

Date: April 23, 2025

To: T10 Committee

From: Brad Besmer, Broadcom

Subject: T10/25-023 SNT Add several SBC-6 translations

## Overview

This proposal adds translations for the following commands defined in SBC-6:

- FORMAT UNIT
- READ
- READ CAPACITY
- SANITIZE
- START STOP UNIT
- SYNCHRONIZE CACHE
- UNMAP
- VERIFY
- WRITE
- WRITE LONG
- WRITE SAME

## Revision History

R0

- Initial Revision

## SNT Changes

### 3.2.1 Abbreviations

...

**EdNote:** Discuss if we want to introduce this abbreviation as it would affect also adding normative reference

UBM    Uniform Backplane Management (see SFF-TA-1005)

Ednote: New subclauses in the clause that adds the other NVMe capability definitions were added in 24-060. Same rationale as presented in the WZ Supported editors' note.

## 9.5 Verify Supported

Verify Supported is an internal variable that indicates if the NVMe device supports the Verify command.

Verify Supported shall be set to true:

- a) if, in the NVMe Identify Controller data structure (CNS 01h) ONCS field (see NVMeBASE), the NVMVFYS bit is set to 1b; or
- b) if, the NVMe I/O Command Set Specific Identify Controller data structure (CNS 06h, CSI 00h) VSL field (see NVMCS) is set to a non-zero value;

Otherwise, Verify Supported shall be set to false.

## 9.6 WU Supported

WU Supported is an internal variable that indicates if the NVMe device supports the Write Uncorrectable command.

WU Supported shall be set to true:

- a) if, in the NVMe Identify Controller data structure (CNS 01h) ONCS field (see NVMeBASE), the NVMWUSV bit is set to 1b; or
- b) if, the NVMe I/O Command Set Specific Identify Controller data structure (CNS 06h, CSI 00h) WUSL field (see NVMCS) is set to a non-zero value;

otherwise, WU Supported shall be set to false.

## 9 SBC-6 command mapping

This clause describes the SCSI to NVMe translation for the SCSI Block Commands (SBC-6).

Table 1 summarizes the SCSI command translations for these commands.

*Table 1 SBC-6 Command translations*

SCSI Command	NVMe commands <sup>a</sup>	Reference
FORMAT UNIT	Format NVM	9.1
READ	Read	9.2
READ CAPACITY		9.3
SANITIZE	Sanitize and Get Log Page	9.4
START STOP UNIT	Flush and Set Features	9.5
SYNCHRONIZE CACHE	Flush	9.6
UNMAP	Dataset Management	9.7
VERIFY	Verify and Compare	9.8
WRITE	Write	9.9
WRITE LONG	Write Uncorrectable	9.10
WRITE SAME	Write and Write Zeros	9.11
a Translations for SCSI commands may require one or more of the NVMe commands listed to be sent to the NVMe device.		

### 9.1 FORMAT UNIT command

The FORMAT UNIT command is used to low level format all LBAs accessible to the SCSI application client.

If the NVMe Identify Controller data structure (CNS 01h) FNVMS bit (see NVMeBASE) is set to 0b (i.e., Format NVM not supported), then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID COMMAND OPERATION CODE.

The SNTL shall translate a FORMAT UNIT command to an NVMe Format NVM command and set the Command Specific fields (i.e., Dwords 10-15) in the NVMe Format NVM command not specified by the translation to zero, including the Protection Information fields.

**EdNote: Should the preceding paragraph be made generic and put into an earlier clause?**

Table 2 shows the translation for the fields of the FORMAT UNIT CDB.

*Table 2 FORMAT UNIT CDB field translations*

Field	Description
OPERATION CODE	Set to 04h

Field	Description
FMTPINFO	<p>If this field is non-zero and:</p> <ul style="list-style-type: none"> <li>a) the SNTL does not support protection information; or</li> <li>b) the FMTDATA bit is set to 0b;</li> </ul> <p>then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.</p> <p>See 9.1.3</p>
LONGLIST	See SBC-6
FMTDATA	See 9.1.1
CMPLIST	<p>If a complete list is specified (i.e., the FMTDATA bit is set to 1b and the CMPLIST bit is set to 1b), then the SNTL shall terminate the command with a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.</p>
DEFECT LIST FORMAT	The SNTL shall ignore this field.
FFMT	The SNTL shall ignore this field.
CONTROL	See 5.1.y.3

The SNTL shall process commands received during the processing of the FORMAT UNIT command as specified in SBC-6.

