LEGAL NOTICE:

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 Name	**TP 4006 – Namespace Granularity**
Date	**2018-08-15**
Impacted Specification	**NVM Express 1.3c**

Technical Proposal Author(s)

Name	Company
Paul Suhler	Micron Technology
Fred Knight	NetApp

This technical proposal defines a mechanism for the controller to indicate to the host the allocation granularities for namespaces.

Revision History

Revision Date	Change Description
13 June 2017	Initial version
01 August 2017	Integrate input from 22 June 2017: Added note to indicate granularity at per-format basis.
16 November 2017	Expanded to specify granularities for NVM sets.
06 December 2017	Integrate input from 30 November 2017: Add “should” guidance for host to use NVM set-specific granularities if they exist. Clarified that granularity must be multiplied by formatted LBA size for comparison with namespace size or capacity.
10 January 2018	Added: • Discussion of use cases and some questions to be decided • List of granularity descriptors • Identify command support indication
22 January 2018	Input from 11 January telecon: • Decisions on direction and scope. • Editorial change. • Optimize for use case 1.
29 January 2018	Input from 25 January telecon: • Report only a single parameter in descriptor • Use a flag to indicate use case 1 • Explain “runt” cases in which size or capacity is not an integral multiple of granularity.
6 March 2018	Input from 8 February telecon: • Revert to two parameters in descriptor
8 March 2018	Input from 8 March telecon: • Annotate with changes to be made in Phase 3.
21 June 2018	Integrated changes from end of Phase 2: • Moved explanation of usage of granularities from 5.20 (Namespace Management command) to 8.12 (Namespace Management) • Revised explanation of granularities and accessible capacity.
24 July 2018	Input from e-mails and 28 June and 19 July telecons: • 8.12.1 Corrected “namespace granularity” to “namespace capacity”. • Revert from tabular representation of calculations to textual.

27 July 2018	Input from 26 July telecon: • Wordsmithing of section 8.12.1. • Included the original “wall of text” version of 8.12.1.
2 August 2018	Input from 2 August telecon: • Delete last change paragraph about thin provisioning. • Ready for member review.
5 October 2018	Integration
15 October 2018	Ratified

Discussion

Namespace allocation granularity may be the same for an entire NVM subsystem, or may vary depending upon the implementation. Examples include:

1. Same granularity for entire NVM subsystem, regardless of LB format and regardless of number and sizes of NVM sets. Number of granularities is one.
2. Granularity varies per LB format, but regardless of number and sizes of NVM sets. Number of granularities is less than or equal to the number of LB formats.
3. Same granularity for all LB formats within a particular NVM set. Granularity may be different for each set. Number of granularities is less than or equal to the number of NVM sets.
4. Granularity varies per LB format per NVM set. Number of granularities is less than or equal to the number of formats times the number of NVM sets.

The most flexible approach is for the controller to report a list of granularity descriptors (with each descriptor corresponding to an LB format) when the Identify command is invoked specifying a new CNS value. Case 1 can be handled by having a single descriptor apply to all LB formats. Case 2 can be handled by having a descriptor per LBA format.

Case 4 could be handled by utilizing the NVM Set Identifier field in the Identify command to specify the NVM set of interest. Index zero indicates the descriptor applying to LBAF0, etc. Case 3 could be handled as a special subcase of #4, where there is a single descriptor.

This TP addresses cases 1 and 2 only.

Description for NVMe 1.4 Changes Document

Namespace Granularity Reporting (optional)

- Identify command returns a list of Granularity Descriptors
- Identify Controller CTRATT field has a bit indicating support for the Granularity Descriptors List
- References:
    - NVMe 1.3c (5.15, 8.12)
    - TP 4006

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 Section 5 as shown below:

