



**LEGAL NOTICE:**

© Copyright 2007 - 2020 NVM Express, Inc. **ALL RIGHTS RESERVED.**

This erratum to the NVM Express revision 1.4 specification 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 erratum to the NVM Express revision 1.4 specification 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 - 2020 NVM Express, Inc. **ALL RIGHTS RESERVED.**" When making any such citations or references to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of NVM Express, Inc. Nothing contained in this document shall be deemed as granting you any kind of license to implement or use this document or the specification described therein, or any of its contents, either expressly or impliedly, or to any intellectual property owned or controlled by NVM Express, Inc., including, without limitation, any trademarks of NVM Express, Inc.

**LEGAL DISCLAIMER:**

THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN "**AS IS**" BASIS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, NVM EXPRESS, INC. (ALONG WITH THE CONTRIBUTORS TO THIS DOCUMENT) HEREBY DISCLAIM ALL REPRESENTATIONS, WARRANTIES AND/OR COVENANTS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, VALIDITY, AND/OR NONINFRINGEMENT.

All product names, trademarks, registered trademarks, and/or servicemarks may be claimed as the property of their respective owners.

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

## NVM Express™ Technical Errata

Errata ID	002
Revision Date	03/09/2020
Affected Spec Ver.	NVM Express™ 1.4
Corrected Spec Ver.	

### Errata Author(s)

Name	Company
Bill Martin	Samsung

## Errata Overview

- Modify “User Data”
- Modify wording using “logical blocks” where it is not required to be logical block based
- Modified several occurrences of LBA that should have been logical block

## Revision History

Revision Date	Change Description
10/23/2019	Initial creation
10/31/2019	Completed changes in SGL and Sanitize
11/04/2019	Cleaned up change marks for final review
11/13/2019	Added changes to fix where LBA was used in places where logical block should have been used.
03/02/2020	Ready for ratification
03/09/2020	Ratified

## Incompatible Changes

- None

## Description of Specification Changes

### 1 Modify a portion of section 1.6.2 as shown below:

#### 1.6.2 administrative controller

A controller that exposes capabilities that allow a host to manage an NVM subsystem. An administrative controller does not implement I/O Queues, provide access to data or metadata associated with ~~user data logical blocks~~ on a non-volatile memory storage medium, or support namespaces attached to the administrative controller (i.e., there are never any active NSIDs).

### 2 Modify a portion of section 1.6.25 as shown below:

#### 1.6.25 metadata

Metadata is contextual information ~~related to about a~~ particular ~~LBA of~~ data. The host may include metadata to be stored by the NVM subsystem if storage space is provided by the controller. Metadata may include Protection Information (refer to section 8.3).

### 3 Modify a portion of section 1.6.37 as shown below:

#### 1.6.37 user data

Data ~~stored in a namespace~~ that is composed of ~~logical block data and optional metadata~~ data that the host may store and later retrieve including metadata if supported.

### 4 Modify a portion of section 4.4 as shown below:

#### 4.4 Scatter Gather List (SGL)

A Scatter Gather List (SGL) is a data structure in memory address space used to describe a data buffer. The controller indicates the SGL types that the controller supports in the Identify Controller data structure. A data buffer is either a source buffer or a destination buffer. An SGL contains one or more SGL segments. The total length of the Data Block and Bit Bucket descriptors in an SGL shall be equal to or exceed the amount of data ~~requested required by the number of logical blocks to be~~ transferred.

## 5 Modify Figure 118 SGL Bit Bucket descriptor as shown below:

The SGL Bit Bucket descriptor, defined in Figure 118, is used to ignore parts of source data.

