



LEGAL NOTICE:

© **Copyright 2007 - 2018 NVM Express, Inc. ALL RIGHTS RESERVED.**

This NVM Express revision 1.3 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 1.3 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 - 2018 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.

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

NVM Express Technical Proposal for New Feature

Technical Proposal ID	4027
Change Date	2018-11-13
Builds on Specification	NVM Express 1.3c TP 4005a TP 4033

Technical Proposal Author(s)

Name	Company
Paul Suhler	Micron Technology
Mike Allison, Jonathan Hughes	Intel
Fred Knight	NetApp
Tom Friend	SK hynix
David Black	DellEMC

UUIDs for Vendor Specific Information

This technical proposal defines a mechanism for commands to select between vendor specific definitions of the information sent or received by that command.

The approach is for:

- the Identify command to return a list of UUIDs; and
- the Get Log Page, Get Features, Identify, and Set Features commands to specify an index into the list of UUIDs.

The organization which defines the vendor specific log page, feature, or other information will specify the UUID.

Revision History

Revision Date	Change Description
2018-01-31	Initial version
2018-02-14	Changes from 2018-02-08 meeting and discussions with Mike Allison
2018-02-15	Changes from recent comments: <ul style="list-style-type: none">Change terminology from VSOI to simpler OUI.Add requirement to support Commands Supported and Effects log page.Fleshed out error cases.
2018-02-28	Changes from reflector e-mail and 22 February Technical WG meeting.
2018-03-01	Changes from reflector e-mail and 1 March Technical WG meeting.
2018-04-10	Changes from e-mail and 5 April Technical WG meeting: <ul style="list-style-type: none">Add a theory of operations section in 8, and factor redundancy out of individual command descriptions.Loosen requirements for future flexibility.

2018-04-18	Changes from e-mail discussions: <ul style="list-style-type: none"> • Change from OUIs to UUIDs, and change terminology to “UUID”. • Allow UUID to be associated with a PCI Vendor ID.
2018-04-23	Changes from 19 April Technical WG meeting: <ul style="list-style-type: none"> • Delete term “Organization Identifier” • Renumbered TBD figures to match order of appearance.
2018-05-16	Changes from 25 April Technical WG meeting and review by Austin Bolen: <ul style="list-style-type: none"> • Deleted valid bit from UUID selector • Require compacted, not sparse, UUID list • Added example to section 8.TBD • Added requirement clarifying handling of commands not specifying a UUID. • Corrected typos.
2018-05-21	Merged TP 4041 into this TP. <ul style="list-style-type: none"> • Added Identify Command. • Corrected order of rows in Figure 106 changes. • Merged the set of Authors
2018-05-23	Corrected consistency of if/then statements. Moved new red text to blue. Corrected naming of Commands Supported and Effects data structure.
2018-06-05	Changes from offline review: <ul style="list-style-type: none"> • Updated to NVMe 1.3b and added additional unchanged text for clarity. • Merged changes from TPs 4005a and 4033. • Replaced “UUID Selector” with “UUID Index”. • Changed “versions of information” to “definitions of information” to avoid other connotations of “version”. • Rewrote explanation of how non-zero entries in the UUID list are contiguous.
2018-06-08	Changes from 7 June Technical WG meeting: <ul style="list-style-type: none"> • Improve explanation of the purpose of this functionality. • For each command’s Dword 14 description, clarified that support is by the controller. • For Get/Set Features, (a) clarified that UUID index is only specified for VS feature IDs, and (b) added a sentence explaining when a UUID index is not specified. • Added note to figure 90 to indicate which log pages might use UUID selection.
2018-08-30	Changes to satisfy member review comments: <ul style="list-style-type: none"> • Re-base to 1.3c • Intel: Apply UUIDs to Telemetry log pages. • Silicon Motion: <ul style="list-style-type: none"> ○ Add support footnote to Identify – CNS Values figure ○ Fix numbering of 8.TBD.3.2 • DellEMC: Only change UUID list at a controller level reset or firmware commit without reset. <ul style="list-style-type: none"> ○ Defined NVMe Invalid UUID and its use. ○ Required firmware commit with reset for certain conditions.
2018-10-30	Integration <ul style="list-style-type: none"> • Removed spaces from thousand separator in Figure TBD4.
2018-11-13	Ratified

Description for NVMe Changes Document

Vendor specific UUIDs (optional)

- Identify command returns a list of UUID Entries as specified by a new CNS value. Each UUID List Entry consists of a UUID and an Association field;
- Get Log Page – Commands Supported and Effects data structure has a new UUID Selection Supported bit to indicate that the command makes use of the UUID List;

Technical input submitted to the NVMe Express™ Workgroup is subject to the terms of the NVMe Express™ Participant’s agreement. Copyright © 2014-2018 NVMe™ Corporation.

