LEGAL NOTICE:

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

NVM Express Technical Proposal for New Feature

Technical Proposal ID	6031 – Progress Detect During Paused Process State
Change Date	2021.03.16
Builds on Specification	NVM Express Management Interface 1.1b
References	TP 6030 More Processing Required Time Reporting

Technical Proposal Author(s)

Name	Company
Myron Loewen	Intel
Austin Bolen	Dell

This Technical Proposal enables Management Controllers to determine if a paused command in process state has made sufficient progress to transmit a response. No new mechanisms are required, the existing Get State primitive will just report Transmit State instead of Process State when the processing completes.

The prior constraint of staying in Process State after completing processing if paused is modified to move a command into Transmit State, just as if not paused. Preventing the transition during pause was an arbitrary decision in prior versions of the specification that did not add value. This new behavior is consistent with the behavior when not paused to simplify the specification and eliminate some corner cases.

A paused command slot is still paused even after transitioning to the transmit state. No traffic is initiated by the command slot until an implicit or explicit resume occurs.

This TP is not backwards compatible for a Management Controller reading Get State with Command Initiated Auto Pause. Now these will enter transmit state instead of sticking in process state until resumed. It may also have been possible to trap a command in process state before, but there were race conditions that would have already required the Management Controller to handle an escape to transmit state.

Revision History

Revision Date	Change Description
2020.11.25	Initial Draft
2021.01.28	Incorporated all feedback: clarification that pausing MPR does not block processing, combining with TP 6030 text, and some text clean up
2021.02.08	New text for “Description for NVMe-MI Changes Document”. Feedback integrated for corner cases around pause/resume involving MPR.
2021.03.09	Integrated into the NVMe Management Interface Specification, Revision 1.1.
2021.03.16	Accepted all changes, removed all comments, and converted references/cross-references to text.

Description for NVMe-MI Changes Document

Feature Enhancement:

- Allow a Management Controller to be able to easily detect when processing of Command Messages that were paused in the Process state have completed. (mandatory)
- Clarify pause/resume primitives impact on More Processing Required responses in 4.1.2
- **New requirement / incompatible change** in section 4.2, 4.2.1.1, 4.2.1.2, 4.2.1.3, 4.2.1.5.
    - Allow a Command Slot that is paused in the Process state to transition to the Transmit state once command processing is complete.
- **New requirement / incompatible change** in section 4.2.1.2.
    - "If the Command Slot was paused and a More Processing Required Response has not yet been transmitted for the Command Message being processed, then the request-to-response timer shall be reset and restarted (refer to section 4.2.2.1 for details on the request-to-response time)."
- References:
    - NVMe-MI revision 1.1b sections 4.1.2, 4.2.1.1, 4.2.1.2, 4.2.1.3, 4.2.1.5.
    - Technical Proposal 6030.

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

Description of Specification Changes

<Note to editor:

1. Globally replace "Pause primitive" with "Pause Control Primitive".
2. Globally replace "Resume primitive" with "Resume Control Primitive".
3. Globally replace "Abort primitive" with "Abort Control Primitive".
4. Globally replace "Get State primitive" with "Get State Control Primitive".
5. Globally replace "Replay primitive" with "Replay Control Primitive".>

Modify section 3.1.1.1 as follows:

…

Refer to Appendix B for artificial messages and their corresponding Message Integrity Check values. See Section 4.2.1.5 for special requirements on how to construct the Response Message when the Management Controller issues a Replay Control Primitive~~of a Response Message~~ with a non-zero Response Replay Offset.

