



LEGAL NOTICE:

© **Copyright 2008 to 2023 NVM Express®, Inc. ALL RIGHTS RESERVED.**

This 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 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: “© 2008 to 2023 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 153rd Drive
Beaverton, OR 97003
USA
info@nvmexpress.org

NVM Express® Technical Proposal (TP)

Technical Proposal ID	TP 4130 Cross-Namespace Copy
Revision Date	2023-09-27
Builds on Specifications	NVM Express NVM Command Set Specification 1.0c NVM Express Base Specification 2.0c
References	TP 4098a Multiple Atomicity TP 4141 Storage Tag Mask Enhancements

Technical Proposal Author(s)

Name	Company
Prakash Venkatanarayanan, David Black	Dell EMC
Murali Rajagopal	VMware

Technical Proposal Overview

Extend Copy command to copy data across different namespaces on the same NVM subsystem via support for specifying a namespace for each source range. Extend Protection Information (PI) support to cross-namespaces copy, including the ability to insert or strip PI in common cases where all logical block metadata is PI.

Clarify ambiguities in originally-specified atomicity properties of original Copy command (also applies to Copy across namespaces).

Revision History

Revision Date	Change Description
2021-12-04	<ul style="list-style-type: none"> Initial rough version
2022-02-09	<ul style="list-style-type: none"> Update to NVM Command Set Spec 1.0b and update references Move NSID to first 4 bytes of Source Range Descriptor Entry Initial draft of Protection Information text, based on FMDS meeting design discussions
2022-02-13	<ul style="list-style-type: none"> Rewrite Protection Information text. Finish cleanup (e.g., table size reduction) started in previous version
2022-02-15	<ul style="list-style-type: none"> Define fields in FLBAS field in identify namespace. Add some description of source range entry descriptor formats
2022-02-23	Reset change tracking, add changes from FMDS meeting discussion: <ul style="list-style-type: none"> Fix spec references Collapse new source range entry descriptor formats into existing tables Add new summary text on PRACT behavior to 5.2.2.5, move details to Copy command specification in 3.2.2
2022-02-27	<ul style="list-style-type: none"> Rewrite PRACT behavior summary to be descriptive. All normative requirements will be in 3.2.2.1 and 3.2.2.2
2022-03-01	Reset change tracking, add notes and changes from FMDS meeting discussion <ul style="list-style-type: none"> Remove PRACT=1 support for cross-namespace copy. Allow both DIF and DIX for cross-namespace copy
2022-03-22	<ul style="list-style-type: none"> Protection Information and namespace format compatibility text Add placeholder for table to characterize functionality of copy descriptor formats
2022-03-23	Reset change tracking <ul style="list-style-type: none"> Expand PRACT bit text.
2022-03-28	<ul style="list-style-type: none"> New table to show copy source descriptor functionality. Extend rules on copy source descriptor format use to new formats. Additional minor edits.
2022-03-29	<ul style="list-style-type: none"> Updates for comments from Paul Suhler and FMDS meeting discussion Reset change tracking and then make significant changes from FMDS meeting discussion: <ul style="list-style-type: none"> Remove TP 4055 dependencies. Shrink copy descriptor format tables to show only first 8 bytes where the changes are. Handle incompatible formats for copy as a separate error case with a new status code.
2022-04-08	<ul style="list-style-type: none"> Add support bits for new copy descriptor formats to Identify Controller in Base Spec Add errors for invalid source NSID, source namespace not attached to controller and source namespace has incompatible command set (i.e., Key Value). Checked command completion text – no change required, as current text describes behavior for any failures, including failures added by this TP.
2022-04-19	<ul style="list-style-type: none"> Add copy support requirement across namespaces with different command sets. Add reference to ZNS command set. Prohibit overlapping source and destination ranges for new descriptor formats.
2022-07-06	<ul style="list-style-type: none"> Initial version/outline for Copy Group text Merge in atomicity and overlap changes from ECN 114
2022-07-12	<ul style="list-style-type: none"> Complete copy group support (UUID in Identify namespace, bit to only do copy within same copy group, new error status associated with that bit).
2022-07-13	<ul style="list-style-type: none"> Reset change tracking Rename Same Copy Group bit to Fast Copy Only to better align with SCSI functionality and use SCSI Fast Copy Not Possible name for error. Define new Overlapping I/O Range error for source/destination overlap when one of the new source descriptor formats is used.
2022-07-15	<ul style="list-style-type: none"> Add Host Behavior Support functionality requiring explicit host enable of cross-namespace copy.

2022-07-19	<ul style="list-style-type: none"> Additional minor edits, plus notes from FMDS meeting on two areas that still need attention: <ul style="list-style-type: none"> Fast Copy Only bit is supposed to mean do fast copy, not check whether namespaces belong to same copy group. New bit for atomicity functionality needs to point to rules in a new atomicity subsection, comment describes what rules will specify.
2022-07-21	<ul style="list-style-type: none"> Minor edits from FMDS and Technical WG discussion
2022-07-25	<ul style="list-style-type: none"> Move FCO bit from command into source range entries for new formats Update caption for copy PI processing picture – Figure 135
2022-08-29	<ul style="list-style-type: none"> Remove copy groups text, leaving FCO (Fast Copy Only), add ONCS bit to indicate that all copy operations are fast copy operations. Move text indicating that partial copy may occur if Copy command aborted with an error to the command completion section. Narrow scope of ONCS atomicity backwards compatibility bit to multiple source ranges. Add additional namespace format match attributes from TP 4141.
2022-08-30	<ul style="list-style-type: none"> FMDS discussion: Requirement to set ONCS atomicity backwards compatibility bit: Requirement to set bit to '1' applies only to NVMe versions subsequent to 2.0.
2022-09-01	<ul style="list-style-type: none"> Updated descriptions for new ONCS bits – revise backwards compatibility bit based on 8/30 FMDS discussion and revise fast copy bit to refer to both NVM and ZNS command sets.
2022-09-06	<ul style="list-style-type: none"> Accept changes reviewed by FMDS, including splitting copy source and destination in reservation conflict tables (eliminates Copy command group). Include all command-specific status values for Copy in TP, as new text indicating that some copying may have occurred applies to all of them.
2022-09-16	<ul style="list-style-type: none"> Rewrite (again) the backwards compatibility bit text. Add fast copy method examples from SCSI (SPC-5).
2022-09-20	<ul style="list-style-type: none"> FMDS discussion added a couple of notes to follow up on: <ul style="list-style-type: none"> Back out mentions of ZNS command set for backwards compatibility bit in favor of a general ECN statement that ONCS also applies to ZNS. Expand PRACT functionality to support Insert, Strip, and Simple Replace (no format changes) of PI.
2022-09-26	<ul style="list-style-type: none"> Back out ZNS command set mentions in backwards compatibility bit.
2022-10-16	<ul style="list-style-type: none"> Move error case text for invalid/inactive source NSIDs Add description of “all copies are fast copies” bit to fast copy text New section 3.2.2.2 on Copy atomicity (with associated renumbering)
2022-10-17	<ul style="list-style-type: none"> Expanded PI functionality – first version of Copy command and protection information handling text for expanded PRACT functionality
2022-10-18	<ul style="list-style-type: none"> Add sentence to say that the “all copies are fast copies” bit should be set if applicable. Add examples for atomic operations being part of a source range and spanning multiple source ranges. Extensive rewrite of section 5.2.2.5.2 including removing duplicative text that specifies protection information checks.
2022-10-21	<ul style="list-style-type: none"> Add summary of format and protection information restrictions for copy
2022-10-24	<ul style="list-style-type: none"> Updates from face-to-face meeting
2022-10-31	<ul style="list-style-type: none"> Pick up changes from Yoni Shternhell, plus additional editorial changes Rework Host Behavior Support bits 0 and 1 to make them do nothing – these would be the enable bits for copy source formats 0h and 1h, which don't have enable bits, and never will.
2022-11-01	<ul style="list-style-type: none"> FMDS discussion result: rewrite 3.2.2.4 text on namespace formats and protection information in order to avoid any implication that the fields of all Storage Range entries have to be scanned and checked before any copying occurs.

2022-11-05	<ul style="list-style-type: none"> Reword backwards compatibility atomicity text to be based on what host is able to expect if new ONCS bit x is cleared to '0' (suggestion from Fred Knight and Randy Jennings).
2022-11-08	<ul style="list-style-type: none"> Editorial corrections from Judy Brock. Updates from FMDS discussion: <ul style="list-style-type: none"> Remove requirement for same PI type and format across Source Ranges when PI is being removed. Notes for massive editorial restructure of PI text in next version – swap use of words “matched and compatible”, and cover all PI processing with “For each Source Range” Unused Host Behavior Support bits (would be enable bits for existing copy formats) are ignored for Set Features and cleared to '0' for Get Features Discussion of “Fast copy” decided not to change name, so that concept has same name in SCSI and NVMe Also converted a bunch of comments to Editor's Notes.
2022-11-09	<ul style="list-style-type: none"> Protection information text restructure Accept other outstanding editorial changes.
2022-11-14	<ul style="list-style-type: none"> Minor edits from 11-10 technical WG meeting
2022-11-15	<ul style="list-style-type: none"> Record decisions from FMDS discussion in comments: <ul style="list-style-type: none"> Structure text on protection information (PI) interaction with copying to depend on whether or not source namespace for Source Range 0 and destination namespace are formatted with PI. Remove PRACT=1 support for namespaces formatted with both PI and non-PI metadata. This is uncommon - can be addressed by host-based copy.
2022-11-16	<ul style="list-style-type: none"> Reset change tracking and make the two changes agreed on in FMDS discussion (see above 2022-11-15 entry). Editorial changes and comments from Mike Allison
2022-11-17	<ul style="list-style-type: none"> Minor error case update from Technical WG meeting – when copying between source and destination namespaces that have PI, consistently use Invalid Field in Command to report invalid PRACT bit values.
2022-12-02	<ul style="list-style-type: none"> Clean Version for Phase 2 exit.
2022-12-09	<ul style="list-style-type: none"> Changes from Randy Jennings, plus additional editorial changes
2022-12-12	<ul style="list-style-type: none"> Add examples of what is and is not expected for fast copy (with contributions from Judy Brock) Update to current TP template
2022-12-14	<ul style="list-style-type: none"> Reset change tracking Revert reference to additional Figure in 5.2.2.5 – change filed as ECN bug because it affects current revision of NVM Command Set spec. Additional minor edits.
2022-12-15	<ul style="list-style-type: none"> Two changes from Fred Knight – 1) always use “stripped” to specify PI removal, and 2) for new status codes, use TBDSC instead of TBD to avoid confusion. Generalize references to ZNS command set to include other logical block command sets that inherit the NVM Command Set Copy command. Convert explanation of how Copy commands are checked for reservation conflict to a table footnote (twice)
2023-01-09	<ul style="list-style-type: none"> Add text on interaction of Fast Copy Not Possible error w/ Do Not Retry in CQE. Add precise definition of LBA range overlap. Update copyright dates to 2023 Major work items still outstanding: <ul style="list-style-type: none"> Descriptions for changes documents Update to Base 2.0c and NVM Command Set 1.0c specs Additional figures for Protection Information processing Change ONCS bits (in Identify Controller) to embedded table and add names for the two new bits