- The following commands have a new field for an index into the UUID List, and the UUID Selection Supported bit is set:
 - Get Features;
 - Get Log Page;
 - Identify; and
 - Set Features;
- References:
 - NVMe 1.3c sections 5.13, 5.14, 5.15, 5.21, and 8.TBD; and
 - Technical Proposal 4027.

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.
<Green Bracketed>:	Notes to editor

Modify Portions of Section 5 as shown below:

5 Admin Command Set

...

5.11 Firmware Commit command

Note: This command was known in NVM Express revision 1.0 and 1.1 as “Firmware Activate.”

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

When modifying a firmware image, the Firmware Commit command verifies that a valid firmware image has been downloaded and commits that revision to a specific firmware slot. The host may select the firmware image to activate on the next Controller Level Reset as part of this command. The currently executing firmware revision may be determined from the Firmware Revision field of the Identify Controller data structure in Figure 223 or as indicated in the Firmware Slot Information log page. All controllers in the NVM subsystem share firmware image slots and the same firmware is applied to all controllers.

Activation of a firmware image may result in a change in controller behavior that is not expected by the host (e.g., an incompatible change in the UUID List (refer to section 8.TBD.2)). In this case, if the Commit Action field is set to 011b, then the controller shall abort the command with a status of Firmware Activation Requires Conventional Reset.

When modifying Boot Partitions, the host may select the Boot Partition to mark as active or replace. A Boot Partition may only be written when it is unlocked (refer to section 8.13).

...

5.13 Get Features command

The Get Features command retrieves the attributes of the Feature specified.

The Get Features command uses the Data Pointer, Command Dword 10, ~~and~~ Command Dword 11, ~~and~~ Command Dword 14 fields. All other command specific fields are reserved.

Figure 82: Get Features – Data Pointer

Bit	Description
127:00	Data Pointer (DPTR): This field specifies the start of the data buffer. Refer to Figure 11 for the definition of this field. If no data structure is used as part of this specified feature, then this field is ignored.

Figure 83: Get Features – Command Dword 10

Bit	Description												
31:11	Reserved												
10:08	Select (SEL): This field specifies which value of the attributes to return in the provided data:												
	<table><tr><th>Select</th><th>Description</th></tr><tr><td>000b</td><td>Current</td></tr><tr><td>001b</td><td>Default</td></tr><tr><td>010b</td><td>Saved</td></tr><tr><td>011b</td><td>Supported capabilities</td></tr><tr><td>100b to 111b</td><td>Reserved</td></tr></table>	Select	Description	000b	Current	001b	Default	010b	Saved	011b	Supported capabilities	100b to 111b	Reserved
	Select	Description											
	000b	Current											
	001b	Default											
	010b	Saved											
	011b	Supported capabilities											
100b to 111b	Reserved												
Refer to section 5.13.1 for details on the value returned in each case.													
The controller indicates in bit 4 of the Optional NVM Command Support field of the Identify Controller data structure in Figure 111 whether this field is supported.													
If a Get Features command is received with the Select field set to 010b (i.e., saved) and the controller does not support the Feature Identifier being saved or does not currently have any saved values, then the controller shall treat the Select field as though it was set to 001b (i.e., default.)													
07:00	Feature Identifier (FID): This field specifies the identifier of the Feature for which to provide data.												

If the controller supports selection of a UUID by the Get Features command (refer to Figure 128 and refer to section 8.TBD) and the controller supports selection of a UUID for the specified vendor specific feature identifier (refer to Figure 128), then Command Dword 14 is used to specify a UUID Index value (refer to Figure TBD1). If the controller does not support selection of a UUID by the Get Features command or the controller does not support selection of a UUID for the specified vendor specific feature identifier, then Command Dword 14 does not specify a UUID Index value.

Figure TBD1: Get Features – Command Dword 14

Bit	Description
31:07	Reserved
06:00	UUID Index: Refer to Figure TBD7.

Figure 84 describes the Feature Identifiers whose attributes may be retrieved using Get Features. The definition of the attributes returned and associated format is specified in the section indicated.

...

5.14 Get Log Page command

The Get Log Page command returns a data buffer containing the log page requested.

The Get Log Page command uses the Data Pointer, Command Dword 10, Command Dword 11, Command Dword 12, ~~and~~ Command Dword 13, and Command Dword 14 fields. All other command specific fields are reserved.

There are mandatory and optional Log Identifiers defined in Figure 90 and Figure 91. If a Get Log Page command is processed that specifies a Log Identifier that is not supported, then the controller should abort the command with status Invalid Field in Command.

The controller indicates if it supports the Log Page Offset and extended Number of Dwords (32 bits rather than 12 bits) in the Log Page Attributes field of the Identify Controller data structure. If extended data is not supported, then bits 27:16 of the Number of Dwords Lower field specify the Number of Dwords to transfer.

...

Figure 89: Get Log Page – Command Dword 13

Bit	Description
31:00	Log Page Offset Upper (LPOU): This field specifies the upper 32 bits of the log page offset. Refer to the Log Page Offset Lower definition.