…

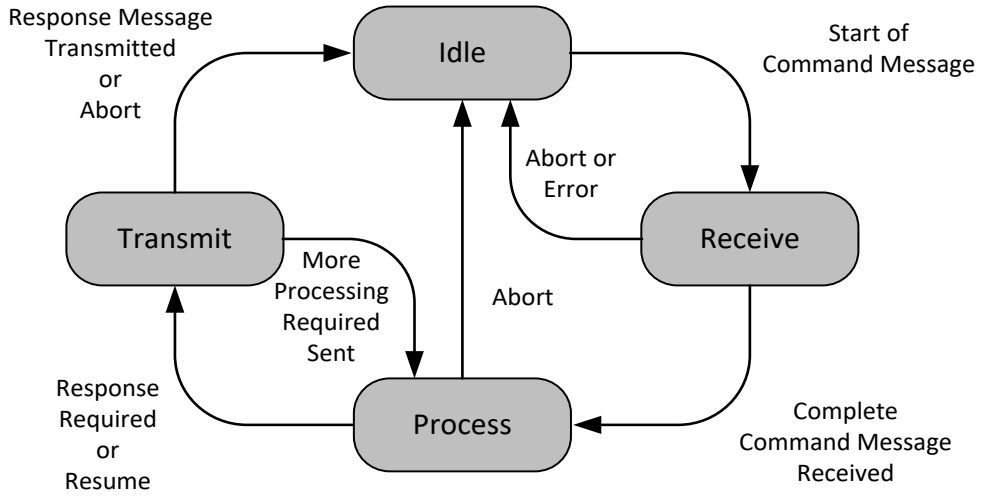
Modify Figure 26 (Response Message Status Values) as follows:

Figure 26: Response Message Status Values

Value	Description	Error Response Format Section
00h	**Success:** The command completed successfully.	4.1.2.1
01h	**More Processing Required:** The Command Message is in progress and requires more time to complete processing. When this Response Message Status is used in a Response Message, a subsequent Response Message contains the result of the Command Message. This Response Message Status shall not be sent more than once per ~~Request~~ Command Message, except for retransmission due to a Replay Control Primitive as described in section 4.2.1.5.	4.1.2.1

Modify one sentence in Section 4.2 Out-of-Band Message Servicing Model as shown below:

Figure 30: Command Servicing State Diagram



1. **Idle:** This is the default state of the command servicing state machine (e.g., following a reset). Command servicing transitions from Idle to the Receive state when the first MCTP packet of a Command Message is received (i.e., an MCTP packet with the SOM bit in the MCTP packet header set to '1' and the Message Type set to 4h).

2. **Receive:** The state when the first packet of a Command Message has been received and the message is being assembled and/or validated. Command servicing transitions from Receive to the Idle state when an Abort Control Primitive is received, an error is detected in message assembly (refer to section 3.2.1.1), or the Message Integrity Check fails (refer to section 3.1.1.1). Command servicing transitions from Receive state to the Process state when a Command Message is assembled and the message integrity check is successful.
3. **Process:** The state when a Command Message is processed. Processing of a Command Message consists of checking for errors with the Command Message and performing the actions specified by the Command Message or aborting the Command Message. Command servicing transitions from Process to the Transmit state when a Response Message is required to be sent (i.e., ~~the Pause Flag is cleared to '0' and either of the following are true: all~~ processing of the Command Message has completed or command processing is expected to exceed the corresponding transport binding specification response timeout). Command servicing transitions from the Process state to the Idle state due to an Abort Control Primitive (refer to section 4.2.1.3).
4. **Transmit:** The state in which a Response Message for the Command Message is transmitted to the Management Controller. Command servicing transitions from the Transmit to the Idle state once the entire NVMe-MI Message associated with the response to the Command Message has been transmitted on the physical medium or due to an Abort Control Primitive (refer to section 4.2.1.3).

   If, and only if, both the command servicing did not complete in the Process state and the Command Slot is not paused, then the Management Endpoint transmits a Response Message with status More Processing Required ~~and the command servicing transitions back to the Process state~~. If the Command Message requires more processing, then the Command Slot shall transition back to the Process state.

Modify Process paragraph in Section 4.2.1.1 Pause as shown below:

Process: The Pause Control ~~p~~Primitive sets the Pause Flag to '1' (refer to section 4.2.1.4) ~~causing the Command Slot to remain in the Process state until a Resume Control Primitive is received~~. The Pause Flag has no effect on the command processing in the Command Slot. Upon completion of command processing ~~Though command processing may complete~~, the Command Slot shall ~~not~~ transition to the Transmit state.