### 9.1.1 FORMAT UNIT parameter list

If the FORMAT UNIT command CDB specifies a FMTDATA bit set to 1b, then the SNTL shall accept a FORMAT UNIT parameter list consisting of a short parameter list header or a long parameter list header.

Table 3 defines the SNTL handling of fields in the FORMAT UNIT parameter list.

*Table 3 FORMAT UNIT parameter list translation*

Field	Description
PROTECTION FIELD USAGE	See 9.1.3
FOV	The SNTL shall ignore this field.
DPRY	The SNTL shall ignore this field.
DCRT	The SNTL shall ignore this field.
STPF	The SNTL shall ignore this field.
IP	The SNTL shall ignore this field.
IMMED	<p>If set to 1b, then the SNTL shall validate all CDB and parameter list fields, if any, and return GOOD status before issuing the NVMe Format NVM command.</p> <p>If set to 0b, then the SNTL shall wait for completion of all commands issued to the NVMe Controller before returning status.</p>
P_I_INFORMATION	<p>If this field is non-zero, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST.</p>

Field	Description
PROTECTION INTERVAL EXPONENT	If this field is non-zero, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST.
DEFECT LIST LENGTH	The SNTL shall ignore this field.
Initialization pattern descriptor (if any)	The SNTL shall ignore this field.
Defect list (if any)	The SNTL shall ignore this field.

### 9.1.2 FORMAT UNIT command processing

To process the FORMAT UNIT command, the SNTL shall:

- 1) if the IMMED bit is set to 1b and no CHECK CONDITION status was returned as a result of processing the fields in *Table 2* and *Table 3*, then return GOOD status;
- 2) find a matching LBA Format in the NVMe Identify Namespace data structure (CNS 00h) (see NVMCS), with:
  - a. 2<sup>^</sup>LBADS field set to the LOGICAL BLOCK LENGTH field in the Mode parameter block descriptor (see 10.4.X); and
  - b. the MS field (i.e., Metadata Size) set to:
    - a. zero, if the FMTPINFO field is set to zero; or
    - b. eight, if the FMTPINFO field is set to a non-zero value;
- 3) if no matching LBA Format is found, then terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB;
- 4) if no CHECK CONDITION status was returned in the previous steps, issue an NVMe Format NVM command with:
  - a. the LBAFU field set to bits 5:4 of the index of the matching LBA Format;
  - b. the LBAFL field set to bits 3:0 of the index of the matching LBA Format;
  - c. the MSET bit set to 0b (i.e., metadata transferred as part of separate buffer);
  - d. the PI field set to the resulting Protection Information Type (see 9.1.3);
  - e. the PIL bit set to 0b;
  - f. the SES field set to 000b; and
  - g. all other fields set to zero.
- 5) if the NVMe Format NVM command completes with an error and the IMMED bit is set to 0b, then terminate the command with CHECK CONDITION status with the sense key set to MEDIUM ERROR, and the additional sense code set to FORMAT COMMAND FAILED.

If an error occurs during the processing of the FORMAT UNIT command and the IMMED bit is set to 1b, then the SNTL shall establish a deferred error condition (see SPC-7) with the sense key set to MEDIUM ERROR, and the additional sense code set to FORMAT COMMAND FAILED.

**EdNote:** Consider future extension to define an NVMe-Specific Mode page that could be used to determine the Format NVM - MSET field setting.

### 9.1.3 Determine Protection Information Type

The SNTL determines the requested Protection Information Type as defined by *Table 4*. If the resulting Protection Information Type is Invalid or the corresponding Protection Information Type bit in the NVMe Identify Namespace data structure (CNS 00h) DPC field (see NVMCS) is set to 0b, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB or INVALID FIELD IN PARAMETER LIST.

*Table 4 Protection Information Type*

FMTPIFNO	PROTECTION FIELD USAGE	Protection Information Type
00b	000b	0
10b	000b	1
11b	000b	2
11b	001b	3
All others		Invalid

## 9.2 READ commands

This subclause applies to the translation of SCSI READ commands.

If the SNTL terminates a READ command with CHECK CONDITION status with the sense key set to a value other than ILLEGAL REQUEST while processing the command, then the SATL may transfer a vendor specific amount of data before terminating the command. If any data is transferred before terminating the command, the sense key shall be set to a value other than ILLEGAL REQUEST.

The SNTL shall translate a READ command to an NVMe Read command and set the Command Specific fields (i.e., Dwords 10-15) in the NVMe Read command not specified by the translation to zero, including the Protection Information fields.

