



LEGAL NOTICE:

© Copyright 2007 - 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 - 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.

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	TP 4065b
Change Date	2/19/2021
Builds on Specification	NVM Express 1.4
References Ratified TPs	TP4040

Technical Proposal Author(s)

Name	Company
Fred Knight	NetApp
Michael Allison	Intel
Klaus Jensen	Samsung

This technical proposal defines a Simple Copy command. This command can be used to copy data between different LBA ranges on a single namespace.

Before TP4065 was published, it was updated to produce this version (TP4065a). As a result, this is the first published version.

Revision History

Revision Date	Change Description
8/13/19	Initial creation (multi namespace, multi source, single destination).
8/15/19	Switch to use a copy function within the DSM command (single namespace, multi source, single destination).
8/28/19	Switch back to a unique command but use the DSM data structure format (single namespace, multi source, single destination).
8/29/19	Move the destination specification into the SQE and make all Ranges be for source data.
9/4/19	Fill in error codes.
9/5/19	Address comments from Mike Allison; add copy to appropriate lists of read, write commands.
9/19/19	Add additional PI clarifications from David Black
10/16/19	Add Per Source Range PI fields, reorganize command registers, use sum of read length fields as the write length.
10/18/19	Add Descriptor version, allow for KPIO fields, move limits from Controller to Namespace.
10/23/19	Incorporate feedback from David Black and Christoph; SMART log, Range descriptor formats, and clarify PI for Copy.
11/21/19	Remove <green> notes
12/19/19	Use same error code added in TP4040 (Command Size Limit Exceeded)
1/9/20	30-day member feedback review at Work group and forward for integration
2/20/2020	Removed editorial notes. Moved 8.3.1.6 to 8.3.1.NEW. Ready for ratification. Fixed reference in MSSRL field.
4/14/2020	Add explicit reference to text about copying unwritten/deallocated blocks. <changes are in blue>
4/17/2020	Remove change tracking, also update short summary description.
4/20/2020	Swap ELBAT and ELBATM in the Source Range Entry – so it matches how the READ command sends those values to the controller – to make it easier for H/W to parse that Dword value.
4/21/2020	Update Endurance Group Log to include counting Copy commands.
4/23/2020	Update description to explain why TP4065 (without the “a”) doesn’t exist.
9/11/2020	Adding Copy command to the Invalid Protection Information status code.
10/22/2020	Removed comments and accepted all changed for member review.
12/14/2020	Integrated into the NVMe Base Specification
1/14/2021	Removed all comments and accepted all changes for ratification.
1/20/2021	Updated the Optional Copy Formats Supported field to specify in figure 247 to specify bits 1:15 are reserved to match TP 4065a. Updated Figure 9991 to list the title for the Invalid Protection Information status code.
1/21/2021	Accepted all changes and removed all comments for ratification.
2/19/2021	Fixup reference ERROR! tags, add missing Figure numbers from some references.

Black text indicates existing material in the NVMe Base 1.4 specification, Red text indicates material added by this TP.

Description of Specification Changes

Modify the following sections as shown below:

4.6.1.2.2 Command Specific Status Definition

...

Figure 129: Status Code – Command Specific Status Values, NVM Command Set

Value	Description	Commands Affected
80h	Conflicting Attributes	Dataset Management, Read, Write
81h	Invalid Protection Information	Compare, Copy, Read, Verify, Write, Write Zeroes
82h	Attempted Write to Read Only Range	Copy, Dataset Management, Write, Write Uncorrectable, Write Zeroes
83h	Command Size Limit Exceeded	Data Set Management, Copy
84h to BFh	Reserved	

4.6.1.2.3 Media and Data Integrity Errors Definition

...

Figure 131: Status Code – Media and Data Integrity Error Values, NVM Command Set

Value	Description
...	
87h	Deallocated or Unwritten Logical Block: The command failed due to an attempt to copy from, read from, or verify an LBA range containing a deallocated or unwritten logical block.
88h to BFh	Reserved

5.14.1.2 SMART / Health Information (Log Identifier 02h)

...