2023-01-10	<ul style="list-style-type: none"> • Updates from FMDS discussion: <ul style="list-style-type: none"> ◦ Accept changes in 12-14, 12-15 and 01-09 entries. ◦ Minor modifications to some of the accepted text, particularly the interaction of Fast Copy Not Possible with Do Not Retry ◦ Reservation conflict checks are on namespaces, not ranges
2023-01-11	<ul style="list-style-type: none"> • Update to NVM Command Set Spec 1.0c • Update to Base Spec 2.0c (no effects on TP4130 content) • Add text for rest of ONCS bits from NVM Command Set Spec 1.0c. • Move ONCS bits text into embedded table, name the two new bits.
2023-01-17	<ul style="list-style-type: none"> • Updates from FMDS discussion: <ul style="list-style-type: none"> ◦ Adjust name of All Fast Copy bit to apply only to NVM Command Set Copy command ◦ Shorten and revise reservation conflict check text. ◦ Fold sentence stating that partial copy may have happened in error cases into specification of CQE Dword 0 contents.
2023-01-18	<ul style="list-style-type: none"> • Minor cleanups from Paul Suhler
2023-01-23	<ul style="list-style-type: none"> • Add description of changes for Base spec
2023-01-24	<ul style="list-style-type: none"> • Updates from FMDS discussion: <ul style="list-style-type: none"> ◦ Accept outstanding changes ◦ Revise text stating that partial copy may have happened in error cases.
2023-01-25	<ul style="list-style-type: none"> • Add proposal overview and description of changes for NVM Command Set spec.
2023-01-31	<ul style="list-style-type: none"> • Reset change tracking. • Add first new PI processing diagram for review.
2023-02-06	<ul style="list-style-type: none"> • Revise PI processing diagram, add the other five. • Editorial and formatting cleanups from Mike Allison
2023-02-07	<ul style="list-style-type: none"> • More Navigation pane cleanup • FMDS discussion: use all-lower-case “metadata size” instead of “Metadata Size” • Minor editorial cleanup
2023-02-16	<ul style="list-style-type: none"> • Correct NVM Command Set PI processing Figure number and text that uses it. • Correct reservation figure number.
2023-02-23	<ul style="list-style-type: none"> • Version for member review – no content changes from 2023-02-16 version
2023-03-01	<ul style="list-style-type: none"> • Member review comment from Murali Rajagopal – make it clearer that overlap means overlap within same namespace.
2023-03-07	<ul style="list-style-type: none"> • Member review comment from Fred Knight – make it clearer that support for the Copy command (ONCS bit) means support for all applicable active namespaces as both copy source and copy destination.
2023-03-28	<ul style="list-style-type: none"> • Move FCO bit into its own Dword in new Source Descriptor formats. • Rewrite paragraph on namespace support to focus on source namespace and add Key Value namespace as an example where copy is not possible. • Extensive editorial changes from Mike Allison’s member review comments. • Consistently use Source Range Entries Copy Descriptor formats
2023-04-04	<ul style="list-style-type: none"> • Minor edits resulting from FMDS review.
2023-04-06	<ul style="list-style-type: none"> • Final review of updates resulting from member review, accept all changes.
2023-04-23	<ul style="list-style-type: none"> • Integrated
2023-04-25	<ul style="list-style-type: none"> • Updated from Mike Allison and David Black edits
2023-04-26	<ul style="list-style-type: none"> • Corrected the Save Options bytes offsets.
2023-05-20	<ul style="list-style-type: none"> • Updated from Mike Allison and David Black edits
2023-05-21	<ul style="list-style-type: none"> • Accept all changes, then back out value assignments to prepare for technical changes and second member review.
2023-05-22	<ul style="list-style-type: none"> • Add CLBNSS field to indicate scope of Copy support for logical block namespaces.

2023-05-23	Results from FMDS meeting: <ul style="list-style-type: none"> Define a command-specific "Insufficient Resources" status code (e.g., that can also be used with Reservations) and specify its use with Copy. Remove CLBNSS field.
2023-05-25	Final edits to prepare for second member review.
2023-06-26	Put back a couple of TBDs that were missed in backing out value assignments in preparation for second member review. Member review comment from Mike Allison.
2023-06-27	Remove possible host responses to new Insufficient Resources error status. Member review comment from FMDS meeting.
2023-09-14	Integrated
2023-09-26	Updated integration from David Black and Mike Allison review comments

Description for Changes Document for NVM Command Set Specification 1.0c

New Features/Feature Enhancements/Required Changes:

- Cross-Namespace Copy (optional)
 - Copy command: Extend existing Copy command to copy across namespaces - two new source range descriptor formats that include a source namespace for each source range (source namespace may be the same as destination namespace). Hosts have to explicitly enable these formats via an extension of the Host Behavior Support feature. New status codes for errors that can only occur when the new formats are used.
 - Protection Information (PI): Extend PI support to cross-namespace copy, including ability to insert or strip PI in common cases where all logical block metadata is PI.
 - Fast copy: New Fast Copy Only (FCO) bit in new source range descriptor formats to enable hosts to avoid slow copying. Setting that bit to '1 causes controller to abort Copy command (with a new status code) instead of performing slow copying.
 - Reservations: Update Reservation Conflict table to separate reservation checks for read and write portions of the Copy command.
- Copy command atomicity clarification (mandatory if NVM Command Set Copy command is supported, including ZNS Command Set usage of that Copy command)
 - **New requirement / incompatible change** in the new Copy atomicity section.
 - Some implementations of the Copy command apply atomicity requirements only to each source range. New implementations are required to apply those requirements to the entire Copy command and set a (new) bit in Identify Controller to indicate this behavior.
 - **New requirement / incompatible change** in in section 4.1.5.2.
 - The FFFFh value (0's-based) for atomicity parameters (e.g., AWUN – Atomic Write Unit Normal) only promises Copy command atomicity up to 10000h logical blocks. It does not promise that all Copy commands are atomic.
- Insufficient Resources status (optional)
 - Add a new command-specific status code for Insufficient Resources and specify its use with the Copy command.

References:

- NVM Command Set Specification 1.0c
- Technical Proposal 4130: Cross-Namespace Copy

- Technical Proposal 4098a: Multiple Atomicity
- Technical Proposal 4141: Storage Tag Mask Enhancements

Description for Changes Document for NVM Express® Base Specification 2.0c

New Features/Feature Enhancements/Required Changes:

- Cross-Namespace Copy (optional)
 - Three new status codes for errors that may result from attempts to copy data across multiple namespaces
 - New ONCS (Optional NVM Command Support) bit in Identify Controller to indicate whether all copies within an NVM Subsystem are fast copies (NVMAFC bit).
 - Two new bits in Identify Controller to indicate support for the two new copy descriptor formats that enable copy across namespaces.
 - Two new Host Behavior Support bits for the host to explicitly enable the two new copy descriptor formats (both new formats start out disabled).
 - Update Reservation Conflict table to separate reservation checks for read and write portions of the Copy command.
- Copy command atomicity clarification (mandatory if NVM Command Set Copy command is supported, including ZNS Command Set usage of that Copy command)
 - **New requirement / incompatible change** in Figure 275 in section 5.17.2.1
 - New ONCS bit in Identify Controller to indicate whether scope of atomicity guarantees for the Copy command is the entire command or each source range (NVMSCA). Implementations that comply with this specification are required to set that bit to '1' to indicate entire command scope – that requirement is stated in the additions to the NVM Command Set Specification.
- Insufficient Resources status (optional)
 - Add a new command-specific status code for Insufficient Resources.
- References:
 - NVM Express Base Specification 2.0c
 - Technical Proposal 4130: Cross-Namespace Copy

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
Purple:	Text moved without change
Orange:	Text added in TP4156
<Green Bracketed>:	Notes to editor

Description of Specification Changes for NVM Command Set Specification 1.0c:

Modify section 1.6 by adding the reference shown below:

1.6 References

NVM Express Zoned Namespace Command Set Specification, Revision 1.1.
Available from <http://www.nvmexpress.org>.

Modify a portion of section 2.1.4 as shown below:

2.1.4 Atomic Operation

...

The NVM subsystem reports in the Identify Controller data structure the size in logical blocks of the write operation guaranteed to be written atomically under various conditions, including normal operation, power fail, ~~and in the write operation for a Compare & Write fused operation command~~ (refer to section 2.1.3.1), and the write portion of a Copy command (refer to section 3.2.2). The values reported in the Identify Controller data structure are valid across all namespaces with any supported namespace format, forming a baseline value that is guaranteed not to change.