5 Admin Command Set

…

5.15 Identify Command

…

Figure 220: 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.	Y	N	5.15.2.1
01h	M	Identify Controller data structure for the controller processing the command.	N	N	5.15.2.2
02h	M	Active Namespace ID list.	Y	N	5.15.2.3
03h	M	Namespace Identification Descriptor list for the specified NSID.	Y	N	5.15.2.4
04h	O	An NVM Set List (refer to Figure 226) is returned to the host for up to 31 NVM Sets. The list contains entries for NVM Set identifiers greater than or equal to the value specified in the NVM Set Identifier (CDW11.NVMSETID) field.	N	N	5.15.2.5
05h to 0Fh		Reserved			
Controller and Namespace Management					
10h	O [4]	Allocated Namespace ID list.	Y	N	5.15.2.6
11h	O [4]	Identify Namespace data structure for the specified allocated NSID.	Y	N	5.15.2.7
12h	O [4]	Controller identifier list of controllers attached to the specified NSID.	Y	Y	5.15.2.8
13h	O [4]	Controller identifier list of controllers that exist in the NVM subsystem.	N	Y	5.15.2.9
14h	O [5]	Primary Controller Capabilities data structure for the specified primary controller.	N	Y	5.15.2.10
15h	O [5]	Secondary Controller list of controllers associated with the primary controller processing the command.	N	Y	5.15.2.11
16h	O	A Namespace Granularity List (refer to Figure TBD1) is returned to the host for up to sixteen Namespace Granularity Entries.	N	N	5.15.2.TBD
~~16h~~ 17h 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).

<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 any other TPs that are also requesting new CNS values.>

…

Figure 223 Identify – Identify Controller data structure

Bytes	O/M [1]	Description
...		
99:96	M	**Controller Attributes (CTRATT):** This field indicates attributes of the controller. Bits ~~31:7~~ 31:8 are reserved. Bit 7 (**Namespace Granularity**): If set to '1', then the controller supports reporting of Namespace Granularity (refer to section 5.20). If cleared to '0', the controller does not support reporting of Namespace Granularity. If the Namespace Management command is not supported, then this bit shall be cleared to '0'. Bit 6 **(Traffic Based Keep Alive Support – TBKAS):** If set to '1', then the controller supports restarting the Keep Alive Timer if an Admin command or an I/O command is processed during the Keep Alive Timeout Interval (refer to section 7.12.2). If cleared to '0', then the controller supports restarting the Keep Alive Timer only if a Keep Alive command is processed during the Keep Alive Timeout Interval (refer to section 7.12.1). 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. 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. 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. 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. 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. 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.

<Editor's Note: The above bit number 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 bit.>

<Editor's Note: Bits 2 - 6 above are included from *NVMe 1.NEXTc*, to facilitate integration.>

5.15.2 Identify Data Structures

...

5.15.2.5 Secondary Controller list (CNS 15h)

…

5.15.2.TBD Namespace Granularity List (16h)

If the controller supports reporting of Namespace Granularity (refer to section 8.12.1), then a Namespace Granularity List (refer to Figure TBD1) is returned to the host for up to sixteen namespace granularity descriptors (refer to Figure TBD2).

Figure TBD1: Namespace Granularity List

Bytes	Description
03:00	**Namespace Granularity Attributes:** This field indicates attributes of the Namespace Granularity List. Bits 31:1 are reserved. Bit 0 (**Granularity Descriptor Mapping**): If set to '1', then each valid namespace granularity descriptor applies to the LBA format having the same index and the Number of Descriptors field shall be equal to the Number of LBA Formats field in the Identify Namespace data structure (refer to Figure 114). If cleared to '0', then NG Descriptor 0 shall apply to all LBA formats and the Number of Descriptors field shall be cleared to 0h.
04	**Number of Descriptors:** This field indicates the number of valid namespace granularity descriptors in the list. This is a 0's based value. The namespace granularity descriptors with an index greater than the value in this field shall be cleared to 0h.
31:05	Reserved
47:32	**NG Descriptor 0:** This field contains the first namespace granularity descriptor in the list. This namespace granularity descriptor applies to LBA formats as indicated by the Granularity Descriptor Mapping bit.
63:48	**NG Descriptor 1:** This field contains the second namespace granularity descriptor in the list. This namespace granularity descriptor applies to LBA Format 1.
…	…
287:272	**NG Descriptor 15:** This field contains the sixteenth namespace granularity descriptor in the list. This namespace granularity descriptor applies to LBA Format 15.

The format of the namespace granularity descriptor is defined in Figure TBD2.

Figure TBD2 Namespace Granularity Descriptor

Bytes	Description
07:00	**Namespace Size Granularity:** Indicates the preferred granularity of allocation of namespace size when a namespace is created. The value is in bytes. A value of 0h indicates that the namespace size granularity is not reported.
15:08	**Namespace Capacity Granularity:** Indicates the preferred granularity of allocation of namespace capacity when a namespace is created. The value is in bytes. A value of 0h indicates that the namespace capacity granularity is not reported.