Figure 194: Get Log Page – SMART / Health Information Log

Bytes	Description
...	
79:64	Host Read Commands: Contains the number of read commands completed by the controller. For the NVM command set, this value is the sum of the number of Compare commands, and the number of Read commands, and the number of Copy commands.
95:80	Host Write Commands: Contains the number of write commands completed by the controller. For the NVM command set, this is the number of Write commands, and Copy commands.
...	

5.14.1.9 Endurance Group Information (Log Identifier 09h)

...

Figure 204: Get Log Page – Endurance Group Log (Log Identifier 09h)

Bytes	Description
111:96	Host Read Commands: Contains the number of read commands completed by all controllers in the NVM subsystem for the Endurance Group. For the NVM command set, this is the number of Compare commands, Copy commands, and Read commands.
127:112	Host Write Commands: Contains the number of write commands completed by all controllers in the NVM subsystem for the Endurance Group. For the NVM command set, this is the number of Write commands and Copy commands.
...	

Modify the following sections as shown below:

5.15.2.1 Identify Namespace data structure (CNS 00h)

...

Figure 245: Identify – Identify Namespace Data Structure, NVM Command Set Specific

Bytes	O/M ¹	Description
...		
75:74	O	Maximum Single Source Range Length (MSSRL): This field indicates the maximum number of logical blocks that may be specified in each valid Source Range field (refer to 3) of a Copy command. If the controller supports the Copy command, then this field shall be set to a non-zero value.
79:76	O	Maximum Copy Length (MCL): This field indicates the maximum number of logical blocks that may be specified in a Copy command (i.e., the sum of the number of logical blocks specified in all Source Range entries). If the controller supports the Copy command, then this field shall be set to a non-zero value.
80	O	Maximum Source Range Count (MSRC): This field indicates the maximum number of Source Range entries that may be used to specify source data in a Copy command. This is a 0's based value.
91:7481		Reserved
...		

5.15.2.2 Identify Controller data structure (CNS 01h)

...

Figure 247: Identify – Identify Controller Data Structure

Bytes	O/M ¹	Description
...		

Figure 247: Identify – Identify Controller Data Structure

Bytes	O/M ¹	Description
521:520	M	<p>Optional NVM Command Support (ONCS): This field indicates the optional NVM commands and features supported by the controller. Refer to section 0.</p> <p>Bits 15:9 are reserved.</p> <p>Bit 8 if set to '1', then the controller supports the Copy command. If cleared to '0', then the controller does not support the Copy command.</p> <p>Bit 7 if set to '1', then the controller supports the Verify command. If cleared to '0', then the controller does not support the Verify command.</p> <p>Bit 6 if set to '1', then the controller supports the Timestamp feature. If cleared to '0', then the controller does not support the Timestamp feature. Refer to section 5.21.1.14.</p> <p>Bit 5 if set to '1', then the controller supports reservations. If cleared to '0', then the controller does not support reservations. If the controller supports reservations then the following commands associated with reservations shall be supported: Reservation Report, Reservation Register, Reservation Acquire, and Reservation Release. Refer to section 8.8 for additional requirements.</p> <p>Bit 4 if set to '1', then the controller supports the Save field set to a non-zero value in the Set Features command and the Select field set to a non-zero value in the Get Features command. If cleared to '0', then the controller does not support the Save field set to a non-zero value in the Set Features command and the Select field set to a non-zero value in the Get Features command.</p> <p>Bit 3 if set to '1', then the controller supports the Write Zeroes command. If cleared to '0', then the controller does not support the Write Zeroes command.</p> <p>Bit 2 if set to '1', then the controller supports the Dataset Management command. If cleared to '0', then the controller does not support the Dataset Management command.</p> <p>Bit 1 if set to '1', then the controller supports the Write Uncorrectable command. If cleared to '0', then the controller does not support the Write Uncorrectable command.</p> <p>Bit 0 if set to '1', then the controller supports the Compare command. If cleared to '0', then the controller does not support the Compare command.</p>
...		
535:534		<p>Optional Copy Formats Supported:</p> <p>Bits 15:1 are reserved.</p> <p>Bit 0 if set to '1', then the controller supports Copy Format 0h. If cleared to '0', then the controller does not support Copy Format 0h.</p>
...		