### 9.2.1 READ (10) command

The READ (10) command requests that the device transfer logical blocks of user data to the application client (see SBC-6). The SNTL shall translate the READ (10) command to the NVMe Read command.

*Table 5* shows the translation for the fields of the READ (10) CDB.

*Table 5 READ (10) CDB field translations*

Field	Description
OPERATION CODE	Set to 28h
RDPROTECT	See 9.2.2
DPO	The SNTL shall ignore this bit.
FUA	The SNTL shall set the FUA bit in the NVMe Read command to the value in this field.
RARC	The SNTL shall ignore this bit.
LOGICAL BLOCK ADDRESS	<p>The SNTL shall set the SLBA field in the NVMe Read command to the value in this field.</p> <p>The SNTL shall set the ELBTL field in the NVMe Read command to bits 31:0 of the value in this field.</p>

---

Field	Description
GROUP NUMBER	The SNTL shall ignore this field.
TRANSFER LENGTH	If this field is set to zero, then the SNTL shall complete the READ (10) command with GOOD status and shall not issue an NVMe Read command. If this field is set to non-zero, then the SNTL shall set the NLB field in the NVMe Read command to the value in this field less one.
CONTROL	See 5.1.y.3

## 9.2.2 RDPROTECT

If the RDPROTECT field is non-zero, then:

- if the SNTL does not support protection information or the NVMe namespace is not formatted for end-to-end protection information, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB; or
- if the SNTL supports protection information and the NVMe namespace is formatted for end-to-end protection information, then the SNTL shall translate the RDPROTECT field as defined in *Table 6*.

*Table 6 RDPROTECT field translations*

Protection Type	RDPROTECT	NVMe PRACT	NVMe PRCHK		
			Bit 2 (Guard)	Bit 1 (App Tag)	Bit 0 (Ref Tag)
1	000b	1b (i.e., stripped)	0b	0b	0b
	001b	0b (i.e., passed)	1b	0b	1b
	010b	0b (i.e., passed)	0b	0b	1b
	011b	0b (i.e., passed)	0b	0b	0b
	100b	0b (i.e., passed)	1b	0b	0b
	101b	0b (i.e., passed)	1b	0b	1b
2	000b	1b (i.e., stripped)	0b	0b	0b
	001b	0b (i.e., passed)	1b	1b	1b
	010b	0b (i.e., passed)	0b	1b	1b
	011b	0b (i.e., passed)	0b	0b	0b
	100b	0b (i.e., passed)	1b	0b	0b
	101b	0b (i.e., passed)	1b	1b	1b
3	000b	1b (i.e., stripped)	0b	0b	0b
	001b	0b (i.e., passed)	1b	0b	0b
	010b	0b (i.e., passed)	0b	0b	0b
	011b	0b (i.e., passed)	0b	0b	0b
	100b	0b (i.e., passed)	1b	0b	0b
	101b	0b (i.e., passed)	1b	0b	0b
Any	110b	The SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB			
	111b				

## 9.2.3 READ (12) command

The READ (12) command requests that the device transfer logical blocks of user data to the application client (see SBC-6). The SNTL shall translate the READ (12) command to the NVMe Read command.

*Table 5* shows the translation for the fields of the READ (12) CDB.

*Table 7 READ (12) CDB field translations*

Field	Description
OPERATION CODE	Set to:A8h
RDPROTECT	See 9.2.2
DPO	The SNTL shall ignore this bit.
FUA	The SNTL shall set the FUA bit in the NVMe Read command to the value in this field.
RARC	The SNTL shall ignore this bit.
LOGICAL BLOCK ADDRESS	The SNTL shall set the SLBA field in the NVMe Read command to the value in this field. The SNTL shall set the ELBTL field in the NVMe Read command to bits 31:0 of the value in this field.
GROUP NUMBER	The SNTL shall ignore this field.
TRANSFER LENGTH	If this field is set to zero, then the SNTL shall complete the READ (12) command with GOOD status and shall not issue an NVMe Read command. If this field is set to non-zero, then the SNTL shall set the NLB field in the NVMe Read command to the value in this field less one.
CONTROL	See 5.1.y.3

## 9.2.4 READ (16) command

The READ (16) command requests that the device transfer logical blocks of user data to the application client (see SBC-6). The SNTL shall translate the READ (16) command to the NVMe Read command.

Table 5 shows the translation for the fields of the READ (16) CDB.