Modify figure 16 in Section 3.1.2 as shown below:

Figure 16: Status Code – Command Specific Status Values

Value	Description	Commands Affected
...
85h	Incompatible Namespace or Format	Copy
86h	Fast Copy Not Possible	Copy
87h	Overlapping I/O Range	Copy
89h	Insufficient Resources	Copy
84h 8Ah to BFh	Reserved	

Modify section 3.2.2 as shown below:

3.2.2 Copy command

The Copy command is used by the host to copy **user** data from one or more source logical block ranges in one or more source namespaces to a single consecutive destination logical block range in a destination namespace. Each source logical block range may be in the same namespace or a different namespace with respect to any other source logical block range and with respect to the destination logical block range.

The command uses Command Dword 2, Command Dword 3, 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 30: Copy – Command Dword 12

Bits	Description	
...		
11:08	Descriptor Format: Specifies the type of the Copy Descriptor Format that is used. The Copy Descriptor Format specifies the starting LBA, number of logical blocks, and parameters associated with the read portion of the operation.	
	Copy Descriptor Format Type	Description
	0h	Source Range Entries Copy Descriptor Format 0h is used (refer to Figure 34).
	1h	Source Range Entries Copy Descriptor Format 1h is used (refer to Figure 35).
	2h	Source Range Entries Copy Descriptor Format 2h is used (refer to Figure 34).
	3h	Source Range Entries Copy Descriptor Format 3h is used (refer to Figure 35).
	All Others	
...		

...

The controller shall indicate the Source Range Entries Copy Descriptor Formats supported by the controller in the Copy Descriptor Formats Supported field in the Identify Controller data structure (refer to the NVM Express Base Specification).

Controller usage of Source Range Entries Copy Descriptor Formats 2h and 3h is further qualified by whether the host has enabled these formats in the Host Behavior Support feature (refer to the NVM Express Base Specification). If the controller supports a Source Range Entries Copy Descriptor Format that has not been enabled, the controller shall process Copy commands as if that format is not supported (e.g., if that format is specified in the Descriptor Format field in Command Dword 12, the controller shall abort the command with a status code of Invalid Field in Command). Source Range Entries Copy Descriptor Formats 0h and 1h are always enabled if supported. A host that enables Source Range Entries Copy Descriptor Formats 2h and/or 3h indicates to the controller that the host accepts the implications (e.g., for namespace access control) of the presence of a Source Namespace Identifier (SNSID) in these formats (refer to Figure 34 and Figure 35).

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 Copy Descriptor Format type of the Source Range Entries is specified in the Descriptor Format field in Command Dword 12. The Copy Descriptor Format types are distinguished by the supported protection information formats (refer to section 5.2.1) and whether the Copy Descriptor Format contains a Source Namespace Identifier (SNSID) field that supports a copy source in a different namespace than the copy destination, as described in Figure nnn. For a Copy Descriptor Format that does not contain an SNSID field, the source namespace is the same as the destination namespace which is specified by the NSID field in the command.

Figure nnn: Copy – Copy Descriptor Formats

Copy Descriptor Format type	Protection Information Formats	SNSID field present	Description
0h	16b Guard Protection Information	No	Protection Information size: 8 bytes, source namespace and destination namespace are the same.
1h	32b Guard Protection Information 64b Guard Protection Information	No	Protection Information size: 16 bytes, source namespace and destination namespace are the same.
2h	16b Guard Protection Information	Yes	Protection Information size: 8 bytes, source namespace may differ from destination namespace.
3h	32b Guard Protection Information 64b Guard Protection Information	Yes	Protection Information size: 16 bytes, source namespace may differ from destination namespace.

If the Copy Descriptor Format specified in the Descriptor Format field is not supported by the controller, then the command shall be aborted with a status code of Invalid Field in Command.

If:

- the Copy Descriptor Format specified in the Descriptor Format field is supported by the controller;
- the destination namespace specified by the NSID field is formatted to use 16b Guard Protection Information; and
- the Descriptor Format field is not cleared to 0h and is not set to 2h,

then the command shall be aborted with the a status code of Invalid Namespace or Format.

If:

- the Copy Descriptor Format specified in the Descriptor Format field is supported by the controller;
- the destination namespace specified by the NSID field is formatted to use 32b Guard Protection Information or 64b Guard Protection Information; and
- the Descriptor Format field is not set to 1h and is not set to 3h,

then the command shall be aborted with the a status code of Invalid Namespace or Format.

Figure 34 shows the Copy Descriptor Format 0h and Format 2h descriptors and with an example with that has 128 Source Range entries.

Figure 34: Copy – Source Range Entries Copy Descriptor Format 0h and Format 2h

Range	Bytes	Description									
Source Range 0	03:00	Source Parameters as follows:									
		<table><tr><th>Format 0h bytes</th><th>Format 2h bytes</th><th>Description</th></tr><tr><td>03:00</td><td>n/a</td><td>Reserved</td></tr><tr><td>n/a</td><td>03:00</td><td>Source Namespace Identifier (SNSID): Specifies the source namespace for this Source Range entry.</td></tr></table>	Format 0h bytes	Format 2h bytes	Description	03:00	n/a	Reserved	n/a	03:00	Source Namespace Identifier (SNSID): Specifies the source namespace for this Source Range entry.
		Format 0h bytes	Format 2h bytes	Description							
		03:00	n/a	Reserved							
	n/a	03:00	Source Namespace Identifier (SNSID): Specifies the source namespace for this Source Range entry.								
	07:04Q	Reserved									
	15:08	Starting LBA									
	19:16	Read Parameters as follows:									
		<table><tr><th>Bits</th><th>Description</th></tr><tr><td>31:16</td><td>Reserved</td></tr><tr><td>15:00</td><td>Number of Logical Blocks (NLB): This field indicates the number of logical blocks to be copied. This is a 0's based value.</td></tr></table>	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.			
		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.										
2321:20	Reserved										
23:22	Source Options as follows:										
	<table><tr><th>Format 0h bits</th><th>Format 2h bits</th><th>Description</th></tr><tr><td>n/a</td><td>15</td><td>Fast Copy Only (FCO): If set to '1' the controller only performs fast copy operations (refer to section 3.2.2.STBD1) for user data in this Source Range. If cleared to '0', this bit has no effect.</td></tr><tr><td>15:00</td><td>14:00</td><td>Reserved</td></tr></table>	Format 0h bits	Format 2h bits	Description	n/a	15	Fast Copy Only (FCO): If set to '1' the controller only performs fast copy operations (refer to section 3.2.2.STBD1) for user data in this Source Range. If cleared to '0', this bit has no effect.	15:00	14:00	Reserved	
	Format 0h bits	Format 2h bits	Description								
	n/a	15	Fast Copy Only (FCO): If set to '1' the controller only performs fast copy operations (refer to section 3.2.2.STBD1) for user data in this Source Range. If cleared to '0', this bit has no effect.								
15:00	14:00	Reserved									
...	...										
Source Range 1	35:32	Source Parameters									
	39:362	Reserved									
	47:40	Starting LBA									
	51:48	Read Parameters									
	53:52	Reserved									
	55:54	Reserved-Source Options									
									
Source Range 127	4067:4064	Source Parameters									
	4071:40684	Reserved									
	4079:4072	Starting LBA									
	4083:4080	Read Parameters									
	4085:4084	Reserved									
	4087:4086	Reserved-Source Options									
									

The SNSID field (refer to [Figure 34](#)) specifies an active NSID that identifies the namespace for the source range. If the SNSID field contains an invalid NSID, the value 0h or the value FFFFFFFFh, then the controller shall abort the Copy command with a status code of Invalid Namespace or Format. If the SNSID field contains an inactive NSID, then the controller shall abort the Copy command with a status code of Invalid Field in Command.

Figure 35 shows the Copy Descriptor Format 1h and Format 3h descriptors and with an example with that has 102 Source Range entries.

Figure 35: Copy – Source Range Entries **Copy Descriptor Format 1h and Format 3h**

Range	Bytes	Description									
Source Range 0	03:00	Source Parameters as follows:									
		<table><tr><th>Format 1h bytes</th><th>Format 3h bytes</th><th>Description</th></tr><tr><td>03:00</td><td>n/a</td><td>Reserved</td></tr><tr><td>n/a</td><td>03:00</td><td>Source Namespace Identifier (SNSID): Specifies the source namespace for this Source Range entry.</td></tr></table>	Format 1h bytes	Format 3h bytes	Description	03:00	n/a	Reserved	n/a	03:00	Source Namespace Identifier (SNSID): Specifies the source namespace for this Source Range entry.
		Format 1h bytes	Format 3h bytes	Description							
		03:00	n/a	Reserved							
	n/a	03:00	Source Namespace Identifier (SNSID): Specifies the source namespace for this Source Range entry.								
	07:040	Reserved									
	15:08	Starting LBA									
	19:16	Read Parameters as follows:									
		<table><tr><th>Bits</th><th>Description</th></tr><tr><td>31:16</td><td>Reserved</td></tr><tr><td>15:00</td><td>Number of Logical Blocks (NLB): This field indicates the number of logical blocks to be copied. This is a 0's based value.</td></tr></table>	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.			
		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	Source Options as follows:										
	<table><tr><th>Format 1h bits</th><th>Format 3h bits</th><th>Description</th></tr><tr><td>n/a</td><td>15</td><td>Fast Copy Only (FCO): If set to '1' the controller only performs fast copy operations (refer to section 3.2.2.STBD1) for user data in this Source Range. If cleared to '0', this bit has no effect.</td></tr><tr><td>15:00</td><td>14:00</td><td>Reserved</td></tr></table>	Format 1h bits	Format 3h bits	Description	n/a	15	Fast Copy Only (FCO): If set to '1' the controller only performs fast copy operations (refer to section 3.2.2.STBD1) for user data in this Source Range. If cleared to '0', this bit has no effect.	15:00	14:00	Reserved	
	Format 1h bits	Format 3h bits	Description								
	n/a	15	Fast Copy Only (FCO): If set to '1' the controller only performs fast copy operations (refer to section 3.2.2.STBD1) for user data in this Source Range. If cleared to '0', this bit has no effect.								
15:00	14:00	Reserved									
25:240	Reserved										
...	...										
Source Range 1	43:40	Source Parameters									
	47:440	Reserved									
	55:48	Starting LBA									
	59:56	Read Parameters									
	...										
	63:62	Source Options									
	65:640	Reserved									
...	...										
Source Range 101	4043:4040	Source Parameters									
	4047:40440	Reserved									
	4055:4048	Starting LBA									
	4059:4056	Read Parameters									
	40615:4060	Reserved									
	4063:4062	Source Options									
	4065:4064	Reserved									
									

The SNSID field (refer to [Figure 35](#)) specifies an active NSID that identifies the namespace for the source range. If the SNSID field contains an invalid NSID, the value 0h or the value FFFFFFFFh, then the controller shall abort the Copy command with a status code of Invalid Namespace or Format. If the SNSID field contains an inactive NSID, then the controller shall abort the Copy command with a status code of Invalid Field in Command.