5.27 Get LBA Status command – NVM Command Set Specific

...

Figure 345: LBA Status Descriptor Entry

Bytes	Description
07:00	Descriptor Starting LBA (DSLBA): This field specifies the 64-bit address of the first logical block of the LBA range for which this LBA Status Descriptor reports LBA status.

Figure 345: LBA Status Descriptor Entry

Bytes	Description								
11:08	Number of Logical Blocks (NLB): This field contains the number of contiguous logical blocks reported in this LBA Status Descriptor. The controller should return the largest aggregated possible value in this field. This is a 0's based value.								
12	Reserved								
13	Status: This field contains additional status about this LBA range.								
	<table><tr><th>Bits</th><th>Definition</th></tr><tr><td>7:2</td><td>Reserved</td></tr><tr><td>1</td><td>If set to '1', this LBA range describes LBAs written with a Write Uncorrectable command. If cleared to '0', this LBA range may or may not describe LBAs written with a Write Uncorrectable command.</td></tr><tr><td>0</td><td>If set to '1', a Copy, Read, Verify, or Compare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status. If cleared to '0', the controller has not detected that a Copy, Read, Verify, or Compare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status.</td></tr></table>	Bits	Definition	7:2	Reserved	1	If set to '1', this LBA range describes LBAs written with a Write Uncorrectable command. If cleared to '0', this LBA range may or may not describe LBAs written with a Write Uncorrectable command.	0	If set to '1', a C opy, R ead, V erify, or C ompare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status. If cleared to '0', the controller has not detected that a C opy, R ead, V erify, or C ompare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status.
	Bits	Definition							
	7:2	Reserved							
1	If set to '1', this LBA range describes LBAs written with a Write Uncorrectable command. If cleared to '0', this LBA range may or may not describe LBAs written with a Write Uncorrectable command.								
0	If set to '1', a C opy, R ead, V erify, or C ompare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status. If cleared to '0', the controller has not detected that a C opy, R ead, V erify, or C ompare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status.								
15:14	Reserved								

...

6 NVM Command Set

...

Figure 346: Opcodes for NVM Commands

Opcode by Field			Combined Opcode ¹	Command ²	Reference Section
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer ³			
0b	000 00b	00b	00h	Flush ⁴	6.8
0b	000 00b	01b	01h	Write	6.15
0b	000 00b	10b	02h	Read	6.9
0b	000 01b	00b	04h	Write Uncorrectable	6.16
0b	000 01b	01b	05h	Compare	6.6
0b	000 10b	00b	08h	Write Zeroes	6.17
0b	000 10b	01b	09h	Dataset Management	6.7
0b	000 11b	00b	0Ch	Verify	6.14
0b	000 11b	01b	0Dh	Reservation Register	6.11
0b	000 11b	10b	0Eh	Reservation Report	6.13
0b	001 00b	01b	11h	Reservation Acquire	6.10
0b	001 01b	01b	15h	Reservation Release	6.12
0b	001 10b	01b	19h	Copy	6.TBD
Vendor Specific					
1b	n/a	NOTE 3	80h to FFh	Vendor specific	
NOTES: 1. Opcodes not listed are reserved. 2. All NVM commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value. 3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional. 4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.					

Add the following sections alphabetically (after the Compare command and before the Dataset Management command) as shown below:

6.TBD Copy command

The Copy command is used by the host to copy data from one or more source logical block ranges to a single consecutive destination logical block range.

