr><th>Bytes</th><th>O/M [1]</th><th>Description</th></tr>
<tr><td colspan="3">…</td></tr>
<tr><td>260</td><td>M</td><td>Firmware Updates (FRMW): This field indicates capabilities regarding firmware updates. Refer to section 8.1 for more information on the firmware update process.

~~Bits 7:5 are reserved.~~

~~Bit 4 if set to ‘1’ indicates that the controller supports firmware activation without a reset. If cleared to ‘0’, then the controller requires a reset for firmware to be activated.~~

~~Bits 3:1 indicate the number of firmware slots that the controller supports. This field shall specify a value from one to seven, indicating that at least one firmware slot is supported and up to seven maximum. This corresponds to firmware slots 1 through 7.~~

~~Bit 0 if set to ‘1’ indicates that the first firmware slot (i.e., slot 1) is read only. If cleared to ‘0’, then the first firmware slot (i.e., slot 1) is read/write. Implementations may choose to have a baseline read only firmware image.~~
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>7:6</td><td>Reserved</td></tr>
<tr><td>5</td><td>Support Multiple Update Detection (SMUD): If set to ‘1’ indicates that the controller is able to detect overlapping firmware/boot partition image update command sequences (refer to section 8.1 and section 8.13.2). If cleared to ‘0’, then the controller is not able to detect overlapping firmware/boot partition image update command sequences.</td></tr>
<tr><td>4</td><td>Firmware Activation Without Reset (FAWR): I~~i~~f set to ‘1’ indicates that the controller supports firmware activation without a reset. If cleared to ‘0’, then the controller requires a reset for firmware to be activated.</td></tr>
<tr><td>3:1</td><td>Number Of Firmware Slots (NOFS): This field indicates the number of firmware slots that the controller supports. This field shall specify a value from one to seven, indicating that at least one firmware slot is supported and up to seven maximum. This corresponds to firmware slots 1 through 7</td></tr>
<tr><td>0</td><td>First Firmware Slot Read Only (FFSRO): I~~i~~f set to ‘1’ indicates that the first firmware slot (i.e., slot 1) is read only. If cleared to ‘0’, then the first firmware slot (i.e., slot 1) is read/write. Implementations may choose to have a baseline read only firmware image.</td></tr>
</table></td></tr>
</table>

Modify a portion of section 8.1 as defined below:

8.1 Firmware Update Process

The process for a firmware update to be activated in a domain (refer to section 7.NEW <TP 4009>) by a reset is:

1. The host issues a Firmware Image Download command to download the firmware image to a ~~the~~ 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. After the firmware is downloaded to ~~the~~ that controller, the next step is for the host to submit a Firmware Commit command to that controller. The Firmware Commit command verifies that the last firmware image downloaded is valid and commits that image to the firmware slot indicated for future use. A firmware image that does not start at offset zero, contains gaps, or contains overlapping regions is considered invalid. A controller may employ additional vendor specific means (e.g., checksum, CRC, cryptographic hash or a digital signature) to determine the validity of a firmware image:

Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant’s agreement. Copyright © 2014 to 2020 NVMe™ Corporation.

a. The Firmware Commit command may also be used to activate a firmware image associated with a previously committed firmware slot;

3. The last step is to perform a reset that then causes the firmware image specified in the Firmware Slot field in the Firmware Commit command to be activated. The reset may be an NVM Subsystem Reset, Conventional Reset, Function Level Reset, or Controller Reset (CC.EN transitions from '1' to '0'):

   a. In some cases a Conventional Reset or NVM Subsystem Reset is required to activate a f~~F~~irmware image. This requirement is indicated by Firmware Commit command specific status (refer to section 5.11.1);

   and

4. After the reset has completed, host software re-initializes the controller. This includes re-allocating I/O Submission and Completion Queues. Refer to section 7.6.1.

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 ~~the~~ 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 image should be activated immediately without reset. The downloaded image should replace the image in the firmware slot. If no image was downloaded since the last reset or Firmware Commit command, (i.e., the first step was skipped), then ~~the~~ that controller shall verify and activate the image in the specified slot. If ~~the~~ that controller starts to activate the firmware, any controllers affected by the new firmware send a Firmware Activation Starting asynchronous event to the host if Firmware Activation Notices are enabled (refer to Figure 287):

   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 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, 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 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 MTFA 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 image activated using a reset.