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 97), 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 97), 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 97), 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.

Two LBA ranges overlap if they specify LBAs in the same namespace and there is at least one LBA that is part of both LBA ranges.

For Source Range Entries Copy Descriptor Formats 0h and 1h, the host should not specify a destination LBA range that overlaps the LBA in any of the Source Range entries. If the host specifies a destination LBA range that overlaps with any LBAs specified in one or more of the Source Range entries, then upon completion of the Copy command, the data stored in each logical block in that overlapping destination LBA range may, within the constraints of the atomicity rules described in section 2.1.4, be from any of the one or more Source Range entries in which that LBA is contained. This is a result of the possibility that overlapping Source Range entries may be processed in any order.

For Source Range Entries Copy Descriptor Formats 2h and 3h, overlap of any source LBA range that is located in the destination namespace with the destination LBA range is prohibited. If a Copy command uses either Source Range Entries Copy Descriptor Format 2h or 3h and any specified source LBA range that is located in the destination namespace has any LBAs in common with the specified destination LBA range, then the controller shall abort the command with a status code of Overlapping I/O Range.

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 3.2.3.2.1.

~~If the namespace is formatted to use end-to-end protection information, then the protection information is handled as described in section 5.2.2.5.~~

...

<Note: section 3.2.2.STBD1 is before 3.2.2.1 .>

3.2.2.STBD1 Fast copy operations

A fast copy operation is a copy operation that uses a method that is expected to be no slower in total elapsed time than the alternative of the host copying the user data by issuing Read commands and Write commands. Unexpected NVM subsystem operating conditions (e.g., nature of concurrent I/O traffic, availability of controller buffer space, errors, and failures) may cause individual copy operations to be slower than host copying of the user data. High performance methods for fast copy operations include non-read/write methods such as copy on write snapshot and copy on write clone and high-bandwidth copies within a tightly integrated NVM subsystem such as an SSD.

A host is able to restrict a Copy command to only perform fast copy operations for user data specified by a Source Range entry by setting the Fast Copy Only (FCO) bit to '1' in that Source Range entry (refer to

Figure 34 and Figure 35). If the FCO bit is set to '1' in a Source Range entry and the controller is unable to use fast copy operations to copy the user data specified by that Source Range entry, then the controller shall abort the command with a status code of Fast Copy Not Possible.

If the controller aborts a Copy command with a status code of Fast Copy Not Possible and clears the Do Not Retry (DNR) bit to '0' in the CQE for that command, then the host may retry that Copy command (e.g., by submitting that command to a different controller). If the controller aborts a Copy command with a status code of Fast Copy Not Possible and sets the Do Not Retry (DNR) bit to '1' in the CQE for that command, then the host should not retry that Copy command (e.g., the host may submit Read commands and Write commands to copy the user data).

A controller sets the NVM All Fast Copy (NVMAFC) bit to '1' in the Optional NVM Command Support (ONCS) field of the I/O Command Set Independent Identify Controller data structure (refer to the NVM Express Base Specification) to indicate that for Copy commands for this specification, all copy operations are fast copy operations within the NVM subsystem that contains the controller. If all copy operations within that NVM subsystem are fast copy operations, then the controller should set that bit to '1'.

3.2.2.STBD2 Copy atomicity

If a controller:

- complies with a version of the NVM Command Set Specification later than revision 1.0;
- complies with a version of the NVM Express Base Specification later than revision 2.0; or
- supports the Copy command and either:
 - Source Range Entries Copy Descriptor Format 2h; or
 - Source Range Entries Copy Descriptor Format 3h,

then the controller shall set the NVM Copy Single Atomicity (NVMCSA) bit to '1' in the ONCS field and shall perform the write portion of a Copy command as a single write command to which the atomicity requirements specified in section 2.1.4 apply.

In some situations, these atomicity requirements require the controller to process user data specified by a portion of a Source Range entry as an atomic write operation, or to process user data specified by multiple Source Range entries (and/or portions thereof) as an atomic write operation. For example, consider a controller that is in Multiple Atomicity Mode for the destination namespace with an atomic write size of 8 logical blocks (e.g., as a consequence of the controller setting the NAWUN field to 8h in the Identify Namespace data structure for the NVM Command Set (refer to Figure 97)):

- if that controller processes a Copy command with 3 Source Ranges that each consist of 4 logical blocks and a destination LBA range that starts at an atomic boundary, then that controller performs 2 atomic write operations, where the first atomic write operation consists of the 8 logical blocks described by the first two Source Range entries and the second atomic write operation that consists of the 4 logical blocks described by the third Source Range entry); and
- if that controller processes a Copy command with 2 Source Ranges that each consist of 16 logical blocks and a destination LBA range that starts at an atomic boundary, then that controller performs 4 atomic write operations, 2 atomic write operations for each Source Range entry where each atomic write operation consists of half of the logical blocks described by a Source Range entry.

A controller is able to limit the implementation impact of these atomicity requirements by reporting appropriate values in the MSRC field, the MSSRL field, and the MCL field (refer to Figure 97).

If the NVMCSA bit in the ONCS field is cleared to '0', then the controller is based on an older version of this specification and the controller:

- always performs the write portion of a Copy command that has a single Source Range entry as a single write command to which the atomicity requirements specified in section 2.1.4 apply; and
- may or may not perform the write portion of a Copy command that has more than one Source Range entry as a separate write command for each Source Range entry. If the write portion of a Copy command is performed as a separate write command for each Source Range entry, then an

independent instance of the atomicity requirements in section 2.1.4 applies to copying the user data specified by each Source Range entry.

The value FFFFh for an atomicity parameter specified in section 2.1.4 (refer to Figure 4) indicates that the atomicity of any write command is 10000h logical blocks. For write commands other than the Copy command, that is the largest command size and hence indicates that the command is always atomic.

For a Copy command, 10000h logical blocks is not the largest command size and hence the value FFFFh indicates only that the atomicity of the write portion of the Copy command is 10000h logical blocks. This version of the NVM Command Set specification does not provide any means for a controller to indicate that the write portion of all Copy commands is atomic.

3.2.2.STBD3 Copy within a Single Namespace

This section applies to Copy commands that copy user data within a single namespace and use Source Range Entries Copy Descriptor Format 0h or 1h. Refer to section 3.2.2.TBD4 for Copy commands that use Source Range Entries Copy Descriptor Format 2h or 3h.

If the single namespace that is the source namespace and the destination namespace for a Copy command is formatted with protection information (PI), then the PRINFOR.PRACT bit and the PRINFOR.PRACT bit in the Copy command affect the processing of PI as follows:

- If the PRINFOR.PRACT bit and the PRINFOR.PRACT bit have the same value (i.e., both are set to '1' or both are cleared to '0'), then the controller shall perform the user data copying specified by the Copy command for each Source Range with PI processed as specified in section 5.2.2.5.TBD1; and
- If the PRINFOR.PRACT bit and the PRINFOR.PRACT bit have different values (i.e., one bit is set to '1' and one bit is cleared to '0') then the controller shall abort that Copy command with a status code of Invalid Field in Command.

If the single namespace that is the source namespace and the destination namespace for a Copy command is not formatted with PI, then the controller shall ignore the PRINFOR field and the PRINFOR field.

3.2.2.STBD4 Copy across Multiple Namespaces

This section applies to Copy commands that use Source Range Entries Copy Descriptor Format 2h or 3h to copy user data across multiple namespaces and/or within the same namespace. Refer to section 3.2.2.STBD3 for Copy commands that use Source Range Entries Copy Descriptor Formats 0h or 1h.

A controller that supports copying user data across multiple logical block namespaces (i.e., namespaces that use any I/O Command Set that specifies logical blocks) supports any attached logical block namespace as a copy source or a copy destination.

3.2.2.STBD4.1 Matching and Corresponding Formats

Copy command processing imposes restrictions on reformatting of logical block data and metadata that are copied across different namespaces. For all copy source namespaces and the copy destination namespace:

- logical block data size is required to be the same;
- metadata size is required to be the same, except that the copy destination namespace may have a different metadata size if protection information (PI) metadata is inserted or stripped as part of Copy command processing and all metadata is PI metadata; and
- PI type and format settings are required to be the same unless PI is being inserted or stripped as part of Copy command processing.

If the copy source namespaces and the copy destination namespace do not satisfy these restrictions, then the Copy command is unable to copy user data among them. As an alternative, a host is able to use Read commands and Write commands to copy user data via the host.

The specific restrictions are specified in section 3.2.2.STBD4.2, which uses the following terms for formatting restrictions on namespaces specified by a Copy command:

- **matching namespace formats for copy**: requirements that namespace format, PI type, and all other PI parameters for the namespaces be the same.
- **corresponding protection information formats for copy (corresponding PI formats for copy)**: requirements that the namespace formats differ only by the presence or absence of PI.

The specific requirements for each of these terms are specified in the remainder of this section.