The command uses Command Dword 10, Command Dword 11, Command Dword 12, Command Dword 13, Command Dword 14, and Command Dword 15 fields. If the command uses PRPs for the data transfer, then the PRP Entry 1 and PRP Entry 2 fields are used. If the command uses SGLs for the data transfer, then the SGL Entry 1 field is used. All other command specific fields are reserved.

Figure 9999: Copy – Data Pointer

Bits	Description
127:00	Data Pointer (DPTR): This field specifies the data to use for the command. Refer to Figure 105 for the definition of this field.

Figure 9998: Copy – Command Dword 10 and Command Dword 11

Bits	Description
63:00	Starting Destination LBA (SDLBA): This field indicates the 64-bit address of the first logical block to be written as part of the copy operation. Command Dword 10 contains bits 31:00; Command Dword 11 contains bits 63:32.

Figure 9997: Copy – Command Dword 12

Bits	Description												
31	Limited Retry (LR): If set to '1', the controller should apply limited retry efforts for the write portion of the copy operation. If cleared to '0', the controller should apply all available error recovery means to write the data to the NVM.												
30	Force Unit Access (FUA): If set to '1', then for data and metadata, if any, associated with logical blocks specified by the write portion of the copy operation, the controller shall write that data and metadata, if any, to non-volatile media before indicating command completion. There is no implied ordering with other commands. If cleared to '0', then this bit has no effect.												
29:26	Protection Information Field Write (PRINFOW): Specifies the protection information action and check field, as defined in Figure 355, to be used for the write portion of the copy operation.												
25:24	Reserved												
23:20	Directive Type (DTYPE): Specifies the Directive Type associated with the Directive Specific field (refer to section 9.1) used for the write portion of the copy operation.												
19:16	Reserved												
15:12	Protection Information Field Read (PRINFOR): Specifies the protection information action and check field, as defined in Figure 355, to be used for the read portion of the copy operation specified by each Source Range Entries.												
11:08	<table><tr><td colspan="3">Descriptor Format: Specifies the format of the Source Range Entries as follows:</td></tr><tr><td>Code</td><td>Description</td><td>Reference</td></tr><tr><td>0h</td><td>The Source Range Entries specify starting LBA, number of logical blocks, and parameters associated with the read portion of the operation.</td><td>Figure 9993</td></tr><tr><td>All Others</td><td colspan="2">Reserved</td></tr></table>	Descriptor Format: Specifies the format of the Source Range Entries as follows:			Code	Description	Reference	0h	The Source Range Entries specify starting LBA, number of logical blocks, and parameters associated with the read portion of the operation.	Figure 9993	All Others	Reserved	
Descriptor Format: Specifies the format of the Source Range Entries as follows:													
Code	Description	Reference											
0h	The Source Range Entries specify starting LBA, number of logical blocks, and parameters associated with the read portion of the operation.	Figure 9993											
All Others	Reserved												
07:00	Number of Ranges (NR): Specifies the number of Source Range Entries that are specified in the command. This is a 0's based value.												

Figure 9996: Copy – Command Dword 13

Bits	Description
31:16	Directive Specific (DSPEC): Specifies the Directive Specific value associated with the Directive Type field (refer to section 9.1).
15:00	Reserved

Figure 9995: Copy – Command Dword 14

Bits	Description
31:00	Initial Logical Block Reference Tag (ILBRT): This field specifies the Initial Logical Block Reference Tag value for the write portion of the copy operation. This field is only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.3.

Figure 9994: Copy – Command Dword 15

Bits	Description
31:16	Logical Block Application Tag Mask (LBATM): This field specifies the Application Tag Mask value for the write portion of the copy operation. This field is only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.3.
15:00	Logical Block Application Tag (LBAT): This field specifies the Application Tag value for the write portion of the copy operation. This field is only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.3.

The data that the Copy command provides is a list of Source Range Entries that describe the data to be copied to the destination range starting at the SDLBA. The format of the Source Range Entries is specified in the Descriptor Format field. If the specified Descriptor Format is not supported by the controller, then the command shall be aborted with a status code of Invalid Field in Command.