**Figure 1: SGL Bit Bucket descriptor**

Bytes	Description						
07:00	Reserved						
11:08	<b>Length:</b> The Length field specifies the amount of source data that is discarded. An SGL Bit Bucket descriptor specifying that no source data be discarded (i.e., the length field cleared to 0h) is a valid SGL Bit Bucket descriptor. If the SGL Bit Bucket descriptor describes a destination data buffer (e.g., a read from the controller to memory), then the Length field specifies the number of bytes of the source data which the controller shall discard (i.e., not transfer to the destination data buffer). If the SGL Bit Bucket descriptor describes a source data buffer (e.g., a write from memory to the controller), then the Bit Bucket Descriptor shall be treated as if the Length field were cleared to 0h (i.e., the Bit Bucket Descriptor has no effect). If SGL Bit Bucket descriptors are supported, their length in a destination data buffer shall be included in the specified length of data to be transferred (e.g., <del>Number of Logical Blocks (NLB) parameter specified in NVM Command Set data transfer commands.</del> (i.e., their length in a source data buffer is not included in the transfer length specified by the NLB parameter).						
14:12	Reserved						
15	<b>SGL Identifier:</b> The definition of this field is described in the table below. <table><tr><th>Bits</th><th>Description</th></tr><tr><td>03:00</td><td><b>SGL Descriptor Sub Type:</b> Valid values are specified in Figure 116.</td></tr><tr><td>07:04</td><td><b>SGL Descriptor Type:</b> 1h as specified in Figure 115.</td></tr></table>	Bits	Description	03:00	<b>SGL Descriptor Sub Type:</b> Valid values are specified in Figure 116.	07:04	<b>SGL Descriptor Type:</b> 1h as specified in Figure 115.
Bits	Description						
03:00	<b>SGL Descriptor Sub Type:</b> Valid values are specified in Figure 116.						
07:04	<b>SGL Descriptor Type:</b> 1h as specified in Figure 115.						

## 6 Modify section 5.2 as shown below:

- a) **NVM Command Set Specific events:** Events that are defined by an I/O command set:
- Reservation Log Page Available event:** Indicates that one or more Reservation Notification log pages (refer to section 5.14.1.16.1) are available. To clear this event, host software reads the Reservation Notification log page using the Get Log Page command with the Retain Asynchronous Event bit cleared to '0';
  - Sanitize Operation Completed event:** Indicates that a sanitize operation has completed (including any associated additional media modification, refer to the No-Deallocate Modifies Media After Sanitize field in Figure 247) without unexpected deallocation of all ~~LBAs-logical blocks~~ (refer to section 5.21.1.23) and status is available in the Sanitize Status log page (refer to section 5.14.1.16.2). To clear this event, host software reads the Sanitize Status log page using the Get Log Page command with the Retain Asynchronous Event bit cleared to '0'; and

**Figure 2: Asynchronous Event Information – NVM Command Set Specific Status**

Value	Description
00h	<b>Reservation Log Page Available:</b> Indicates that one or more Reservation Notification log pages (refer to section 5.14.1.16.1) have been added to the Reservation Notification log.
01h	<b>Sanitize Operation Completed:</b> Indicates that a sanitize operation has completed (including any associated additional media modification, refer to the No-Deallocate Modifies Media After Sanitize field in Figure 247) without unexpected deallocation of all <del>LBAs-logical blocks</del> (refer to section 5.21.1.23) and status is available in the Sanitize Status log page (refer to section 5.14.1.16.2).
02h	<b>Sanitize Operation Completed With Unexpected Deallocation:</b> Indicates that a sanitize operation for which No-Deallocate After Sanitize (refer to Figure 330) was requested has completed with the unexpected deallocation of all <del>LBAs-logical blocks</del> (refer to section 5.21.1.23) and status is available in the Sanitize Status log page (refer to section 5.14.1.16.2).
03h to FFh	Reserved