Multiple namespaces that are formatted with PI have matching namespace formats for copy if each namespace uses the NVM Command Set or any other command set that both specifies logical blocks and includes the Copy command specified for the NVM Command Set (e.g., the Zoned Namespace Command Set) and all of the namespaces:

- are formatted with the same logical block data size;
- are formatted with the same metadata size; and
- have the same value for each of the following:
 - the End-to-end Data Protection Type Settings (DPS) field in the Identify Namespace data structure (refer to Figure 97);
 - the Protection Information Format Attribute (PIFA) field in the NVM Command Set I/O Command Set specific Identify Namespace data structure (refer to Figure 100);
 - the bits in the Logical Block Storage Tag Mask (LBSTM) field in the NVM Command Set I/O Command Set specific Identify Namespace data structure (refer to Figure 100) that are not ignored by the host; and
 - the following fields in the extended LBA format (refer to Figure 101) that was used to format the namespace;
 - the Protection Information Format (PIF) field;
 - the Qualified Protection Information Format (QPIF) if the PIF field is set to 11b (i.e., Qualified Type); and
 - the Storage Tag Size (STS) field.

<Note: the PIFA and QPIF fields are specified by TP 4141.>

Multiple namespaces that are formatted without PI have matching namespace formats for copy if each namespace uses the NVM Command Set or any other I/O Command Set that both specifies logical blocks and includes the Copy command specified for the NVM Command Set (e.g., the Zoned Namespace Command Set) and all of the namespaces are formatted with both the same logical block data size and the same metadata size.

Multiple namespaces where at least one of the namespaces is formatted with PI and at least one of the namespaces is formatted without PI do not have matching namespace formats for copy.

A namespace that is formatted with PI and a namespace that is formatted without PI have corresponding PI formats for copy if:

- each namespace uses either the NVM Command Set or any other I/O Command Set that both specifies logical blocks and includes the Copy command specified for the NVM Command Set (e.g., the Zoned Namespace Command Set);
- both namespaces are formatted with the same logical block data size;
- the namespace that is formatted with PI is formatted with metadata size equal to PI size (refer to section 5.2.1) (i.e., does not contain any metadata other than PI); and
- the namespace that is formatted without PI is also formatted without any metadata.

A namespace that is formatted with any metadata that is not PI metadata does not have a corresponding PI format for copy with any other namespace.

3.2.2.STBD4.2 Handling of Protection Information

Protection information (PI) and data copying functionality depends on the formats of the source and destination namespaces, the value of the PRINFOR.PRACT bit in the Copy command, and the value of the PRINFOW.PRACT bit in the Copy command as applicable. This functionality is specified for each Source Range, as the controller may process Source Ranges concurrently and in any order provided that all Copy command requirements (e.g., atomicity) are satisfied.

If the source namespace for Source Range 0 is formatted without PI, and the destination namespace is formatted without PI, then for each Source Range, including Source Range 0:

- if the source namespace for that Source Range and the destination namespace have matching namespace formats for copy, then the controller shall perform the user data copying specified by the Copy command for that Source Range; and
- if the source namespace for that Source Range and the destination namespace do not have matching namespace formats for copy, then the controller:
 - shall not copy any user data specified by that Source Range; and
 - shall abort the command with a status code of Incompatible Namespace or Format.

If the source namespace for Source Range 0 is formatted with PI, the destination namespace is formatted with PI, the PRINFOR.PRACT bit is cleared to '0', and the PRINFOW.PRACT bit is cleared to '0', then for each Source Range, including Source Range 0:

- if the source namespace for that Source Range and the destination namespace have matching namespace formats for copy, then the controller shall perform the user data copying specified by the Copy command for that Source Range with PI passed through as specified in section 5.2.2.5.TBD2; and
- if the source namespace for that Source Range and the destination namespace do not have matching namespace formats for copy, then the controller:
 - shall not copy any user data specified by that Source Range; and
 - shall abort the command with a status code of Incompatible Namespace or Format.

If the source namespace for Source Range 0 is formatted with PI, the destination namespace is formatted with PI, the PRINFOR.PRACT bit is set to '1' and the PRINFOW.PRACT bit is set to '1', then for each Source Range, including Source Range 0:

- if the source namespace for that Source Range and the destination namespace have matching namespace formats for copy, then the controller shall perform the user data copying specified by the Copy command for that Source Range with PI replaced as specified in section 5.2.2.5.TBD2; and
- if the source namespace for that Source Range and the destination namespace do not have matching namespace formats for copy, then the controller:
 - shall not copy any user data specified by that Source Range; and
 - shall abort the command with a status code of Incompatible Namespace or Format.

If the source namespace for Source Range 0 is formatted with PI, the destination namespace is formatted with PI, and the PRINFOR.PRACT bit has a different value from the PRINFOW.PRACT bit, then the controller shall abort the command with a status code of Invalid Field in Command.

If the source namespace for Source Range 0 is formatted without PI, the destination namespace is formatted with PI, and the PRINFOR.PRACT bit is set to '1', then for each Source Range, including Source Range 0:

- if the source namespace for that Source Range is formatted without PI and has a corresponding PI format for copy with the destination namespace, then the controller shall perform the user data copying specified by the Copy command for that Source Range with PI inserted as specified in section 5.2.2.5.TBD2; and
- if the source namespace for that Source Range is formatted with PI or does not have a corresponding PI format for copy with the destination namespace, then the controller:
 - shall not copy any user data specified by that Source Range; and
 - shall abort the command with a status code of Incompatible Namespace or Format.

If the source namespace for Source Range 0 is formatted without PI the destination namespace is formatted with PI, and the PRINFOR.PRACT bit is cleared to '0' then the controller shall abort the command with a status code of Incompatible Namespace or Format.

If the source namespace for Source Range 0 is formatted with PI, the destination namespace is formatted without PI, and the PRINFOR.PRACT bit is set to '1', then for each Source Range, including Source Range 0:

- if the source namespace for that Source Range is formatted with PI and has a corresponding PI format for copy with the destination namespace, then the controller shall perform the data copying specified by the Copy command for that Source Range with PI stripped as specified in section 5.2.2.5.TBD2; and
- if the source namespace for that Source Range is not formatted with PI or does not have a corresponding PI format for copy with the destination namespace, then the controller:
 - shall not copy any user data specified by that Source Range; and
 - shall abort the command with a status code of Incompatible Namespace or Format.

If the source namespace for Source Range 0 is formatted with PI, the destination namespace is formatted without PI, and the PRINFOR.PRACT bit is cleared to '0', then the controller shall abort the command with a status code of Incompatible Namespace or Format.

Modify section 3.2.2.1, renumbering it to section 3.2.2.TBD5, as shown below:

3.2.2.TBD5 4 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 (i.e., completes with a status code other than Successful Completion), then the controller may or may not have copied some of the user data and 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 cleared to 0h.

Copy command specific errors are defined in Figure 37.

Figure 37: 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 Read (PRINFOR) field or Protection Information Write (PRINFOW) field (refer to Figure 30) containing an invalid value for the Protection Information with which the namespace was formatted (refer to the PI field in the Format NVM Command section in the NVM Express Base Specification and the DPS field in Figure 97) • the ILBRT field being invalid (refer to section 5.2.2.5); or • the EILBRT field in a Source Range Entry being invalid (refer to section 5.2.2.5)
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 the Namespace Write Protection section in the NVM Express Base Specification).
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.
85h	Incompatible Namespace or Format: At least one source namespace and the destination namespace have incompatible formats (refer to section 3.2.2.STBD4).
86h	Fast Copy Not Possible: The Fast Copy Only (FCO) bit was set to '1' in a Source Range entry and the controller was not able to use fast copy operations to copy the specified data (refer to section 3.2.2.STBD1).
87h	Overlapping I/O Range: A source logical block range overlaps the destination logical block range (refer to section 3.2.2).
89h	Insufficient Resources: A resource shortage prevented the controller from performing the requested copy. The host should not retry the command on the same controller.

Modify a portion of section 4.1.5.2 as shown below:

4.1.5.2 I/O Command Set specific fields within Identify Controller data structure (CNS 01h)

...

Figure 99: Identify – Identify Controller data structure, NVM Command Set Specific Fields

Bytes	O/M ¹	Description
527:526	M	<p>Atomic Write Unit Normal (AWUN): This field indicates the size of the write operation guaranteed to be written atomically to the NVM across all namespaces with any supported namespace format during normal operation. This field is specified in logical blocks and is a 0's based value.</p> <p>If a specific namespace guarantees a larger size than is reported in this field, then this namespace specific size is reported in the NAWUN field in the Identify Namespace data structure. Refer to section 2.1.4.</p> <p>If a write command is submitted with size less than or equal to the AWUN value, the host is guaranteed that the write command is atomic to the NVM with respect to other read or write commands. If a write command is submitted with size greater than the AWUN value, then there is no guarantee of command atomicity. AWUN does not have any applicability to write errors caused by power failure (refer to Atomic Write Unit Power Fail).</p> <p>For any write command other than the Copy command, a value of FFFFh indicates all the commands are always atomic as this is the largest command size. For a Copy command, a value of FFFFh indicates that the atomicity of the write portion of the Copy command is 10000h logical blocks. It is recommended that implementations support a minimum of 128 KiB (appropriately scaled based on LBA size).</p>
...		