Figure 9993 shows the 0h descriptor format and an example with 128 Source Range entries.

Figure 9993: Copy – Source Range Entries

Range	Bytes	Description	
Source Range 0	07:00	Reserved	
	15:08	Starting LBA	
	19:16	Read Parameters as follows:	
		Bits	Description
		31:16	Reserved
	15:00	Number of Logical Blocks (NLB): This field indicates the number of logical blocks to be copied. This is a 0's based value.	
	23:20	Reserved	
	27:24	Expected Initial Logical Block Reference Tag (EILBRT): This field specifies the Initial Logical Block Reference Tag expected value used for the read portion of the copy operation for the LBAs specified in this Source Range entry. This field is only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.3.	
29:28	Expected Logical Block Application Tag (ELBAT): This field specifies the Application Tag expected value used for the read portion of the copy operation for the LBAs specified in this Source Range entry. This field is only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.3.		
31:30	Expected Logical Block Application Tag Mask (ELBATM): This field specifies the Application Tag Mask expected value used for the read portion of the copy operation for the LBAs specified in this Source Range entry. This field is only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.3.		
Source Range 1	39:32	Reserved	
	47:40	Starting LBA	
	51:48	Read Parameters	
	55:52	Reserved	
	59:56	EILBRT	
	61:60	ELBAT	
	63:62	ELBATM	
...			
Source Range 127	4071:4064	Reserved	
	4079:4072	Starting LBA	
	4083:4080	Read Parameters	
	4087:4084	Reserved	
	4091:4088	EILBRT	
	4093:4092	ELBAT	
	4095:4094	ELBATM	

If the number of Source Range entries (i.e., the value in the NR field) is greater than the value in the MSRC field (refer to Figure 247), then the Copy command shall be aborted with a status code of Command Size Limit Exceeded.

If a valid Source Range Entry specifies a Number of Logical Blocks field that is greater than the value in the MSSRL field (refer to Figure 247), then the Copy command shall be aborted with a status code of Command Size Limit Exceeded.

If the sum of all Number of Logical Blocks fields in all Source Range entries is greater than the value in the MCL field (refer to Figure 247), then the Copy command shall be aborted with a status code of Command Size Limit Exceeded.

The number of logical blocks written by the Copy command is the sum of all Number of Logical Blocks fields in all Source Range entries specified in the list of Source Range entries.

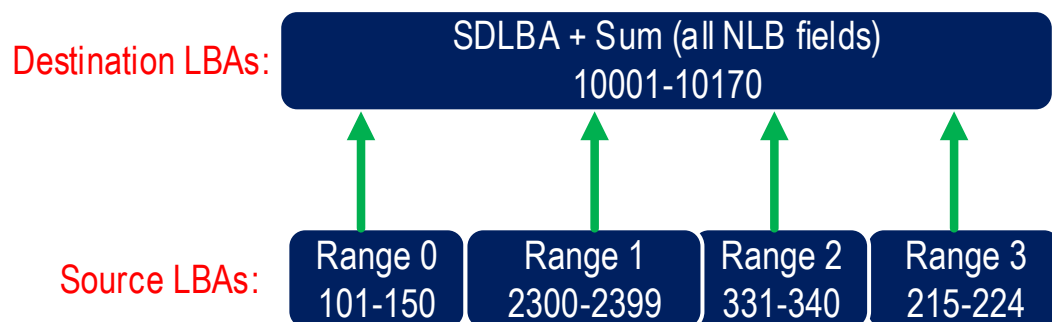
The data bytes in the LBAs specified by each Source Range Entry shall be copied to the destination LBA range in the same order those LBAs are listed in the Source Range entries (e.g., the LBAs specified by Source Range entry 0 are copied to the lowest numbered LBAs specified by the SDLBA field, the LBAs specified by Source Range entry 1 are copied to the next consecutively numbered LBAs specified by the SDLBA field). The read operations and write operations used to perform the copy may operate sequentially or in parallel.