If the controller supports selection of a UUID by the Get Log Page command (refer to Figure 90 and to section 8.TBD), then Command Dword 14 is used to specify a UUID Index value (refer to Figure TBD2).

Figure TBD2: Get Log Page – Command Dword 14

Bit	Description
31:07	Reserved
06:00	UUID Index: Refer to Figure TBD7.

5.14.1 Log Specific Information

Figure 90 and Figure 91 define the Log pages that may be retrieved with the Get Log Page command and the scope of the information that is returned in those Log pages.

Log pages that indicate a scope of NVM subsystem return information that is global to the NVM subsystem. Log pages that indicate a scope of controller return information that is specific to the controller that is processing the command. Log pages that indicate a scope of Namespace return information that is specific to the specified namespace. For log pages that indicate multiple scopes, the namespace identifier that is specified determines which information is returned. The definition of any individual field within a Log page may indicate a different scope that is specific to that individual field.

For Log Pages with a scope of NVM subsystem or controller (as shown in Figure 90 and Figure 91), the controller should abort commands that specify namespace identifiers other than 0h or FFFFFFFFh with status Invalid Field in Command. Otherwise the rules for namespace identifier usage in Figure 11 apply.

Figure 1: Get Log Page – Log Page Identifiers

Log Identifier	O/M	Scope	Description	Reference Section
00h	Reserved			
01h	M	Controller	Error Information	5.14.1.1
02h	M	Controller ¹	SMART / Health Information	5.14.1.2
	O	Namespace ²		
03h	M	NVM subsystem	Firmware Slot Information	5.14.1.3
04h	O	Controller	Changed Namespace List	5.14.1.4
05h	O	Controller	Commands Supported and Effects	5.14.1.5
06h	O	NVM subsystem	Device Self-test ³	5.14.1.6
07h	O	Controller	Telemetry Host-Initiated ³	5.14.1.7
08h	O	Controller	Telemetry Controller-Initiated ³	5.14.1.8
09h – 6Fh	Reserved			
70h	Discovery (refer to the NVMe over Fabrics specification)			
71h – 7Fh	Reserved for NVMe over Fabrics			
80h – BFh	I/O Command Set Specific			
C0h – FFh	Vendor specific ³			
KEY: O/M definition: O = Optional, M = Mandatory				
NOTES:				
1. For namespace identifiers of 0h or FFFFFFFFh				
2. For namespace identifiers other than 0h or FFFFFFFFh				
3. Selection of a UUID may be supported. Refer to section 8.TBD.				

Figure 2: Get Log Page – Log Page Identifiers, NVM Command Set Specific

Log Identifier	O/M	Scope	Description	Reference Section
80h	O	Controller	Reservation Notification	5.14.1.9.1
81h	O	NVM subsystem	Sanitize Status	5.14.1.9.2
82h – BFh	Reserved			
KEY: O/M definition: O = Optional, M = Mandatory				

...

5.14.1.5 Commands Supported and Effects (Log Identifier 05h)

...

Modify a portion of Figure 97 (Get Log Page – Commands Supported and Effects Data Structure) as shown below:

Bits	Description										
34:19 31:20	Reserved										
19	UUID Selection Supported: If set to '1', then the controller supports selection of a UUID by this command (refer to section 8.TBD). If cleared to '0', then the controller does not support selection of a UUID by this command.										
18:16	Command Submission and Execution (CSE): This field defines the command submission and execution recommendations for the associated command. <table><tr><th>Value</th><th>Definition</th></tr><tr><td>000b</td><td>No command submission or execution restriction</td></tr><tr><td>001b</td><td>Command may be submitted when there is no other outstanding command to the same namespace and another command should not be submitted to the same namespace until this command is complete</td></tr><tr><td>010b</td><td>Command may be submitted when there is no other outstanding command to any namespace and another command should not be submitted to any namespace until this command is complete</td></tr><tr><td>011b – 111b</td><td>Reserved</td></tr></table>	Value	Definition	000b	No command submission or execution restriction	001b	Command may be submitted when there is no other outstanding command to the same namespace and another command should not be submitted to the same namespace until this command is complete	010b	Command may be submitted when there is no other outstanding command to any namespace and another command should not be submitted to any namespace until this command is complete	011b – 111b	Reserved
Value	Definition										
000b	No command submission or execution restriction										
001b	Command may be submitted when there is no other outstanding command to the same namespace and another command should not be submitted to the same namespace until this command is complete										
010b	Command may be submitted when there is no other outstanding command to any namespace and another command should not be submitted to any namespace until this command is complete										
011b – 111b	Reserved										
15:05	Reserved										
...											

<Editor's Note: The above TBD bit number should be assigned when the TP is sent for ratification. That will resolve a race condition with other TPs requesting new bits in this log page.>

...

5.15 Identify Command

5.15.1 Identify command overview

The Identify command returns a data buffer that describes information about the NVM subsystem, the controller or the namespace(s). The data structure is 4096 bytes in size.

...