…

8 Features

…

Modify Portions of Section 8.12 as shown below:

8.12 Namespace Management (Optional)

…

The total and unallocated NVM capacity for the NVM subsystem is reported in the Identify Controller data structure (refer to Figure 223). For controllers that support NVM Sets, the total and unallocated NVM capacity for each NVM Set is reported as part of the NVM Set Attributes Entry (refer to Figure 227). For

each namespace, the NVM Set in which the namespace is allocated is reported in the Identify Namespace data structure. The NVM Set to be used for a namespace is based on the value in the NVM Set Identifier field in a create operation. If the NVM Set Identifier field is cleared to 0h in a create operation, then the controller shall choose the NVM Set from which to allocate the namespace.

For each namespace, the NVM capacity used for that namespace is reported in the Identify Namespace data structure (refer to Figure 221). The controller may allocate NVM capacity in units such that the requested size for a namespace may be rounded up to the next unit boundary. The units in which NVM capacity is allocated are reported in the Namespace Granularity List (refer to Figure TBD1), if supported. For example, if host software requests a namespace of 32 logical blocks with a logical block size of 4KB for a total size of 128KB and the allocation unit for the implementation is 1MB, then the NVM capacity consumed may be rounded up to 1MB. The NVM capacity fields may not correspond to the logical block size multiplied by the total number of logical blocks.

The method of allocating ANA Group identifiers is outside the scope of this specification. If the ANA Group Identifier (refer to Figure 234) is cleared to 0h, then the controller shall determine the ANAGRPID that is assigned to that namespace.

To create a namespace, host software performs the following actions:

1. Host software requests the Identify Namespace data structure that specifies common namespace capabilities (Identify command with ~~a setting of~~ CDW1.NSID set to FFFFFFFFh and CNS cleared to 0h);
2. If the controller supports reporting of Namespace Granularity, host software optionally requests the Namespace Granularity List defined in Figure TBD1 (Identify command with CNS set to 16h).
3. Host software creates the data structure defined in Figure 237. Host software sets the host software specified fields defined in Figure 234 to the desired values (taking into account the common namespace capabilities);
4. Host software issues the Namespace Management command specifying the Create operation and the data structure. On successful completion of the command, the Namespace Identifier of the new namespace is returned in Dword 0 of the completion queue entry. At this point, the new namespace is not attached to any controller; and
5. Host software requests the Identify Namespace data structure for the new namespace to determine all attributes of the namespace.

To attach a namespace, host software performs the following actions:

1. Host software issues the Namespace Attachment command specifying the Controller Attach operation to attach the new namespace to one or more controllers; and
2. If Namespace Attribute Notices are enabled, the controller(s) newly attached to the namespace report a Namespace Attribute Changed asynchronous event to the host.

To detach a namespace, host software performs the following actions:

1. Host software issues the Namespace Attachment command specifying the Controller Detach operation to detach the namespace from one or more controllers; and
2. If Namespace Attribute Notices are enabled, the controllers that were detached from the namespace report a Namespace Attribute Changed asynchronous event to the host.

To delete a namespace, host software performs the following actions:

1. Host software should detach the namespace from all controllers;
2. Host software issues the Namespace Management command specifying the Delete operation for the specified namespace. On successful completion of the command, the namespace has been deleted; and
3. If Namespace Attribute Notices are enabled, any controller(s) that was attached to the namespace reports a Namespace Attribute Changed asynchronous event to the host.

8.12.1 Namespace Granularity

If the controller supports reporting of Namespace Granularity, then the Namespace Granularity Descriptor List (refer to Figure TBD1) contains one or more Namespace Granularity Descriptors (refer to Figure TBD2) indicating the size granularity and the capacity granularity at which the controller allocates namespaces.

The size granularity and the capacity granularity are hints which may be used by the host to minimize the capacity that is allocated for a namespace and that is not able to be addressed by logical block addresses. The granularities are used in specifying values for the Namespace Size (NSZE) and Namespace Capacity (NCAP) fields of the data structure used for the create operation of the Namespace Management command (refer to section 5.20).

If a Namespace Management command create operation specifies values such that:

a) the product of NSZE and the Formatted LBA Size value is an integral multiple of the Namespace Size Granularity;
b) the product of NCAP and the Formatted LBA Size value is an integral multiple of the Namespace Capacity Granularity; and
c) NSZE is equal to NCAP,

then the namespace is fully provisioned and all of the capacity allocated for the namespace is able to be addressed by logical block addresses,

otherwise:

a) not all of the capacity allocated for the namespace is able to be addressed by logical block addresses; and
b) if the Namespace Management command is otherwise valid, then the controller shall not abort the command (i.e., the granularity values are hints).