If the read portion of a copy operation attempts to access a deallocated or unwritten logical block, the controller shall operate as described in section 6.7.1.1.

If the namespace is formatted to use end-to-end protection information, then the protection information is handled as described in section 0.

Figure 9992 shows an example of the relationship between the source LBAs and the destination LBAs.

Figure 9992: Source LBA and Destination LBA Relationship Example



6.TBD.1 Command Completion

When the command is completed, the controller shall post a completion queue entry to the associated I/O Completion Queue indicating the status for the command.

If the command completes with failure, then Dword 0 of the completion queue entry contains the number of the lowest numbered Source Range entry that was not successfully copied (e.g., if Source Range 0, Source Range 1, Source Range 2, and Source Range 5 are copied successfully and Source Range 3 and Source Range 4 are not copied successfully, then Dword 0 is set to 3). If no data was written to the destination LBAs, then Dword 0 of the completion queue entry shall be set to 0h.

Copy command specific errors are defined in Figure 9991.

Figure 9991: Copy – Command Specific Status Values

Value	Description
81h	Invalid Protection Information: The protection information specified by the command is invalid due to: <ul style="list-style-type: none"> the Protection Information Field Read (PRINFOR) field or Protection Information Field Write (PRINFOW) field (refer to Figure 359) containing an invalid value for the Protection Information with which the namespace was formatted (refer to the PI field in Figure 328 and the DPS field in Figure 245); the ILBRT field being invalid (refer to section 8.3.1.5); or the EILBRT field in a Source Range Entry being invalid (refer to section 8.3.1.5).

Figure 9991: Copy – Command Specific Status Values

Value	Description
82h	Attempted Write to Read Only Range: The destination LBA range specified contains read-only blocks. The controller shall not return this status value if the read-only condition on the media is a result of a change in the write protection state of a namespace (refer to section 8.19).
83h	Command Size Limit Size Exceeded: One or more of the Copy command processing limits (i.e., non-zero value of the NR, MSSRL, and MCL fields in the Identify Namespace data structure) was exceeded.
87h	Deallocated or Unwritten Logical Block: The command failed due to an attempt to copy from an LBA range containing a deallocated or unwritten logical block.

...

6.7.1.1 Deallocate

...

Using the Error Recovery feature (refer to section 5.21.1.5), host software may enable an error to be returned if a deallocated or unwritten logical block is read. If this error is supported for the namespace and enabled, then a **Copy**, Read, Verify, or Compare command that includes a deallocated or unwritten logical block shall fail with the ~~Unwritten or~~ Deallocated **or Unwritten** Logical Block status code. Note: Legacy software may not handle an error for this case.

8.3 End-to-end Data Protection (Optional)

...

8.3.1.NEW Protection Information and Copy commands

Protection information processing during a Copy command parallels both Write and Read commands. For the portion of the Copy command that transfers data and protection information from the LBAs described by a Source Range Entry (refer to Figure 9993), the protection information checks performed by the controller are controlled by the PRINFOR field in Copy command Dword 12 (refer to Figure 9997) and parallels the Read command protection information checks (refer to section 8.3.1.2) as follows:

- The logical block data and metadata is transferred from the NVM to the controller.
- As the logical block data and metadata pass through the controller, the protection information within the metadata is checked. If a protection information check error is detected, the command completes with the status code of the error detected (i.e., End-to-end Guard Check Error, End-to-end Application Tag Check Error, or End-to-end Reference Tag Check Error).