The Identify command uses the Data Pointer, ~~and Command Dword 10~~, and Command Dword 14 fields. All other command specific fields are reserved.

Figure 106: Identify – Data Pointer

Bit	Description
127:00	Data Pointer (DPTR): This field specifies the start of the data buffer. Refer to Figure 11 for the definition of this field. If using PRPs, this field shall not be a pointer to a PRP List as the data buffer may not cross more than one page boundary.

Figure 107: Identify – Command Dword 10

Bit	Description
31:16	Controller Identifier (CNTID): This field specifies the controller identifier used as part of some Identify operations. If the field is not used as part of the Identify operation, then host software shall clear this field to 0h for backwards compatibility (0h is a valid controller identifier). Controllers that support the Namespace Management capability (refer to section 8.12) shall support this field.
15:08	Reserved
07:00	Controller or Namespace Structure (CNS): This field specifies the information to be returned to the host. Refer to Figure 108.

If the controller supports selection of a UUID by the Identify command (refer to section 8.TBD), then Command Dword 14 is used to specify a UUID Index value (refer to Figure TBD3).

Figure TBD3: Identify – Command Dword 14

Bit	Description
31:07	Reserved
06:00	UUID Index: Refer to Figure TBD7.

The data structure returned is based on the Controller or Namespace Structure (CNS) field as shown in Figure 108. If there are fewer entries to return for the data structure indicated based on CNS value, then the unused portion of the list is zero filled. If a controller does not support a CNS value, then it shall abort the command with a status of Invalid Field in Command.

Note: The CNS field was specified as a one bit field in revision 1.0 and as a two bit field in revision 1.1. Host software should only issue CNS values defined in revision 1.0 to controllers compliant with revision 1.0. Host software should only issue CNS values defined in revision 1.1 to controllers compliant with revision 1.1. The results of issuing other CNS values to controllers compliant with revision 1.0 or 1.1, respectively, are indeterminate.

The Identify Controller data structure and Identify Namespace data structure include several identifiers. The format and layout of these identifiers is described in section 7.10.

Figure 3: Identify – CNS Values

CNS Value	O/M ¹	Definition	NSID ²	CNTID ³	Reference Section
Active Namespace Management					
00h	M	Identify Namespace data structure for the specified NSID or the common namespace capabilities. ⁶	Y	N	5.15.2
01h	M	Identify Controller data structure for the controller processing the command. ⁶	N	N	5.15.3
02h	M	Active Namespace ID list.	Y	N	5.15.4
03h	M	Namespace Identification Descriptor list for the specified NSID.	Y	N	5.15.5

CNS Value	O/M ¹	Definition	NSID ²	CNTID ³	Reference Section
04h to 0Fh		Reserved			

Controller and Namespace Management					
10h	O ⁴	Allocated Namespace ID list.	Y	N	5.15.6
11h	O ⁴	Identify Namespace data structure for the specified allocated NSID.	Y	N	5.15.7
12h	O ⁴	Controller identifier list of controllers attached to the specified NSID.	Y	Y	5.15.8
13h	O ⁴	Controller identifier list of controllers that exist in the NVM subsystem.	N	Y	5.15.9
14h	O ⁵	Primary Controller Capabilities data structure for the specified primary controller.	N	Y	5.15.10
15h	O ⁵	Secondary Controller list of controllers associated with the primary controller processing the command.	N	Y	5.15.11
17h	O	A UUID List (refer to Figure TBD4) is returned to the host.	N	N	5.15.TBD
16h 18h to 1Fh		Reserved			
Future Definition					
20h to FFh		Reserved			
NOTES: 1. O/M definition: O = Optional, M = Mandatory. 2. The CDW1.NSID field is used: Y = Yes, N = No. 3. The CDW10.CNTID field is used: Y = Yes, N = No. 4. Mandatory for controllers that support the Namespace Management capability (refer to section 8.12). 5. Mandatory for controllers that support Virtualization Enhancements (refer to section 8.5). 6. Selection of a UUID may be supported. Refer to section 8.TBD.					

<Editor's Note: The above CNS-TBD value should be assigned when the TP is sent for ratification. That will resolve a race condition with another TP that is also requesting a new CNS value.>

...

5.15.2 Identify Controller data structure (CNS 01h)

The Identify Controller data structure (refer to Figure 111) is returned to the host for this controller.

...