**Figure 3: Get Log Page – Sanitize Status Log**

Bytes	Description														
01:00	<p><b>Sanitize Progress (SPROG):</b> This field indicates the fraction complete of the sanitize operation. The value is a numerator of the fraction complete that has 65,536 (10000h) as its denominator. This value shall be set to FFFFh if bits 2:0 of the SSTAT field are not set to 010b.</p> <p>If a sanitize operation has been started by a Sanitize command with the No-Deallocate After Sanitize bit set to '1' (refer to section 5.24) and if NODMMAS field in the Identify Controller data structure is set to 10b (refer to Figure 247), then the fraction reported shall include the time related to the additional media modification.</p>														
03:02	<p><b>Sanitize Status (SSTAT):</b> This field indicates the status associated with the most recent sanitize operation.</p> <p>Bits 15:9 are reserved.</p> <p><b>Bit 8 (Global Data Erased):</b> If set to '1', then no namespace logical block in the NVM subsystem has been written to and no Persistent Memory Region in the NVM subsystem has been enabled:</p> <ul style="list-style-type: none"> <li>a) since being manufactured and the NVM subsystem has never been sanitized; or</li> <li>b) since the most recent successful sanitize operation.</li> </ul> <p>If cleared to '0', then a namespace logical block in the NVM subsystem has been written to or a Persistent Memory Region in the NVM subsystem has been enabled:</p> <ul style="list-style-type: none"> <li>a) since being manufactured and the NVM subsystem has never been sanitized; or</li> <li>b) since the most recent successful sanitize operation of the NVM subsystem.</li> </ul> <p>Bits 7:3 contains the number of completed passes if the most recent sanitize operation was an Overwrite. This field shall be cleared to 0h if the most recent sanitize operation was not an Overwrite.</p> <p>Bits 2:0 contains the status of the most recent sanitize operation as shown below.</p> <table border="1"> <thead> <tr> <th>Value</th><th>Definition</th></tr> </thead> <tbody> <tr> <td>000b</td><td>The NVM subsystem has never been sanitized.</td></tr> <tr> <td>001b</td><td>The most recent sanitize operation completed successfully including any additional media modification (refer to the No-Deallocate Modifies Media After Sanitize field in Figure 247).</td></tr> <tr> <td>010b</td><td>A sanitize operation is currently in progress.</td></tr> <tr> <td>011b</td><td>The most recent sanitize operation failed.</td></tr> <tr> <td>100b</td><td>The most recent sanitize operation for which No-Deallocate After Sanitize (refer to section 5.24) was requested has completed successfully with deallocation of all <b>LBA</b>s logical blocks (refer to section 5.21.1.23).</td></tr> <tr> <td>101b to 111b</td><td>Reserved</td></tr> </tbody> </table>	Value	Definition	000b	The NVM subsystem has never been sanitized.	001b	The most recent sanitize operation completed successfully including any additional media modification (refer to the No-Deallocate Modifies Media After Sanitize field in Figure 247).	010b	A sanitize operation is currently in progress.	011b	The most recent sanitize operation failed.	100b	The most recent sanitize operation for which No-Deallocate After Sanitize (refer to section 5.24) was requested has completed successfully with deallocation of all <b>LBA</b> s logical blocks (refer to section 5.21.1.23).	101b to 111b	Reserved
Value	Definition														
000b	The NVM subsystem has never been sanitized.														
001b	The most recent sanitize operation completed successfully including any additional media modification (refer to the No-Deallocate Modifies Media After Sanitize field in Figure 247).														
010b	A sanitize operation is currently in progress.														
011b	The most recent sanitize operation failed.														
100b	The most recent sanitize operation for which No-Deallocate After Sanitize (refer to section 5.24) was requested has completed successfully with deallocation of all <b>LBA</b> s logical blocks (refer to section 5.21.1.23).														
101b to 111b	Reserved														
07:04	<p><b>Sanitize Command Dword 10 Information (SCDW10):</b> This field contains the value of the Command Dword 10 field of the Sanitize command that started the sanitize operation whose status is reported in the SSTAT field. Refer to Figure 330.</p>														
11:08	<p><b>Estimated Time For Overwrite:</b> This field indicates the number of seconds required to complete an Overwrite sanitize operation with 16 passes in the background (refer to section 5.24) when the No-Deallocate Modifies Media After Sanitize field (refer to Figure 247) is not set to 10b. A value of 0h indicates that the sanitize operation is expected to be completed in the background when the Sanitize command that started that operation is completed. A value of FFFFFFFFh indicates that no time period is reported.</p>														
15:12	<p><b>Estimated Time For Block Erase:</b> This field indicates the number of seconds required to complete a Block Erase sanitize operation in the background (refer to section 5.24) when the No-Deallocate Modifies Media After Sanitize field (refer to Figure 247) is not set to 10b. A value of 0h indicates that the sanitize operation is expected to be completed in the background when the Sanitize command that started that operation is completed. A value of FFFFFFFFh indicates that no time period is reported.</p>														