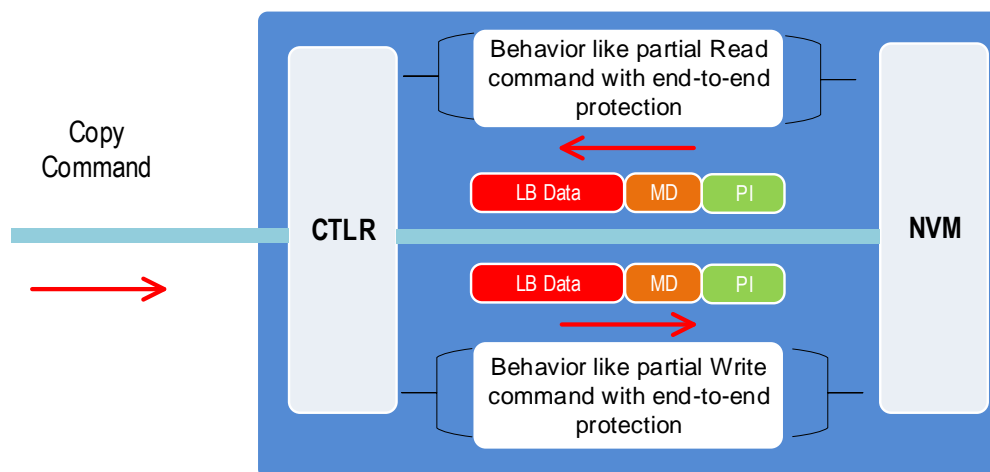
For the portion of the Copy command that transfers data and protection information to the LBAs starting at the SDLBA field (refer to Figure 9998), the protection information operations performed by the controller are controlled by the PRINFOW field in Copy command Dword 12 (refer to Figure 9997) and parallels the Write command protection information checks (refer to section 8.3.1.1) as follows:

- The logical block data and metadata are transferred from the controller to the NVM.
- As the logical block data and metadata passes through the controller, the protection information is handled as described in section 8.3.1.1.

If the PRACT bit is cleared to '0' in the PRINFOR field and the PRACT bit is set to '1' in the PRINFOW field, then the Copy command shall be aborted with a status code of Invalid Field in Command. If the PRACT bit is

set to '1' in the PRINFOR field and the PRACT bit is cleared to '0' in the PRINFOR field, then the Copy command shall be aborted with a status code of Invalid Field in Command.

Figure 9990: Protection Information Processing for Copy



Protection Information with PRACT bit cleared to '0' (i.e., pass)

While data is passing through the controller the data should never be unprotected (e.g., Calculate the PI data associated with the write portion of the copy operation occurs before verification and removal of PI data associated with the read portion of the copy operation). If the guard field is recalculated, it should be compared to the original guard field (i.e., the guard field associated with the read portion of the copy operation).

8.8.3 Reservation Types

...

Figure 462: Command Behavior in the Presence of a Reservation

NVMe Command	Write Exclusive Reservation		Exclusive Access Reservation		Write Exclusive Registrants Only or Write Exclusive All Registrants Reservation		Exclusive Access Registrants Only or Exclusive Access All Registrants Reservation	
	Non-Registrant	Registrant	Non-Registrant	Registrant	Non-Registrant	Registrant	Non-Registrant	Registrant
NVM Copy Command Group								
Copy	C	C	C	C	C	A	C	A
NVM Read Command Group								
Compare Read Security Receive (Admin) Verify	A	A	C	C	A	A	C	A
NVM Write Command Group								

Figure 462: Command Behavior in the Presence of a Reservation

NVMe Command	Write Exclusive Reservation		Exclusive Access Reservation		Write Exclusive Registrants Only or Write Exclusive All Registrants Reservation		Exclusive Access Registrants Only or Exclusive Access All Registrants Reservation	
	Non-Registrant	Registrant	Non-Registrant	Registrant	Non-Registrant	Registrant	Non-Registrant	Registrant
Dataset Management Flush Format NVM (Admin) Namespace Attachment (Admin) Namespace Management (Admin) Sanitize (Admin) Security Send (Admin) Write Write Uncorrectable Write Zeroes	C	C	C	C	C	A	C	A
Reservation Command Groups								
Reservation Acquire - Acquire	C	C	C	C	C	C	C	C
Reservation Acquire - Preempt Reservation Acquire - Preempt and Abort Reservation Release	C	A	C	A	C	A	C	A
All Other Commands Group								
All other commands ¹	A	A	A	A	A	A	A	A
Key: A definition: A=Allowed, command processed normally by the controller C definition: C=Conflict, command aborted by the controller with status Reservation Conflict Notes: 1. The behavior of a vendor specific command is vendor specific.								