Modify section 5.2.2.5 as shown below:

5.2.2.5 Protection Information and Copy commands

Protection information (PI) processing during a Copy command parallels both Write and Read commands. For the portion of the Copy command that transfers data and PI ~~protection information~~ from the LBAs described by a Source Range Entry (refer to Figure 35), the PI ~~protection information~~ checks performed by the controller are controlled by the PRINFOR field in Copy command Dword 12 (refer to Figure 31) and parallels the Read command PI ~~protection information~~ checks (refer to section 5.2.2.2) as follows:

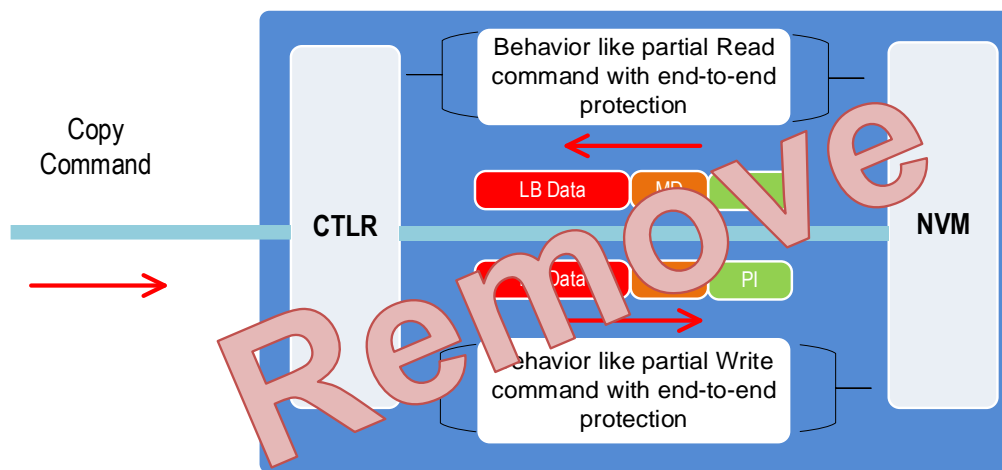
- The logical block data and metadata is transferred from the NVM to the controller.
- As the logical block data and metadata pass through the controller, the PI ~~protection information~~ within the metadata is checked. If a PI ~~protection information~~ check error is detected, the command completes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

For the portion of the Copy command that transfers data and ~~PI protection information~~ to the LBAs starting at the SDLBA field (refer to Figure 30), the ~~PI protection information~~ operations performed by the controller are controlled by the PRINFOW field in Copy command Dword 12 (refer to Figure 31) and parallels the Write command ~~PI protection information~~ checks (refer to section 5.2.2.1) as follows:

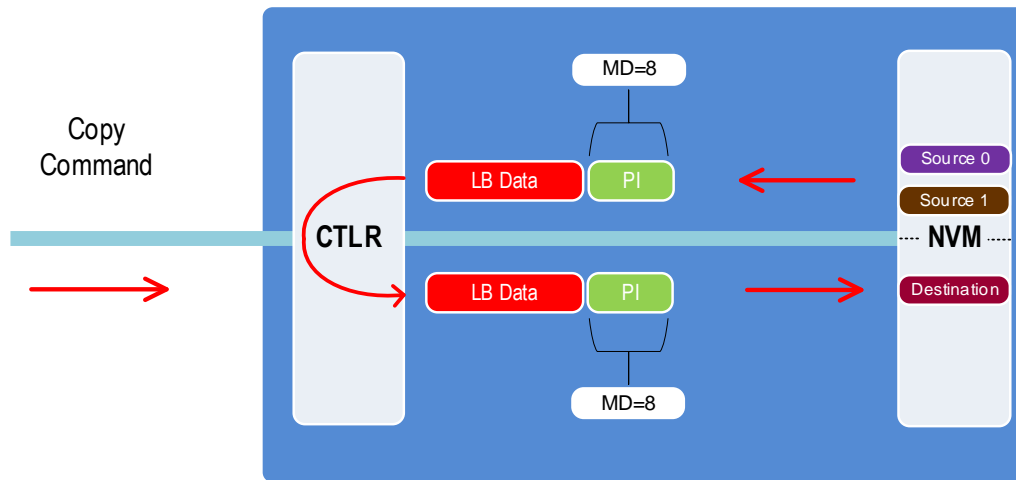
- The logical block data and metadata are transferred from the controller to the NVM.
- As the logical block data and metadata passes through the controller, the ~~PI protection information~~ is handled as described in section 5.2.2.1

~~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 PRINFOW field, then the Copy command shall be aborted with a status code of Invalid Field in Command.~~

Figure 136: PI Processing for Copy MD=8 Pass-through
Copy

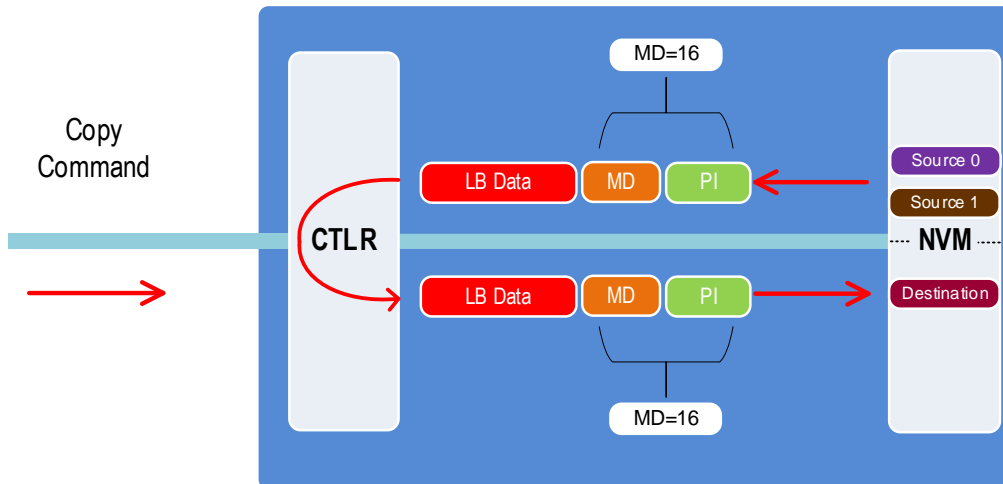


~~Protection Information with PRACT bit cleared to '0' (i.e., pass)~~



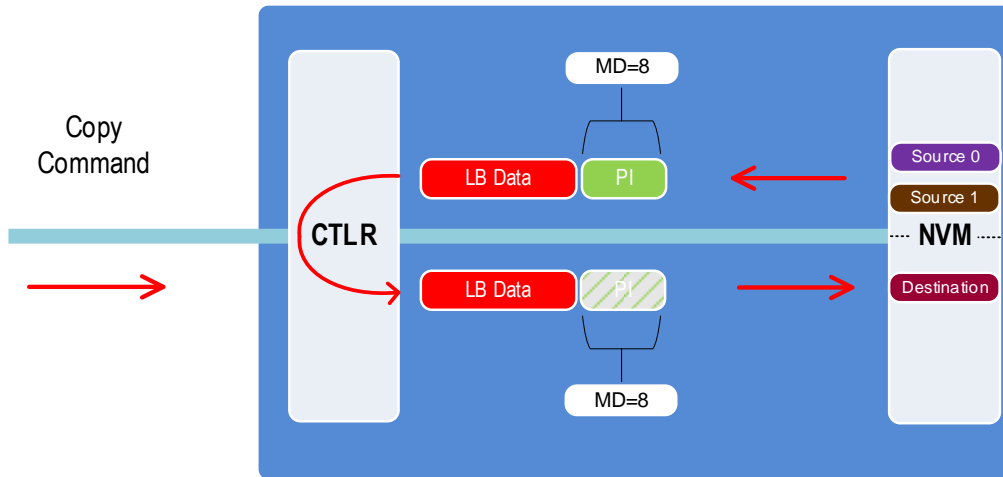
MD=8, PI, PRINFOR.PRACT=PRINFOW.PRACT=0 : PI pass-through

Figure TBDda1: PI Processing for Copy MD=16 Pass-through



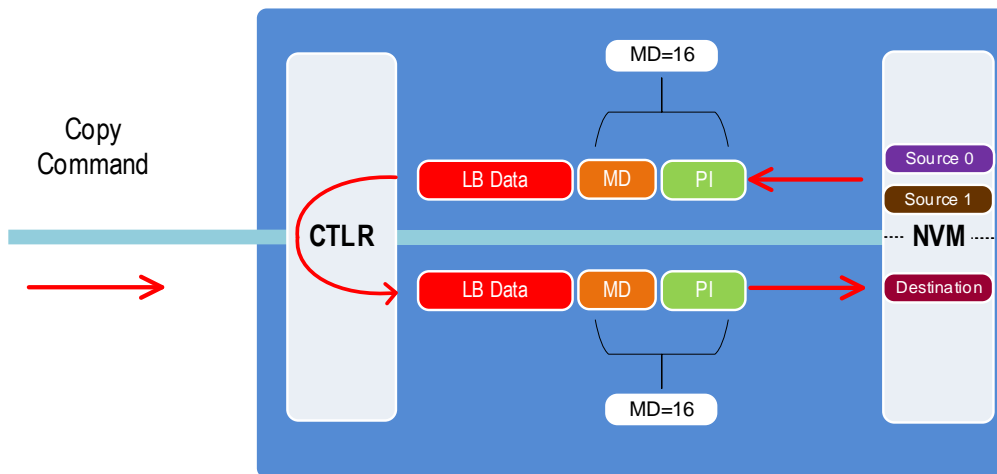
MD=16, PI, PRINFOR.PRACT=PRINFOW.PRACT=0 : PI pass-through

Figure TBDda2: PI Processing for Copy MD=8 Replace



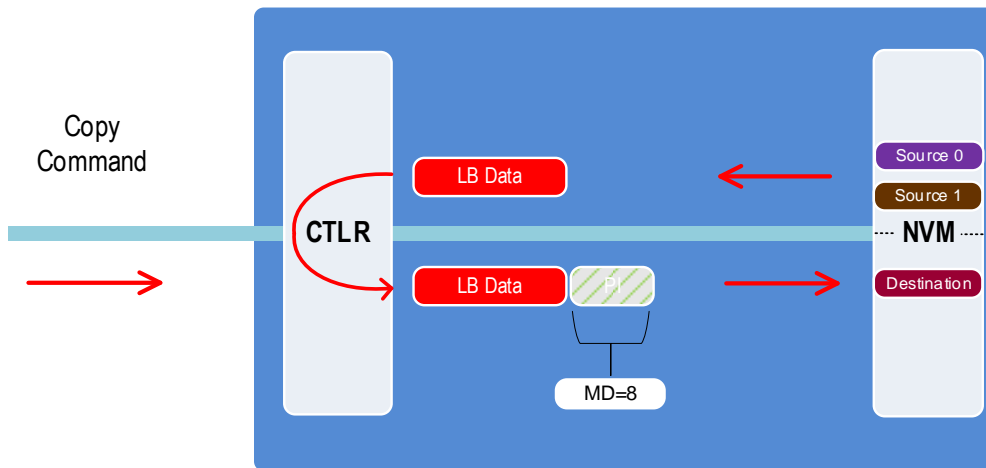
MD=8, PI, PRINFOR.PRACT=PRINFOW.PRACT=1 : PI replace