If the controller transitions to the $D3_{cold}$ state (refer to the PCI Express Base Specification) after the submission of a Firmware Commit command that attempts to activate a firmware image and before the completion of that command, then the controller may resume operation with either the firmware image active at the time the Firmware Commit command was submitted or the firmware image that was activated by that command.

Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant's agreement. Copyright © 2014 to 2020 NVMe™ Corporation.

If the firmware is not able to be successfully loaded, then the controller shall revert to the firmware image present in the most recently activated firmware slot or the baseline read-only firmware image, if available, and indicate the failure as an asynchronous event with a Firmware Image Load Error.

If a host overwrites (i.e., updates) the firmware in the active firmware slot, then the previously active firmware image may no longer be available. As a result, any action (e.g., power cycling the controller) that requires the use of that firmware slot may instead use the firmware image that is currently in that firmware slot.

~~Host software should not update multiple firmware images or simultaneously.~~

Host software should not overlap firmware/boot partition image update command sequences (refer to section 1.6.TBD). During a firmware image update command sequence, if a Firmware Image Download command or a Firmware Commit command is submitted for another firmware/boot partition image update command sequence, the results of both that command and the in-progress firmware image update are undefined.

Host software should use the same controller or Management Endpoint (refer to the NVMe Management Interface specification) for all commands that are part of a firmware image update command sequence. If the commands for a single firmware/boot partition image update command sequence are submitted to more than one controller and/or Management Endpoint, the controller may abort the Firmware Commit command with Invalid Firmware Image status.

After downloading an image, host software issues a Firmware Commit command before downloading additional firmware images. Processing of the first Firmware Download command after completion of a Firmware Commit command shall cause the controller to discard remaining portions, if any, of downloaded images. If a reset occurs between a firmware download and completion of the Firmware Commit command, then the controller shall discard all portion(s), if any, of downloaded images.

Modify a portion of section 8.13.2 as defined below:

8.13.2 Writing to a Boot Partition

Boot Partition contents may be modified by the host using the Firmware Image Download and Firmware Commit commands while the controller is enabled (CC.EN set to ‘1’).

The process for updating a Boot Partition is:

1. The host issues a Firmware Image Download command to download the contents of the Boot Partition to a ~~the~~ controller. There may be multiple portions of the Boot Partition to download, thus the offset for each portion of the Boot Partition being downloaded is specified in the Firmware Image Download command. Host software shall send the Boot Partition image in order starting with the beginning of the Boot Partition;
2. Unlock Boot Partitions for writing (refer to section 8.13.3);
3. The host submits a Firmware Commit command on that controller with a Commit Action of 110b which specifies that the downloaded image replaces the contents of the Boot Partition specified in the Boot Partition ID field;
4. The controller completes the Firmware Commit command. The following actions are taken in certain error scenarios:
   a. If the firmware activation was not successful because the Boot Partition could not be written, then the controller reports an error of Boot Partition Write Prohibited;
5. (Optional) The host reads the contents of the Boot Partition to verify they are correct (refer to section 8.3.1). Host software updates the active Boot Partition ID by issuing a Firmware Commit command with a Commit Action of 111b; and
6. The host locks Boot Partition access to prevent further modification (refer to section 8.13.3).

If an internal error, reset, or power loss condition occurs while committing the downloaded image to a Boot Partition, the contents of the Boot Partition may contain the old contents, new contents, or a mixture of both. Host software should verify the contents of a Boot Partition before marking that Boot Partition active to ensure the active Boot Partition is stable.

Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant’s agreement. Copyright © 2014 to 2020 NVMe™ Corporation.

Host software should not read the contents of a Boot Partition while writing to the Boot Partition. The controller may return a combination of new and old data if the host attempts to perform a Boot Partition read operation while overwriting the contents.

Host software should not overlap firmware/boot partition image update command sequences (refer to section 1.6.TBD). During a boot partition image update command sequence, if a Firmware Image Download command or a Firmware Commit command is submitted for another firmware/boot partition image update command sequence, the results of both that command and the in-progress firmware image update are undefined.

Host software should use the same controller or Management Endpoint (refer to the NVMe Management Interface specification) for all commands that are part of a boot partition image update command sequence. If the commands for a single firmware/boot partition image update command sequence are submitted to more than one controller and/or Management Endpoint, the controller may abort the Firmware Commit command with Invalid Firmware Image status.

Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant's agreement. Copyright © 2014 to 2020 NVMe™ Corporation.