...

8.25 Improving Performance through I/O Size and Alignment Adherence

NVMe controllers may require constrained I/O sizes and alignments to achieve the full performance potential. There are a number of optional attributes that the controller uses to indicate these recommendations. If hosts do not follow these constraints, then the controller shall function correctly, but performance may be limited.

Each **Copy**, Write, Write Uncorrectable, or Write Zeroes commands should address a multiple of Namespace Preferred Write Granularity (NPWG) (refer to Figure 245) and Stream Write Size (SWS) (refer to Figure 515) logical blocks (as expressed in the NLB field), and the SLBA field of the command should be aligned to Namespace Preferred Write Alignment (NPWA) (refer to Figure 245) for best performance. Each range in a Dataset Management command with the Attribute - Deallocate (AD) bit set to '1' should contain a multiple of Namespace Preferred Deallocate Granularity (NPDG) (refer to Figure 245) logical blocks and the start of each range should be aligned to Namespace Preferred Deallocate Alignment (NPDA) (refer to Figure 245) and Stream Granularity Size (SGS) logical blocks.

...

8.25.1 Improved I/O examples (non-normative)

It is recommended that the host utilize the I/O attributes as reported by the controller to receive optimal performance from the NVM subsystem. This section summarizes performance related attributes from namespaces, streams, NVM Sets and the NVM command set. The I/O commands discussed throughout this section include those that interact with non-volatile storage in either a Read, Compare, **Copy**, Verify, Write, Write Uncorrectable, Write Zeroes operation, or Dataset Management operation with the Attribute - Deallocate bit set to '1'. The I/O command properties of length and alignment are discussed throughout this section.

...

In Figure 499 an example namespace is diagrammed with three Namespace I/O Boundaries (NOIOB) (refer to Figure 245). The NOIOB attribute should be applied to Read, Compare, **Copy**, Verify, Write, Write Uncorrectable, and Write Zeroes I/O commands. The four green lines are example I/O commands from the host that adhere to the recommendations of NOIOB settings for this namespace. None of the four I/O commands shown in green on the top of Figure 499 cross an NOIOB. The three I/O commands shown in red on the bottom of Figure 499 violate the recommendations for improved performance. The longest I/O command shown in red crosses one NOIOB and ends aligned with a different NOIOB.

...

Continuing with the same namespace example from Figure 499, an illustration of Namespace Atomic Boundary Offset (NABO) (refer to Figure 245) and Namespace Atomic Boundary Size Normal (NABSN) (refer to Figure 245) is shown in Figure 500. NABSN and NABO attributes apply to **Copy**, Write, Write Uncorrectable, and Write Zeroes commands. NABSN and NOIOB may not be related to each other, and there may be an offset of NABO to locate the first NABSN starting. The NOIOBs are not shown in Figure 500. The I/O commands shown in green on the top of Figure 500 illustrate I/O commands that adhere to the namespace's guidance for optimal performance. The I/O commands shown in red on the bottom illustrate I/O commands that do not follow the optimal performance guidelines.

...

NPWG and NPWA are namespace internal constructs, and they are illustrated in Figure 501. The box at the top of Figure 501 is the namespace. The series of boxes in the middle layer indicate many namespace optimal write units described by NPWA (refer to Figure 245) and NPWG (refer to Figure 245), and the bottom layer is a series of eight logical blocks that in aggregate form the NPWG for this example. Sometimes NPWG are useful because several sequential logical blocks (refer to Figure 245) may be placed and tracked together on the media, or the NPWG may be related to NVM subsystem data reliability implementation constraints. NPWG and NPWA attributes apply to **Copy**, Write, Write Uncorrectable, and Write Zeroes commands.