Figure 111 Identify – Identify Controller Data Structure

...		
99:96	M	<p>Controller Attributes (CTRATT): This field indicates attributes of the controller. Bits 34:6 31:10 are reserved.</p> <p>Bit 9 (UUID List): If set to '1', then the controller supports reporting of a UUID List (refer to Figure TBD4). If cleared to '0', then the controller does not support reporting of a UUID List (refer to section 8.TBD).</p> <p>Bit 5 (Predictable Latency Mode): If set to '1' then the controller supports Predictable Latency Mode (refer to section 8.18). If cleared to '0' then the controller does not support Predictable Latency Mode.</p> <p>Bit 4 (Endurance Groups): If set to '1', then the controller supports Endurance Groups (refer to section 8.17). If cleared to '0', then the controller does not support Endurance Groups.</p> <p>Bit 3 (Read Recovery Levels): If set to '1', then the controller supports Read Recovery Levels (refer to section 8.16). If cleared to '0', then the controller does not support Read Recovery Levels.</p> <p>Bit 2 (NVM Sets): If set to '1', then the controller supports NVM Sets (refer to section 4.9). If cleared to '0', then the controller does not support NVM Sets.</p> <p>Bit 1 (Non-Operational Power State Permissive Mode): If set to '1', then the controller supports host control of whether the controller may temporarily exceed the power of a non-operational power state for the purpose of executing controller initiated background operations in a non-operational power state (i.e., Non-Operational Power State Permissive Mode supported). If cleared to '0', then the controller does not support host control of whether the controller may exceed the power of a non-operational state for the purpose of executing controller initiated background operations in a non-operational state (i.e., Non-Operational Power State Permissive Mode not supported). Refer to section 5.21.1.17.</p> <p>Bit 0 if set to '1' then the controller supports a 128-bit Host Identifier. Bit 0 if cleared to '0' then the controller does not support a 128-bit Host Identifier.</p>
...		

<Editor's Note: The above TBD bit number should be assigned when the TP is sent for ratification. That will resolve a race condition with other TPs requesting new bits in this structure.>

<Editor's Note: Bits 2 – 5 were incorporated not in 1.3c, but in 1.NEXTc. They are included here for clarity.>

...

Figure 116: Secondary Controller Entry

Bytes	Description
1:0	Secondary Controller Identifier (SCID): This field indicates the Controller Identifier of the secondary controller described by this entry.
3:2	Primary Controller Identifier (PCID): This field indicates the Controller Identifier of the associated primary controller.
4	<p>Secondary Controller State (SCS): This field indicates the state of the secondary controller.</p> <p>Bits 7:1 are reserved.</p> <p>Bit 0 if set to '1' then the controller is in the Online state. Bit 0 if cleared to '0' then the controller is in the Offline state.</p>
7:5	Reserved
9:8	Virtual Function Number (VFN): If the secondary controller is an SR-IOV VF, this field indicates its VF Number, where VF Number > 0, and VF Number is no larger than the total number of VFs indicated by the TotalVFs register (refer to Single Root I/O Virtualization and Sharing Specification) in the PF's SR-IOV Extended Capability structure. If the secondary controller is not an SR-IOV VF, then this field is cleared to zero.

Bytes	Description
11:10	Number of VQ Flexible Resources Assigned (NVQ): This field indicates the number of VQ Flexible Resources currently assigned to the indicated secondary controller.
13:12	Number of VI Flexible Resources Assigned (NVI): This field indicates the number of VI Flexible Resources currently assigned to the indicated secondary controller.
31:14	Reserved

5.15. TBD UUID List (CNS 17h)

The format of the UUID List is defined in Figure TBD4. Each UUID List entry is either 0h, the NVMe Invalid UUID, or a valid UUID. Valid UUIDs are those which are non-zero and are not the NVMe Invalid UUID (refer to section 8.TBD).

If bit 9 (UUID List) is set to '1' in the Controller Attributes (CTRATT) field in the Identify Controller data structure (refer to Figure 111), then:

- The UUID List shall contain at least one valid UUID (refer to section 8.TBD);
- The UUID 1 field shall contain a non-zero value; and
- A UUID field cleared to 0h indicates the end of the UUID List.

The list may be in any order.

Figure TBD4: UUID List

Bytes	Description
31:00	Reserved
63:32	UUID 1: This field contains the first UUID List Entry in the list.
95:64	UUID 2: This field contains the second UUID List Entry in the list, if present, otherwise cleared to 0h.
...	...
4063:4032	UUID 126: This field contains the last non-zero UUID List Entry in the list, if present, otherwise cleared to 0h.
4095:4064	UUID 127: This field shall be cleared to 0h.

The format of a UUID List Entry is defined in Figure TBD5.

