



#### **LEGAL NOTICE:**

© Copyright 2007 to 2021 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 2021 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 153<sup>rd</sup> Drive  
Beaverton, OR 97003  
USA  
info@nvmexpress.org

## NVM Express Technical Proposal for New Feature

Technical Proposal ID	4075a Get Log Page Index Offsets
Change Date	2021-05-06
Builds on Specification	NVM Express 1.4
References Specification	TP 4071a CEL Enhancements TP 4056a Namespace Types

### Technical Proposal Author(s)

Name	Company
Mike Allison	Intel
Fred Knight	NetApp

This proposal intends allow the Get Log Page command to specify an index offset to a specific data structure in the list of data structures.

### Revision History

Revision Date	Change Description
2020-02-09	Initial version
2020-02-27	Fixed spacing and blue text from light blue.
2020-05-14	Moved red new text to blue new text
2020-06-16	Updates section 5.14.1.12 to reference the bit name and use a bulleted list to not have to repeat it.
2020-07-09	Accepted all changes and removed comments for 30 day member review.
2020-08-25	Integrated into the NVM Express Base Specification.
2020-08-26	Editorial fix in Get Log Page command. Removed repeated shall on same requirement. Command Dword 10 reserved bits were taken by TP 4046 so moved the OT bit to another Command Dword.
2020-08-27	Added reference to TP 4046a and the usage of bits in Command Dword 14.
2020-09-02	Removed all comments and accepts all changes.
2020-10-07	Editorial changes from 30 day member review comments.
2020-10-08	Approved for integration. Changed "status" to "status code".
2020-10-12	Fixed LSUPP to remove extra "by the controller".
2020-10-14	Integrated into the NVM Express Base Specification.

2020-10-15	Replace “than” with “then”.
2021-04-49	Replacing text that was dropped just before the review in development that should not have been.
2021-05-06	Edited during review in WG and removed the one bullet list not separating multiple if's.

## Description for NVMe 1.4 Changes Document

This technical proposal defines the scope of each Feature where the scoping is implied. In addition, Features that require the option of multiple scoping requirements are defined.

## Description of Specification Changes

### Markup Conventions:

Black:	Unchanged (however, hot links are removed)
<del>Red Strikethrough:</del>	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:

### Modify portions of section 5.14 as shown below:

#### 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, 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 191 and Figure 192. 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 support for 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.

If the Log Page Offset is supported, then

- a byte offset shall be supported (i.e., Offset Type bit cleared to ‘0’) for all log pages; and
- for each log page that has the IOS bit set to ‘1’ for the specified LID in the LID Supported and Effects Data Structure log page (refer to Figure TBD\_FX2) an index offset shall be supported (i.e., Offset Type bit set to ‘1’).

If the IOS bit is cleared to ‘0’ for the specified LID in the LID Supported and Effects Data Structure log page and a Get Log Page command specifies Type Offset field set to ‘1’, then that command shall be aborted with a status code of Invalid Field in Command.

**Figure 185: Get Log Page – Data Pointer**

Bits	Description
127:00	<b>Data Pointer (DPTR):</b> This field specifies the start of the data buffer. Refer to Figure 105 for the definition of this field.

**Figure 186: Get Log Page – Command Dword 10**

Bits	Description
31:16	<b>Number of Dwords Lower (NUMDL):</b> This field specifies the lower 16 bits of the number of dwords to return. If host software specifies a size larger than the log page requested, the controller returns the complete log page with undefined results for dwords beyond the end of the log page. The combined NUMDL and NUMDU fields form a 0's based value.
15	<b>Retain Asynchronous Event (RAE):</b> This bit specifies when to retain or clear an Asynchronous Event. If this bit is cleared to '0', the corresponding Asynchronous Event is cleared after the command completes successfully. If this bit is set to '1', the corresponding Asynchronous Event is retained (i.e., not cleared) after the command completes successfully.  Host software should clear this bit to '0' for log pages that are not used with Asynchronous Events. Refer to section 5.2.
14:08	<b>Log Specific Field (LSP):</b> If not defined for the log specified by the Log Page Identifier field, this field is reserved.
07:00	<b>Log Page Identifier (LID):</b> This field specifies the identifier of the log page to retrieve.

**Figure 187: Get Log Page – Command Dword 11**

Bits	Description
31:16	<b>Log Specific Identifier:</b> This field specifies an identifier that is required for a particular log page. The log pages that require a log specific identifier are indicated in the table below.
15:00	<b>Number of Dwords (NUMDU):</b> This field specifies the upper 16 bits of the number of dwords to return.