**Figure 3: Get Log Page – Sanitize Status Log**

Bytes	Description
19:16	<b>Estimated Time For Crypto Erase:</b> This field indicates the number of seconds required to complete a Crypto Erase sanitize operation in the background (refer to section 5.24) when the No-Deallocate Modifies Media After Sanitize field (refer to Figure 247) is not set to 10b. A value of 0h indicates that the sanitize operation is expected to be completed in the background when the Sanitize command that started that operation is completed. A value of FFFFFFFFh indicates that no time period is reported.
23:20	<p><b>Estimated Time For Overwrite With No-Deallocate Media Modification:</b> This field indicates the number of seconds required to complete an Overwrite sanitize operation and the associated additional media modification after the Overwrite sanitize operation in the background (refer to section 5.24) when:</p> <ul style="list-style-type: none"> <li>a) the No-Deallocate bit was set to '1' in the Sanitize command that requested the Overwrite sanitize operation; and</li> <li>b) the No-Deallocate Modifies Media After Sanitize field (refer to Figure 247) is set to 10b.</li> </ul> <p>A value of 0h indicates that the sanitize operation is expected to be completed in the background when the Sanitize command that started that operation is completed. A value of FFFFFFFFh indicates that no time period is reported.</p>
27:24	<p><b>Estimated Time For Block Erase With No-Deallocate Media Modification:</b> This field indicates the number of seconds required to complete a Block Erase sanitize operation and the associated additional media modification after the Block Erase sanitize operation in the background (refer to section 5.24) when:</p> <ul style="list-style-type: none"> <li>a) the No-Deallocate bit was set to '1' in the Sanitize command that requested the Block Erase sanitize operation; and</li> <li>b) the No-Deallocate Modifies Media After Sanitize field (refer to Figure 247) is set to 10b.</li> </ul> <p>A value of 0h indicates that the sanitize operation is expected to be completed in the background when the Sanitize command that started that operation is completed. A value of FFFFFFFFh indicates that no time period is reported.</p>
31:28	<p><b>Estimated Time For Crypto Erase With No-Deallocate Media Modification:</b> This field indicates the number of seconds required to complete a Crypto Erase sanitize operation and the associated additional media modification after the Crypto Erase sanitize operation in the background (refer to section 5.24) when:</p> <ul style="list-style-type: none"> <li>a) the No-Deallocate bit was set to '1' in the Sanitize command that requested the Crypto Erase sanitize operation; and</li> <li>b) the No-Deallocate Modifies Media After Sanitize field (refer to Figure 247) is set to 10b.</li> </ul> <p>A value of 0h indicates that the sanitize operation is expected to be completed in the background when the Sanitize command that started that operation is completed. A value of FFFFFFFFh indicates that no time period is reported.</p>
511:32	Reserved

Figure 4: Format NVM – Command Dword 10

Description	
Reserved	
<b>Secure Erase Settings (SES):</b> This field specifies whether a secure erase should be performed as part of the format and the type of the secure erase operation. The erase applies to all user data, regardless of location (e.g., within an exposed LBA, within a cache, within deallocated <b>LBAs logical blocks</b> , etc.).	
Value	Definition
000b	No secure erase operation requested
001b	<b>User Data Erase:</b> All user data shall be erased, contents of the user data after the erase is indeterminate (e.g., the user data may be zero filled, one filled, etc.). The controller may perform a cryptographic erase when a User Data Erase is requested if all user data is encrypted.
010b	<b>Cryptographic Erase:</b> All user data shall be erased cryptographically. This is accomplished by deleting the encryption key.
011b to 111b	Reserved

## 7 Modify Figure 333: Sanitize – Command Dword 10

No modifications are necessary – No Deallocate ONLY refers to drives formatted for logical block data.

Figure 5: Sanitize – Command Dword 10

Bits	Description														
31:10	Reserved														
09	<b>No Deallocate After Sanitize:</b> If set to '1' and the No-Deallocate Inhibited bit (refer to Figure 250) is cleared to '0', then the controller shall not deallocate any <b>logical blocks</b> as a result of successfully completing the sanitize operation. If: <ul style="list-style-type: none"> <li>a) cleared to '0'; or</li> <li>b) set to '1' and the No-Deallocate Inhibited bit is set to '1',</li> </ul> then the controller should deallocate <b>logical blocks</b> as a result of successfully completing the sanitize operation. This bit shall be ignored if the Sanitize Action field is set to 001b (i.e., Exit Failure Mode).														
08	<b>Overwrite Invert Pattern Between Passes (OIPBP):</b> If set to '1', then the Overwrite Pattern shall be inverted between passes. If cleared to '0', then the overwrite pattern shall not be inverted between passes. This bit shall be ignored unless the Sanitize Action field is set to 011b (i.e., Overwrite).														
07:04	<b>Overwrite Pass Count (OWPASS):</b> This field specifies the number of overwrite passes (i.e., how many times the media is to be overwritten) using the data from the Overwrite Pattern field of this command. A value of 0h specifies 16 overwrite passes. This field shall be ignored unless the Sanitize Action field is set to 011b (i.e., Overwrite).														
03	<b>Allow Unrestricted Sanitize Exit (AUSE):</b> If set to '1', then the sanitize operation is performed in unrestricted completion mode. If cleared to '0', then the sanitize operation is performed in restricted completion mode. This bit shall be ignored if the Sanitize Action field is set to 001b (i.e., Exit Failure Mode).														
02:00	<b>Sanitize Action (SANACT):</b> This field specifies the sanitize action to perform. <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>000b</td><td>Reserved</td></tr> <tr> <td>001b</td><td>Exit Failure Mode</td></tr> <tr> <td>010b</td><td>Start a Block Erase sanitize operation</td></tr> <tr> <td>011b</td><td>Start an Overwrite sanitize operation</td></tr> <tr> <td>100b</td><td>Start a Crypto Erase sanitize operation</td></tr> <tr> <td>101b to 111b</td><td>Reserved</td></tr> </table>	Value	Description	000b	Reserved	001b	Exit Failure Mode	010b	Start a Block Erase sanitize operation	011b	Start an Overwrite sanitize operation	100b	Start a Crypto Erase sanitize operation	101b to 111b	Reserved
Value	Description														
000b	Reserved														
001b	Exit Failure Mode														
010b	Start a Block Erase sanitize operation														
011b	Start an Overwrite sanitize operation														
100b	Start a Crypto Erase sanitize operation														
101b to 111b	Reserved														