Figure TBD5 UUID List Entry

Bytes	Description																	
00	UUID List Entry Header:																	
	<table><tr><th>Bits</th><th>Description</th></tr><tr><td>7:2</td><td>Reserved</td></tr><tr><td>1:0</td><td>Identifier Association: This field indicates whether the UUID is associated with a vendor.</td></tr><tr><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>00b</td><td>No association reported.</td></tr><tr><td>01b</td><td>The UUID is associated with the vendor reported in the PCI Vendor ID field of the Identify Controller Data Structure (refer to Figure 111).</td></tr><tr><td>10b</td><td>The UUID is associated with the vendor reported in the PCI Subsystem Vendor ID field of the Identify Controller data structure.</td></tr><tr><td>11b</td><td>Reserved</td></tr></table></td></tr></table>	Bits	Description	7:2	Reserved	1:0	Identifier Association: This field indicates whether the UUID is associated with a vendor.	<table><tr><th>Value</th><th>Description</th></tr><tr><td>00b</td><td>No association reported.</td></tr><tr><td>01b</td><td>The UUID is associated with the vendor reported in the PCI Vendor ID field of the Identify Controller Data Structure (refer to Figure 111).</td></tr><tr><td>10b</td><td>The UUID is associated with the vendor reported in the PCI Subsystem Vendor ID field of the Identify Controller data structure.</td></tr><tr><td>11b</td><td>Reserved</td></tr></table>	Value	Description	00b	No association reported.	01b	The UUID is associated with the vendor reported in the PCI Vendor ID field of the Identify Controller Data Structure (refer to Figure 111).	10b	The UUID is associated with the vendor reported in the PCI Subsystem Vendor ID field of the Identify Controller data structure.	11b	Reserved
	Bits	Description																
	7:2	Reserved																
	1:0	Identifier Association: This field indicates whether the UUID is associated with a vendor.																
	<table><tr><th>Value</th><th>Description</th></tr><tr><td>00b</td><td>No association reported.</td></tr><tr><td>01b</td><td>The UUID is associated with the vendor reported in the PCI Vendor ID field of the Identify Controller Data Structure (refer to Figure 111).</td></tr><tr><td>10b</td><td>The UUID is associated with the vendor reported in the PCI Subsystem Vendor ID field of the Identify Controller data structure.</td></tr><tr><td>11b</td><td>Reserved</td></tr></table>	Value	Description	00b	No association reported.	01b	The UUID is associated with the vendor reported in the PCI Vendor ID field of the Identify Controller Data Structure (refer to Figure 111).	10b	The UUID is associated with the vendor reported in the PCI Subsystem Vendor ID field of the Identify Controller data structure.	11b	Reserved							
	Value	Description																
	00b	No association reported.																
01b	The UUID is associated with the vendor reported in the PCI Vendor ID field of the Identify Controller Data Structure (refer to Figure 111).																	
10b	The UUID is associated with the vendor reported in the PCI Subsystem Vendor ID field of the Identify Controller data structure.																	
11b	Reserved																	
15:01	Reserved																	
31:16	UUID: This field contains a 128-bit Universally Unique Identifier (UUID) as specified in RFC 4122. Refer to section 7.10.6.																	

...

5.21 Set Features command

The Set Features command specifies the attributes of the Feature indicated.

The Set Features command uses the Data Pointer, 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.

Figure 4: Set Features – Data Pointer

Bit	Description
127:00	Data Pointer (DPTR): This field specifies the start of the data buffer. Refer to Figure 11 for the definition of this field. If using PRPs, this field shall not be a pointer to a PRP List as the data buffer may not cross more than one page boundary. If no data structure is used as part of the specified feature, then this field is not used.

Figure 127: Set Features – Command Dword 10

Bit	Description
31	Save (SV): This field specifies that the controller shall save the attribute so that the attribute persists through all power states and resets. The controller indicates in bit 4 of the Optional NVM Command Support field of the Identify Controller data structure in Figure 111 whether this field is supported. If the Feature Identifier specified in the Set Features command is not saveable by the controller and the controller receives a Set Features command with the Save bit set to one, then the command shall be aborted with a status of Feature Identifier Not Saveable.
30:08	Reserved
07:00	Feature Identifier (FID): This field indicates the identifier of the Feature that attributes are being specified for.

If the controller supports selection of a UUID by the Set Features command (refer to [Figure 128](#) and refer to section [8.TBD](#)) and the controller supports selection of a UUID for the specified vendor specific feature identifier (refer to [Figure 128](#)), then Command Dword 14 is used to specify a UUID Index value (refer to [Figure TBD6](#)). If the controller does not support selection of a UUID by the Set Features command or the controller does not support selection of a UUID for the specified vendor specific feature identifier, then Command Dword 14 does not specify a UUID Index value.

Figure TBD6: Set Features – Command Dword 14

Bit	Description
31:07	Reserved
06:00	UUID Index: Refer to Figure TBD7 .

5.21.1 Feature Specific Information

[Figure 128](#) defines the Features that may be configured with Set Features and retrieved with Get Features. [Figure 129](#) defines Features that are specific to the NVM Command Set. Some Features utilize a memory buffer to configure or return attributes for a Feature, whereas others only utilize a Dword in the command or completion queue entry. Feature values that are not persistent across power cycles and resets are restored to their default values as part of a controller reset operation. The default value for each Feature is vendor specific and set by the manufacturer unless otherwise specified; it is not changeable. For more information on Features, including default, saveable, and current value definitions, refer to section [7.8](#).

There may be commands in execution when a Feature is changed. The new settings may or may not apply to commands already submitted for execution when the Feature is changed. Any commands submitted to a Submission Queue after a Set Features is successfully completed shall utilize the new settings for the associated Feature. To ensure that a Feature applies to all subsequent commands, commands being processed should be completed prior to issuing the Set Features command.