Figure TBDda3: PI Processing for Copy MD=16 Replace



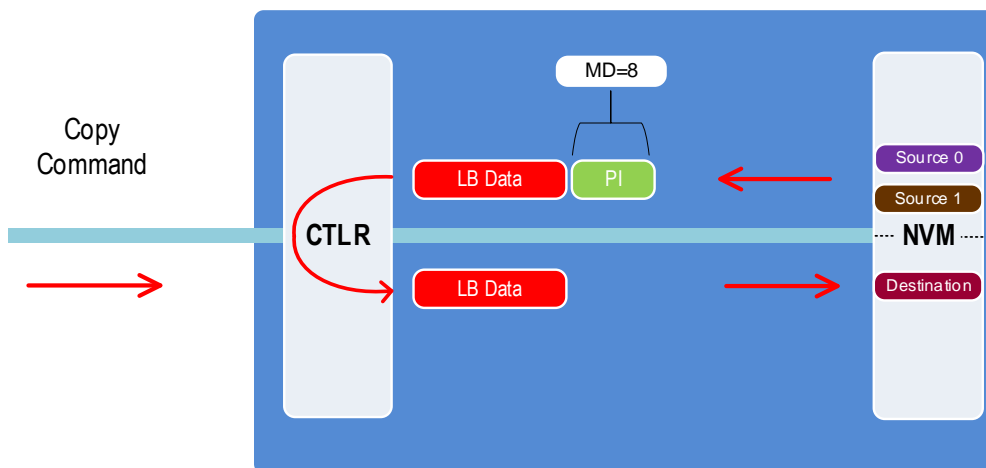
MD=16, PI, PRINFOR.PRACT=PRINFOW.PRACT=1 : PI replace

Figure TBDda4: PI Processing for Copy MD=8 Insert



destination MD=8, PI, PRINFOW.PRACT=1: PI insert

Figure TBDda5: PI Processing for Copy MD=8 Strip



source MD=8, PI, PRINFOR.PRACT=1: PI strip

Figure 136, Figure TBDda1, Figure TBDda2, Figure TBDda3, Figure TBDda4, and Figure TBDda5 shows six examples of PI processing for the Copy command where the Copy command in each example copies from two source regions in two different source namespaces to a destination region in a third destination namespace.

While user 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).

5.2.2.5.TBD1 Protection Information and copying within the same namespace

If a Copy command uses Source Range Entries Copy Descriptor Format 0h or 1h to request that user data in a Source Range be copied to the same namespace and that namespace is formatted with protection information (PI), then that user data is copied only if the PRINFOR.PRACT bit and the PRINFOW.PRACT bit in the Copy command have the same value (refer to section 3.2.2.STBD3):

- If the PRINFOR.PRACT bit and the PRINFOW.PRACT bit are both set to '1', then as the logical block data and metadata pass through the controller, the PI is replaced with PI that is generated by the controller as shown for the examples in Figure 136 and Figure TBDda1; and
- If the PRINFOR.PRACT bit and the PRINFOW.PRACT bit are both cleared to '0', then the logical block data and metadata pass through the controller without change to the PI as shown for the examples in Figure TBDda2 and Figure TBDda3.

If the PRINFOR.PRACT bit and the PRINFOW.PRACT bit do not have the same value, then the Copy command is aborted with a status code of Invalid Field in Command (refer to section 3.2.2.STBD3).

PI checks for copied user data are performed as described in section 5.2.2.5.

5.2.2.5.TBD2 Protection Information and copying across different namespaces

This section applies to Copy commands that use either Source Range Entries Copy Descriptor format 2h or 3h to copy data across multiple namespaces and/or within the same namespace. Refer to section 5.2.2.5.TBD1 for Copy commands that use Source Range Entries Copy Descriptor Format 0h or 1h.

Controller processing of protection information (PI) as part of a copy operation depends on the format of the destination namespace, the format of each source namespace and the values of the PRINFOR.PRACT bit and the PRINFOW.PRACT bit in the Copy command.

If the destination namespace and a source namespace are both formatted with PI and the two namespaces have matching namespace formats for copy (refer to section 3.2.2.STBD4.1), then:

- if the PRINFOR.PRACT bit and the PRINFOW.PRACT bit are both set to '1', then as the logical block data and metadata pass through the controller, the PI is overwritten with PI that is generated by the controller as shown for the examples in Figure 136 and Figure TBDda1; and
- if the PRINFOR.PRACT bit and the PRINFOW.PRACT bit are both cleared to '0' then the logical block data and metadata pass through the controller without change as shown for the examples in Figure TBDda2 and Figure TBDda3.

All other cases in which the destination namespace and a source namespace are both formatted with PI result in the controller aborting the Copy command with an error status (refer to section 3.2.2.STBD4.2).

If the destination namespace is formatted with PI, a source namespace is formatted without PI, and:

- the two namespaces have corresponding PI formats for copy (refer to section 3.2.2.STBD4.1); and
- the PRINFOW.PRACT bit is set to '1',

then as the logical block data passes through the controller, the controller generates and appends PI to the end of the logical block data as shown for the example in Figure TBDda4. All other cases in which the destination namespace is formatted with PI and a source namespace is formatted without PI result in the controller aborting the Copy command with an error status (refer to section 3.2.2.STBD4.2), including cases where the metadata size of the destination namespace is larger than the PI size of that namespace.

If the destination namespace is formatted without PI, a source namespace is formatted with PI, and:

- the two namespaces have corresponding PI formats for copy; and
- the PRINFOR.PRACT bit is set to '1',

then as the logical block data and metadata pass through the controller, the PI is stripped, and the controller only writes the logical block data to the NVM as shown for the example in [Figure TBDda5](#). All other cases in which the destination namespace is formatted without PI and a source namespace is formatted with PI result in the controller aborting the Copy command with an error status (refer to section [3.2.2.STBD4.2](#)), including cases where the metadata size of a source namespace is larger than the PI size of that namespace.

PI checks for copied user data are performed as described in section [5.2.2.5](#).

Modify Section 5.5 as shown below:

5.5 Reservations

Reservations operate as defined in the NVM Express Base Specification with the additional I/O Command Set specific Command Behavior in the Presence of a Reservation defined in Figure 137.

Figure 137: 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
Copy Command Group								
Copy	C	C	C	C	C	A	C	A
Read Command Group								
Compare								
Copy (source) ¹	A	A	C	C	A	A	C	A
Read								
Verify								
Write Command Group								
Copy (destination) ¹								
Dataset Management								
Write	C	C	C	C	C	A	C	A
Write Uncorrectable								
Write Zeroes								
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. Each source namespace of a Copy command is checked for reservation conflict as if accessed by a read command and the destination namespace of a Copy command is checked for reservation conflict as if accessed by a write command.								

<Editor's Note: The ZNS Command Set spec incorporates the above table by reference, so no change is needed to the Reservation material there.>

Description of Specification Changes for NVM Express Base Specification 2.0c:

Modify Figure 96 in section 3.3.3.2.1.2 as shown below:

Figure 96: Status Code – Command Specific Status Values, I/O Commands

Value	Description
80h	Conflicting Attributes
81h	Invalid Protection Information
82h	Attempted Write to Read Only Range
83h	Command Size Limit Exceeded
84h	Reserved <Editor's Note: Assigned to a different TP>
85h	Incompatible Namespace or Format
86h	Fast Copy Not Possible
87h	Overlapping I/O Range
88h	Namespace Not Reachable
89h	Insufficient Resources
84h 8Ah to B7h	Reserved
B8h	Zoned Boundary Error
B9h	Zone Is Full
BAh	Zone Is Read Only
BBh	Zone Is Offline
BCh	Zone Invalid Write
BDh	Too Many Active Zones
BEh	Too Many Open Zones
BFh	Invalid Zone State Transition
NOTES:	
1. A = All I/O Command Sets, C = Command Set Specific.	

Modify Figure 275 in section 5.17.2.1 as shown below:

Figure 275: Identify – Identify Controller Data Structure, I/O Command Set Independent

Bytes	I/O ¹	Admin ¹	Disc ¹	Description
...				

521:520	M	M	R	Optional NVM Command Support (ONCS): This field indicates the optional I/O commands and features supported by the controller. Refer to section 3.1.2.	
				Bit	Description
				15:911	Reserved
				10	NVM All Fast Copy (NVMAFC): If set to '1', then within this NVM subsystem, all copy operations performed by the controller for NVM Command Set Copy commands are fast copy operations (refer to the Fast copy operations section of the NVM Command Set Specification). If cleared to '0', then within this NVM subsystem, some copy operations performed by the controller for NVM Command Set Copy commands may not be fast copy operations. This bit is ignored by the host if the controller does not support the NVM Command Set Copy command.
				9	NVM Copy Single Atomicity (NVMCSA): If set to '1', then the write portion of any NVM Command Set Copy command is performed as a single write command to which the atomicity requirements described in the Atomic operation section of the NVM Command Set Specification apply. If cleared to '0', then the atomicity behavior of the write portion of Copy commands is specified by the Copy atomicity section of the NVM Command Set Specification. This bit should be set to '1' if the controller supports the NVM Command Set Copy command. The Copy atomicity section of the NVM Command Set Specification specifies additional conditions under which this bit shall be set to '1'. This bit is ignored by the host if the controller does not support the NVM Command Set Copy command.
				8	Bit 8 if If set to '1', then the controller supports the NVM Command Set Copy command. If cleared to '0', then the controller does not support the NVM Command Set Copy command.
				7	Bit 7 if If set to '1', then the controller supports the NVM Command Set Verify command and the Verify Size Limit (VSL) field indicates the recommended maximum data size for Verify commands. If cleared to '0', then controller support of the NVM Command Set Verify command is indicated by a non-zero data size limit in the VSL field.
				6	Bit 6 if 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 5.27.1.11.
				5	Bit 5 if 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 8.19 for additional requirements.
				4	Bit 4 if 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.
				3	Bit 3 if If set to '1', then the controller supports the NVM Command Set Write Zeroes command and the Write Zeroes Size Limit (WZSL) field indicates the recommended maximum data size for Write