Modify Process paragraph in Section 4.2.1.2 Resume as shown below:

Process: ~~The Resume primitive allows a previously paused Command Slot to transition to the Transmit state once processing is complete and starts transmitting a response after responding to the Resume primitive.~~ If the Command Slot is paused and a More Processing Required Response has not yet been transmitted for the Command Message being processed, then the request-to-response timer shall be reset and restarted (refer to section 4.2.2.1 for details on the request-to-response time). The Pause Flag is cleared to '0' (refer to section 4.2.1.4).

Transmit: The Management Endpoint resumes transmission of the ~~r~~Response Message corresponding to the ~~c~~Command Message associated with the~~at~~ Command ~~s~~Slot after responding to the Resume Control ~~p~~Primitive. The Pause Flag is cleared to '0' (refer to section 4.2.1.4).

The Management Endpoint shall transmit a Control Primitive Response Message with success status after receiving the Resume Control ~~p~~Primitive.

Modify Process paragraph in Section 4.2.1.3 Abort as shown below:

…

Aborting a Command Message shall have no effect on the other Command Slot of the Management Endpoint, other Management Endpoints, or NVMe Controllers in the NVM Subsystem. Subsequent command servicing in the Command Slot is not affected by the Abort Control Primitive.

…

Process: The Abort primitive causes processing of the command in the Command Slot to be aborted:

a) If the Abort primitive was received before command processing started, the Management Endpoint discards the contents of the Command Slot and transitions to the Idle state. The Management Endpoint shall transmit a Success Response and the CPAS field set to 1h; or

b) If the Abort primitive was received while the command is being processed, the Management Endpoint discards the contents of the Command Slot and transitions to the Idle state. The Management Endpoint attempts to abort the command:

- If the command is aborted and had no effect on the NVM Subsystem, then the Management Endpoint shall transmit a Success Response and the CPAS field set to 1h; or
- If the Management Endpoint is not able to abort the command, then the Management Endpoint shall transmit a Success Response and set the CPAS field to 2h.~~; or~~
- ~~If the command has completed processing (e.g., the Management Endpoint is paused), then the Management Endpoint shall transmit a Success Response and the CPAS field is cleared to 0h.~~

Description of Specification Changes to TP 6030

Modify Process paragraph in Section 4.2.1.5 Replay as shown below:

…

Note that the Management Controller will need extensions to the MCTP Base Specification in its MCTP layer in order to ~~R~~replay a Response Message using a non-zero Response Replay Offset. No extensions to the MCTP Base Specification are needed to ~~R~~replay with Response Replay Offset equal to zero. For the case where a Management Controller chooses to use a non-zero Response Replay Offset, the MCTP Base Specification requires terminating message assembly for certain errors (i.e., receiving a packet with bad packet data integrity).

…

Process: If a More Processing Required Response has not been transmitted for the Command Message being processed, then a Success Response shall be transmitted with the RR bit cleared to '0'.

If a More Processing Required Response has been transmitted, then a Success Response shall be transmitted with the RR bit set to '1' and then the More Processing Required Response shall be retransmitted. The Management Endpoint shall update the More Processing Required Time field in the Response Message with the current worst-case amount of additional time that the Management Controller should wait for the Management Endpoint to complete processing of the Command Message. ~~The Replay primitive requests retransmission of the last response transmitted for the command in this Command Slot:~~

~~a) If a Response Message has not been transmitted for the Command Message (i.e., the slot never entered the Transmit state for the Command Message), then the Management Endpoint transmits a Response Message with success status and the RR bit cleared to '0'; or~~

~~b) If a Response Message has been transmitted for the Command Message (i.e., a Response Message was transmitted indicating that more processing was required), then the Management Endpoint transmits a Response Message with success status with the RR bit set to '1'. The Management Endpoint shall retransmit the response indicating that more processing is required. The Management Endpoint shall update the More Processing Required Time field in the Response Message with the current worst-case amount of additional time in 100 ms units that the~~

~~Management Controller should wait for the Management Endpoint to complete processing of the Command Message.~~

Transmit: The Management Endpoint stops transmitting response packets for the Command Slot and then transmits a Response Message with success status with the RR bit set to ‘1’. The Management Endpoint transmits a Response Message containing the packets starting at the packet offset specified in the Response Replay Offset field of the Replay Control Primitive after the Control Primitive Success Response. The Command Slot remains in the Transmit state until retransmission is complete.