Figure 5: Set Features – Feature Identifiers

Feature Identifier	O/M ⁶	Persistent Across Power Cycle and Reset ²	Uses Memory Buffer for Attributes	Description
00h				Reserved
01h	M	No	No	Arbitration
02h	M	No	No	Power Management
03h	O	Yes	Yes	LBA Range Type
04h	M	No	No	Temperature Threshold
05h	M	No	No	Error Recovery
06h	O	No	No	Volatile Write Cache
07h	M	No	No	Number of Queues
08h	NOTE 5	No	No	Interrupt Coalescing
09h	NOTE 5	No	No	Interrupt Vector Configuration
0Ah	M	No	No	Write Atomicity Normal
0Bh	M	No	No	Asynchronous Event Configuration
0Ch	O	No	Yes	Autonomous Power State Transition
0Dh	O	No ³	No ⁴	Host Memory Buffer
0Eh	O	No	Yes	Timestamp
0Fh	O	No	No	Keep Alive Timer
10h	O	Yes	No	Host Controlled Thermal Management
11h	O	No	No	Non-Operational Power State Config
12h – 77h				Reserved
78h – 7Fh		Refer to the NVMe Management Interface Specification for definition.		
80h – BFh				Command Set Specific (Reserved)
C0h – FFh				Vendor Specific ^{1, 7}
<p>NOTES:</p> <ol style="list-style-type: none"> 1. The behavior of a controller in response to an inactive namespace ID to a vendor specific Feature Identifier is vendor specific. 2. This column is only valid if the feature is not saveable (refer to section 7.8). If the feature is saveable, then this column is not used and any feature may be configured to be saved across power cycles and reset. 3. The controller does not save settings for the Host Memory Buffer feature across power states and reset events, however, host software may restore the previous values. Refer to section 8.9. 4. The feature does not use a memory buffer for Set Features, but it does use a memory buffer for Get Features. Refer to section 8.9. 5. The feature is mandatory for NVMe over PCIe. This feature is not supported for NVMe over Fabrics. 6. O/M: O = Optional, M = Mandatory. 7. Selection of a UUID may be supported. Refer to section 8.TBD. 				

...

8 Features

...

Insert the following in section 8:

8.TBD UUIDs for Vendor Specific Information

8.TBD.1 UUIDs for VS Information Introduction

Several commands send or receive information that contains fields described as Vendor Specific or that is specified by a command field containing a value in a vendor specific range. Examples include the Set Features command, which may specify a vendor specific feature identifier, and the Identify command, which may retrieve a data structure having a vendor specific area.

The vendor specific information may have different definitions (e.g., a vendor specific log page identifier with the contents of the page defined differently by different entities, such as an NVM subsystem vendor and an NVM subsystem customer). By associating each definition of the information with a UUID specified by the defining entity, a command is able to specify the particular definition of the information.

A command specifies a particular definition of the information by specifying an index into a list of UUIDs supported by the controller. The NVMe Invalid UUID is defined for use to replace a previously valid UUID without reorganizing the list of UUIDs, as such reorganizations could change the UUID values specified by indexes into the UUID list.

NVM subsystem vendors and customers communicate (by means outside the scope of this specification) the UUID used for each definition of the information.

8.TBD.2 UUIDs for VS Information Requirements

A UUID list is a list of non-zero UUID values, terminated by a zero UUID value. Each non-zero UUID value may be either a valid UUID or the NVMe Invalid UUID. The NVMe Invalid UUID is the hexadecimal value FFFFFFFF_FFFFFFFF_7FFFFFFF_FFFFFFFFh. A valid UUID is any non-zero value other than the NVMe Invalid UUID.

If a command supports selection of a UUID, then the UUID Selection Supported bit in the Commands Supported and Effects data structure for that command (refer to Figure 97) shall be set to '1'. If a command does not support selection of a UUID, then the UUID Selection Supported bit shall be cleared to '0'.

If the UUID Selection Supported bit is set to '1' for one or more commands, then the UUID List bit in the Controller Attributes field shall be set to '1' (refer to Figure 111), and the controller shall support reporting of a UUID List (refer to Figure TBD4).

If a command supports selection of a UUID, then that command contains a UUID Index field (refer to Figure TBD7).

Figure TBD7: UUID Index Field

Bit	Description
6:0	UUID Index: If this field is set to a non-zero value, then the value of this field is the index of a UUID in the UUID List (refer to Figure TBD4) that is used by the command. If this field is cleared to 0h, then no UUID index is specified.

If the UUID Index field specifies a valid UUID (i.e., the UUID Index field is set to a non-zero value and the UUID at that index indicates a valid UUID) (refer to **section 5.15.TBD**), then the controller shall process the command using the vendor specific information specified by that UUID. If the UUID Index field is cleared to 0h, then the command does not specify a UUID.

If no UUID is specified by the command, then the controller shall process the command, returning vendor specific information.