					Zeroes commands. If cleared to '0', then controller support of the NVM Command Set Write Zeroes command is indicated by a non-zero data size limit in the WZSL field.
			2		Bit 2 if If set to '1', then the controller supports the NVM Command Set Dataset Management command and limits, if any, on controller support of the Dataset Management command are indicated by non-zero values in the Dataset Management Ranges Limit (DMRL) field, the Dataset Management Size Limit (DMSL) field and the Dataset Management Range Size Limit (DMRSL) field. If cleared to '0', then controller support of the NVM Command Set Dataset Management command is indicated by a non-zero data size limit in the DMRL, DMSL, and DMRSL fields.
			1		Bit 1 if If set to '1', then the controller supports the NVM Command Set Write Uncorrectable command and the Write Uncorrectable Size Limit (WUSL) field indicates the recommended maximum data size for Write Uncorrectable commands. If cleared to '0', then controller support of the NVM Command Set Write Uncorrectable command is indicated by a non-zero data size limit in the WUSL field.
			0		Bit 0 if If set to '1', then the controller supports the NVM Command Set Compare command. If cleared to '0', then the controller does not support the NVM Command Set Compare command.
					<p>Bits 15:9 are reserved.</p> <p>Bit 8 if set to '1', then the controller supports the NVM Command Set Copy command. If cleared to '0', then the controller does not support the NVM Command Set Copy command.</p> <p>Bit 7 if set to '1', then the controller supports the NVM Command Set Verify command and the Verify Size Limit (VSL) field indicates the recommended maximum data size for Verify commands. If cleared to '0', then controller support of the NVM Command Set Verify command is indicated by a non-zero data size limit in the VSL field.</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 5.27.1.11.</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 8.19 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 NVM Command Set Write Zeroes command and the Write Zeroes Size Limit (WZSL) field indicates the recommended maximum data size for Write Zeroes commands. If cleared to '0', then controller support of the NVM Command Set Write Zeroes command is indicated by a non-zero data size limit in the WZSL field.</p> <p>Bit 2 if set to '1', then the controller supports the NVM Command Set Dataset Management command and limits, if any, on controller support of the Dataset Management command are indicated by non-zero values in the Dataset Management Ranges Limit (DMRL) field, the Dataset Management Size Limit (DMSL) field and the Dataset Management Range Size Limit (DMRSL) field. If cleared to '0', then controller support of the NVM Command Set Dataset Management command is indicated by a non-zero data size limit in the DMRL, DMSL, and DMRSL fields.</p> <p>Bit 1 if set to '1', then the controller supports the NVM Command Set Write Uncorrectable command and the Write Uncorrectable Size Limit (WUSL) field indicates the recommended maximum data size for Write Uncorrectable commands. If cleared to</p>

				<div>'0', then controller support of the NVM Command Set Write Uncorrectable command is indicated by a non-zero data size limit in the WUSL field.</div> <div>Bit 0 if set to '1', then the controller supports the NVM Command Set Compare command. If cleared to '0', then the controller does not support the NVM Command Set Compare command.</div> <div>NOTE: This field applies to all I/O Command Sets. The original name has been retained for historical continuity.</div>												
...																
535:534	M	R	R	Copy Descriptor Formats Supported:												
				<table><tr><th>Bits</th><th>Description</th></tr><tr><td>15:42</td><td>Reserved</td></tr><tr><td>3</td><td>If set to '1', then the controller supports Copy Descriptor Format 3h. If cleared to '0', then the controller does not support Copy Descriptor Format 3h.</td></tr><tr><td>2</td><td>If set to '1', then the controller supports Copy Descriptor Format 2h. If cleared to '0', then the controller does not support Copy Descriptor Format 2h.</td></tr><tr><td>1</td><td>If set to '1', then the controller supports Copy Descriptor Format 1h. If cleared to '0', then the controller does not support Copy Descriptor Format 1h.</td></tr><tr><td>0</td><td>If set to '1', then the controller supports Copy Descriptor Format 0h. If cleared to '0', then the controller does not support Copy Descriptor Format 0h.</td></tr></table>	Bits	Description	15:42	Reserved	3	If set to '1', then the controller supports Copy Descriptor Format 3h. If cleared to '0', then the controller does not support Copy Descriptor Format 3h.	2	If set to '1', then the controller supports Copy Descriptor Format 2h. If cleared to '0', then the controller does not support Copy Descriptor Format 2h.	1	If set to '1', then the controller supports Copy Descriptor Format 1h. If cleared to '0', then the controller does not support Copy Descriptor Format 1h.	0	If set to '1', then the controller supports Copy Descriptor Format 0h. If cleared to '0', then the controller does not support Copy Descriptor Format 0h.
				Bits	Description											
				15:42	Reserved											
				3	If set to '1', then the controller supports Copy Descriptor Format 3h. If cleared to '0', then the controller does not support Copy Descriptor Format 3h.											
				2	If set to '1', then the controller supports Copy Descriptor Format 2h. If cleared to '0', then the controller does not support Copy Descriptor Format 2h.											
1	If set to '1', then the controller supports Copy Descriptor Format 1h. If cleared to '0', then the controller does not support Copy Descriptor Format 1h.															
0	If set to '1', then the controller supports Copy Descriptor Format 0h. If cleared to '0', then the controller does not support Copy Descriptor Format 0h.															
...																

Modify Figure 351 in Section 5.27.1.18 as shown below:

Figure 351: Host Behavior Support – Data Structure

Bytes	Description										
...	...										
02	LBA Format Extension Enable (LBAFEE): I/O Command Set specific definition. Refer to the applicable I/O Command Set specification for details. All values other than 0h and 1h are reserved.										
03	Reserved										
05:04	Copy Descriptor Formats Enable (CDFE): The bits in this field that are used (i.e., are neither reserved nor unused) enable the corresponding Copy Descriptor Formats (e.g., bit 2 enables Copy Descriptor Format 2h). Bits in this field that this specification defines as unused (i.e., bits 1:0) correspond to Copy Descriptor Formats (refer to the NVM Command Set Specification) that are always enabled if they are supported. <table border="1"> <thead> <tr> <th>Bits</th><th>Description</th></tr> </thead> <tbody> <tr> <td>15:4</td><td>Reserved</td></tr> <tr> <td>3</td><td>Copy Descriptor Format 3h Enable (CDF3E): If set to '1' and the controller supports Copy Descriptor Format 3h, then Copy Descriptor Format 3h is enabled. If cleared to '0', or the controller does not support Copy Descriptor Format 3h, then Copy Descriptor Format 3h is disabled.</td></tr> <tr> <td>2</td><td>Copy Descriptor Format 2h Enable (CDF2E): If set to '1' and the controller supports Copy Descriptor Format 2h, then Copy Descriptor Format 2h is enabled. If cleared to '0', or the controller does not support Copy Descriptor Format 2h, then Copy Descriptor Format 2h is disabled.</td></tr> <tr> <td>1:0</td><td>Unused. The controller shall ignore these bits when processing a Set Features command that specifies this Feature and shall clear these bits to '0' in the attributes returned in the data buffer for a Get Features command that specifies this Feature.</td></tr> </tbody> </table>	Bits	Description	15:4	Reserved	3	Copy Descriptor Format 3h Enable (CDF3E): If set to '1' and the controller supports Copy Descriptor Format 3h, then Copy Descriptor Format 3h is enabled. If cleared to '0', or the controller does not support Copy Descriptor Format 3h, then Copy Descriptor Format 3h is disabled.	2	Copy Descriptor Format 2h Enable (CDF2E): If set to '1' and the controller supports Copy Descriptor Format 2h, then Copy Descriptor Format 2h is enabled. If cleared to '0', or the controller does not support Copy Descriptor Format 2h, then Copy Descriptor Format 2h is disabled.	1:0	Unused. The controller shall ignore these bits when processing a Set Features command that specifies this Feature and shall clear these bits to '0' in the attributes returned in the data buffer for a Get Features command that specifies this Feature.
Bits	Description										
15:4	Reserved										
3	Copy Descriptor Format 3h Enable (CDF3E): If set to '1' and the controller supports Copy Descriptor Format 3h, then Copy Descriptor Format 3h is enabled. If cleared to '0', or the controller does not support Copy Descriptor Format 3h, then Copy Descriptor Format 3h is disabled.										
2	Copy Descriptor Format 2h Enable (CDF2E): If set to '1' and the controller supports Copy Descriptor Format 2h, then Copy Descriptor Format 2h is enabled. If cleared to '0', or the controller does not support Copy Descriptor Format 2h, then Copy Descriptor Format 2h is disabled.										
1:0	Unused. The controller shall ignore these bits when processing a Set Features command that specifies this Feature and shall clear these bits to '0' in the attributes returned in the data buffer for a Get Features command that specifies this Feature.										
511:0306	Reserved										

Modify Figure 473 in Section 8.19.1 as shown below:

Figure 473: 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
Copy Command Group								
I/O Command Set specific Copy Commands²	C	C	C	C	C	A	C	A
Read Command Group								
Security Receive (Admin) I/O Command Set specific Copy Commands (source) ^{2,3} I/O Command Set specific Read Commands ²	A	A	C	C	A	A	C	A
Write Command Group								
Capacity Management (Admin) Flush Format NVM (Admin) Namespace Attachment (Admin) Namespace Management (Admin) Sanitize (Admin) Security Send (Admin) I/O Command Set specific Copy Commands (destination) ^{2,3} I/O Command Set specific Write Commands ²	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
<p>Key: A definition: A=Allowed, command processed normally by the controller C definition: C=Conflict, command aborted by the controller with a status code of Reservation Conflict</p> <p>Notes: 1. The behavior of a vendor specific command is vendor specific. 2. Refer to the applicable I/O Command Set specification 3. For an I/O Command Set specific Copy command, each source namespace is checked for reservation conflict as if accessed by a read command and the destination namespace is checked for reservation conflict as if accessed by a write command, as described in the applicable I/O command Set specification.</p>								