## 8 Modify Section 8.15 Sanitize Operation (optional) as shown below:

The Overwrite sanitize operation is media specific and may not be appropriate for all media types. For example, if the media is NAND, multiple pass overwrite operations may have an adverse effect on media endurance.

**Figure 6: Sanitize Operations – Overwrite Mechanism**

OIPBP <sup>1</sup>	Overwrite Pass Count <sup>1</sup>	Overwrite Pass Number	Logical Block Data and Non-User Data except PI Metadata <sup>2</sup>	Protection Information <sup>3</sup>
‘0’	All	All	Overwrite Pattern <sup>1</sup>	FFFFFFFF_FFFFFFFFh
‘1’	Even	First	Inversion of Overwrite Pattern <sup>1</sup>	00000000_00000000h
		Subsequent	Inversion of Overwrite Pattern <sup>1</sup> from previous pass (i.e., each bit XORed with ‘1’)	
‘1’	Odd	First	Overwrite Pattern <sup>1</sup>	FFFFFFFF_FFFFFFFFh
		Subsequent	Inversion of Overwrite Pattern <sup>1</sup> from previous pass (i.e., each bit XORed with ‘1’)	
NOTES:				
1. Parameters are specified in Command Dword 10 and Command Dword 11 of the corresponding Sanitize command that started the Overwrite operation. The Overwrite Invert Pattern Between Passes (OIPBP) field is defined in Command Dword 10. The Overwrite Pass Count is defined in Command Dword 10. The Overwrite Pattern is defined in Command Dword 11. Refer to section 5.24.				
2. <del>If metadata other than Protection Information is present.</del>				
3. If Protection Information is present within the metadata.				

To start a sanitize operation, the host submits a Sanitize command specifying one of the sanitize operation types (i.e., Block Erase, Overwrite, or Crypto Erase). The host sets command parameters, including the Allow Unrestricted Sanitize Exit bit and the No Deallocate After Sanitize bit. After validating the Sanitize command parameters, the controller starts the sanitize operation in the background, updates the Sanitize Status log page and then completes the Sanitize command with Successful Completion status. If the sanitize operation is to be followed by an associated additional media modification operation (refer to NODMMAS in Figure 250), then the associated additional media modification operation shall be completed before the controller reports sanitize operation complete. If a Sanitize command is completed with any status other than Successful Completion, then the controller shall not start the sanitize operation and shall not update the Sanitize Status log page. The controller ignores Critical Warning(s) in the SMART / Health Information log page (e.g., read only mode) and attempts to complete the sanitize operation requested. While a sanitize operation is in progress, all controllers shall abort any commands not listed in Figure 485 with a status of Sanitize In Progress (refer to section 8.15.1).

The user data values that result from a successful sanitize operation are specified in Figure 484. If the controller deallocates user data after successful completion of a sanitize operation, then values read from **deallocated logical blocks** are described in section 6.7.1.1. The host may specify that **sanitized logical blocks** not be deallocated by setting the No Deallocate After Sanitize bit to '1' in the Sanitize command.

**Figure 7: Sanitize Operations – User Data Values**

Sanitize Operation	<del>Logical Blocks</del> User Data	<del>Non-PI Metadata<sup>1</sup></del>	Protection Information <sup>2</sup>
Block Erase	Vendor specific value	<del>Vendor specific value</del>	<del>Vendor specific value</del>
Crypto Erase	Indeterminate	<del>Indeterminate</del>	<del>Indeterminate</del>
Overwrite	Refer to Figure 483	<del>Refer to Figure 483</del>	<del>Refer to Figure 483</del>
NOTES: 1. <del>If metadata other than Protection Information is present.</del> 2. <del>If Protection Information is present within the metadata.</del>			