**Figure 188: Get Log Page – Command Dword 12**

Bits	Description
31:00	<p><b>Log Page Offset Lower (LPOL):</b> The log page offset specifies the location within a log page to start returning data from.</p> <p>If the OT bit is cleared to '0', then:</p> <ul style="list-style-type: none"> <li>a) This field specifies the lower 32 bits of the log page offset. The offset shall be dword aligned, indicated by bits 1:0 being cleared to 00b.</li> <li>b) The controller is not required to check that bits 1:0 are cleared to 00b. The controller may report an error of Invalid Field in Command if bits 1:0 are not cleared to 00b. If the controller does not report an error of Invalid Field in Command, then the controller shall operate as if bits 1:0 are cleared to 00b.</li> <li>c) If the host specifies an offset (i.e., LPOL and LPOU) that is greater than the size of the log page requested (e.g., a log page containing 100 bytes is requested starting at offset 200), then the controller shall abort the command with a status of Invalid Field in Command.</li> </ul> <p>If the OT bit is set to '1', then:</p> <ul style="list-style-type: none"> <li>a) This field specifies the lower 32 bits of the index into the list of data structures in the log page.</li> <li>b) If the host specifies an index (i.e., LPOL and LPOU) that is greater than the number of entries in the list of data structures in the log page requested (e.g., a log page containing 100 data structures is requested starting at index 200), then the controller shall abort the command with a status of Invalid Field in Command.</li> <li>c) If the IOS bit for the specified LID in the LID Supported and Effects Data Structure log page is cleared to '0', then the controller shall abort the command with a status code of Invalid Field in Command.</li> <li>d) Each log page that supports the use of an index offset value defines the contents of an entry for the purposes of indexing into that log page.</li> </ul>

**Figure 189: Get Log Page – Command Dword 13**

Bits	Description
31:00	<p><b>Log Page Offset Upper (LPOU):</b> This field specifies the upper 32 bits of either the log page offset or the index into the list of data structures. Refer to the Log Page Offset Lower definition.</p>

If the controller supports selection of a UUID by the Get Log Page command (refer to Figure 191 and section 8.24), then Command Dword 14 is used to specify a UUID Index value (refer to Figure 190).

**Figure 190: Get Log Page – Command Dword 14**

Bits	Description
31:24	<p><b>Command Set Identifier:</b> Refer to Figure 270. &lt;&lt;refer to TP 4056&gt;&gt;</p>
23	<p><b>Offset Type (OT):</b> If set to '1' then the Log Page Offset Lower field and the Log Page Offset Upper field specify the index into the list of data structures in the log page to be returned. If cleared to '0', then the Log Page Offset Lower field and the Log Page Offset Upper field specify the byte offset into the log page to be returned.</p>
22:07	Reserved
06:00	<p><b>UUID Index:</b> Refer to Figure 498.</p>

*Modify figure TBD\_FX2 section 5.14.TBD from TP 4071a as shown below:*

## 5.14.TBD2 Supported Log Pages (Log Identifier 00h)

...

Figure <TBD\_FX2>: Get Log Page – LID Supported and Effects Data Structure

Bits	Description
31:2	Reserved
1	<b>Index Offset Supported (IOS):</b> If this bit is set to '1', then the controller supports an index offset for this LID in a Get Log Page command (i.e., the OT bit in the Get Log Page command is allowed to be set to '1'). If this bit is cleared to '0', then the controller does not support an index offset for this LID in a Get Log Page command (i.e., the OT bit in the Get Log Page command is only allowed to be cleared to '0').
0	<b>LID Supported (LSUPP):</b> If this bit is set to '1', then the controller supports this LID <del>is supported</del> for a Get Log Page command <del>by the controller</del> . If this bit is cleared to '0', then the controller does not support this LID <del>is not supported</del> for a Get Log Page command <del>by the controller</del> . Refer to section 7.1 for the LID support requirements for each controller type.

*Modify portions of section 5.14.1.12 as shown below:*

### 5.14.1.12 Asymmetric Namespace Access (Log Identifier 0Ch)

This log consists of a header describing the log and descriptors containing the asymmetric namespace access information for ANA Groups (refer to section 8.18.2) that contain namespaces that are attached to the controller processing the command. If ANA Reporting (refer to section 8.18) is supported, this log page is supported. ANA Group Descriptors shall be returned in ascending ANA Group Identifier order.

If the Index Offset Supported bit is cleared to '0' in the LID Support and Effects data structure for this log page (refer to Figure TBD\_FX2), then:

- if the RGO bit is cleared to '0' in Command Dword 10, then the LPOL field in Command Dword 12 and the LPOU field in Command Dword 13 of the Get Log Page command should be cleared to 0h. ~~and~~

If the Index Offset Supported bit is set to '1' in the LID Support and Effects data structure for this log page (refer to Figure 204), then the entry data structure that is indexed is an ANA Group Descriptor (e.g., specifying an index offset of 2 returns this log page starting at the offset of ANA Group Descriptor 1).

If the host performs multiple Get Log Page commands to read the ANA log page (e.g., using the LPOL field or the LPOU field), the host should re-read the header of the log page and ensure that the Change Count field in the Asymmetric Namespace Access Log matches the original value read. If it does not match, then the data captured is not consistent and needs to be re-read.