The controller shall abort the command with Invalid Field in Command status if:

- The controller does not support the UUID specified by the UUID Index for the specified information;
- The UUID specified by the UUID Index is cleared to 0h; or
- The UUID specified by the UUID Index is the NVMe Invalid UUID.

If a firmware image is activated that has a UUID List in which an entry is different from that of the previously-active firmware image, then a host that is unaware of the change may issue a command with the UUID index value for that entry. Such a command may produce unexpected results because the UUID specified by that UUID Index has changed. To avoid this, vendors should follow the following revision guidelines for UUID lists when constructing firmware images that support UUID selection:

- Add UUIDs that are not supported in prior firmware image revisions to the end of the UUID List in subsequent firmware image revisions;
- Remove UUIDs that are supported in prior firmware image revisions by replacing the UUID with the NVMe Invalid UUID in the same entry in the UUID list in subsequent firmware image revisions;
- Do not replace the NVMe Invalid UUID with a valid UUID in the same UUID list entry in subsequent firmware image revisions; and
- Do not shorten or remove the UUID list in subsequent firmware image revisions.

In these guidelines, the terms “prior” and “subsequent” refer to a linear sequence of firmware versions (e.g., based on the date and time of the construction of the downloadable firmware image).

Following these guidelines prevents the host from inadvertently specifying the wrong UUID because there is at most one valid UUID for each entry in the UUID list. Hence a command that specifies a UUID Index either specifies the intended UUID or is aborted because that entry in the UUID list is empty or contains the NVMe Invalid UUID.

The controller shall require a reset to activate a downloaded firmware image (refer to **section 5.11**) if the downloaded image reports a UUID list with at least one slot in which a valid UUID replaces the NVMe Invalid UUID or a different valid UUID in the existing image. All controllers that are affected by the UUID list change caused by activation of a downloaded firmware image shall be reset as part of activating that downloaded firmware image.

The above requirements for a reset to activate a downloaded firmware image do not require the controller to directly compare the UUID lists in the current and downloaded firmware images. For example, a vendor could use a vendor-specific major.minor firmware image revision numbering system (e.g., 3.5, 4.1) where all downloadable firmware images with the same major revision number follow the above guidelines. In that scenario, the controller is able to meet these reset requirements by requiring a reset if the downloaded firmware image and the currently executing firmware have different major revision numbers.

8.TBD.3 UUIDs for VS Information Examples

This section includes examples of the use of UUIDs to select vendor specific information.

8.TBD.3.1 Vendor Specific Log Page Example

If entity C and entity V create different definitions for a vendor specific log page having the same log page identifier (e.g., D0h), then each assigns a UUID to distinguish their definition (e.g., entity V assigns UUID V and entity C assigns UUID C).

A controller supporting both definitions of the log page:

- a) Sets the UUID List bit to '1' in the CTRATT field of the Identify Controller data structure (refer to Figure 111);
- b) Sets the UUID Selection Supported bit to '1' in the Commands Supported and Effects data structure (refer to Figure 97) corresponding to the Get Log Page command; and
- c) Reports both UUID V and UUID C in the UUID list (refer to Figure TBD4).

A host requesting the log page defined by entity C:

- 1) Determines the index of UUID C in the UUID list;
- 2) Sets the Log Page Identifier field of the Get Log page command to D0h; and
- 3) Sets the UUID Index field of the Get Log Page command to the index of UUID C.

A host requesting the log page defined by entity V:

- 1) Determines the index of UUID V in the UUID list;
- 2) Sets the Log Page Identifier field of the Get Log Page command to D0h; and
- 3) Sets the UUID Index field of the Get Log Page command to the index of UUID V.

A host not specifying the definition of the log page clears the UUID Index field to 0h. The selection of the log page definition returned by the controller is vendor specific (e.g., the controller may select any definition for the returned data).

8.TBD.3.2 Vendor Specific Feature Example

If entity C and entity V create different definitions for a vendor specific feature having the same Feature Identifier (e.g., F1h), then each assigns a UUID to distinguish their definitions (e.g., entity V assigns UUID V and entity C assigns UUID C).

A controller supporting both definitions of the feature for the Get Features command:

- a) Sets the UUID List bit to '1' in the CTRATT field of the Identify Controller data structure (refer to Figure 111);
- b) Sets the UUID Selection Supported bit to '1' in the Commands Supported and Effects data structure (refer to Figure 97) corresponding to the Get Features command; and
- c) Reports both UUID V and UUID C in the UUID list (refer to Figure TBD4).

A host retrieving the attributes of the feature defined by entity C:

- 1) Determines the index of UUID C in the UUID list;
- 2) Sets the Feature Identifier field of the Get Features command to F1h; and
- 3) Sets the UUID Index field of the Get Features command to the index of UUID C.

A host retrieving the attributes of the feature defined by entity V:

- 1) Determines the index of UUID V in the UUID list;
- 2) Sets the Feature Identifier field of the Get Features command to F1h; and
- 3) Sets the UUID Index field of the Get Features command to the index of UUID V.

<End of changes>