*Table 8 READ (16) CDB field translations*

Field	Description
OPERATION CODE	Set to A8h
RDPROTECT	See 9.2.2
DPO	The SNTL shall ignore this bit.
FUA	The SNTL shall set the FUA bit in the NVMe Read command to the value in this field.
RARC	The SNTL shall ignore this bit.
DLD2	The SNTL shall ignore this bit.
LOGICAL BLOCK ADDRESS	The SNTL shall set the SLBA field in the NVMe Read command to the value in this field. The SNTL shall set the ELBTL field in the NVMe Read command to bits 31:0 of the value in this field.
DLD1	The SNTL shall ignore this bit.
DLD0	The SNTL shall ignore this bit.
GROUP NUMBER	The SNTL shall ignore this field.
TRANSFER LENGTH	If this field is set to zero, then the SNTL shall complete the READ (16) command with GOOD status and shall not issue an NVMe Read command. If this field is set to non-zero, then the SNTL shall set the NLB field in the NVMe Read command to the value in this field less one.
CONTROL	See 5.1.y.3

## 9.2.5 READ (32) command

The READ (32) command requests that the device perform the actions defined for the READ (10) command with extensions for type 2 protection.

If the SNTL does not support protection information or the NVMe namespace is not formatted for type 2 end-to-end protection information, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID COMMAND OPERATION CODE.

Table 9 shows the translation for the fields of the READ (32) CDB.

*Table 9 READ (32) CDB field translations*

Field	Description
OPERATION CODE	Set to 7Fh
CONTROL	See 5.1.y.3
GROUP NUMBER	The SNTL shall ignore this field.
ADDITIONAL CDB LENGTH	The SNTL may ignore this field. Otherwise, if this field is not set to 18h, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.
SERVICE ACTION	Set to 0009h
RDPROTECT	See 9.2.2
DPO	The SNTL shall ignore this bit.
FUA	The SNTL shall set the FUA bit in the NVMe Read command to the value in this field.
RARC	The SNTL shall ignore this bit.
DLD2	The SNTL shall ignore this bit.
DLD1	The SNTL shall ignore this bit.
DLD0	The SNTL shall ignore this bit.
LOGICAL BLOCK ADDRESS	The SNTL shall set the SLBA field in the NVMe Read command to the value in this field.
EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG	The SNTL shall set the ELBTL field in the NVMe Read command to the value in this field.
EXPECTED LOGICAL BLOCK APPLICATION TAG	The SNTL shall set the ELBAT field in the NVMe Read command to the value in this field.
LOGICAL BLOCK APPLICATION TAG MASK	The SNTL shall set the ELBATM field in the NVMe Read command to the value in this field.
TRANSFER LENGTH	If this field is set to zero, then the SNTL shall complete the READ (32) command with GOOD status and shall not issue an NVMe Read command. If this field is set to non-zero, then the SNTL shall set the NLB field in the NVMe Read command to the value in this field less one.

### 9.3 READ CAPACITY commands

This subclause applies to the translation of SCSI READ CAPACITY commands.

#### 9.3.1 READ CAPACITY (10) command

The READ CAPACITY (10) command requests that the SNTL transfer eight bytes of parameter data describing the capacity and medium format of the direct access block device to the application client.

Table 10 shows the translation for the fields of the READ CAPACITY (10) CDB.

*Table 10 READ CAPACITY (10) CDB field translations*

Field	Description
OPERATION CODE	Set to 25h
CONTROL	See 5.1.y.3

##### 9.3.1.1 READ CAPACITY (10) parameter data

The SNTL shall return READ CAPACITY (10) parameter data as described in SBC-6.

Table 11 shows the translation for the fields in the READ CAPACITY (10) parameter data.

*Table 11 READ CAPACITY (10) parameter data translations*

Field	Description
RETURNED LOGICAL BLOCK ADDRESS	Shall be set to the lower of: <ul style="list-style-type: none"> <li>a) NVMe Identify Namespace data structure (CNS 00h) NSZE field (see NVMCS) less one; or</li> <li>b) FFFF_FFFFh.</li> </ul>
LOGICAL BLOCK LENGTH IN BYTES	Shall be set to the Logical Block Length as described in 5.1.x.

#### 9.3.2 READ CAPACITY (16) command

The READ CAPACITY (16) command requests that the SNTL transfer parameter data describing the capacity and medium format of the direct access block device to the application client.

Table 12 shows the translation for the fields of the READ CAPACITY (16) CDB.

*Table 12 READ CAPACITY (16) CDB field translations*

Field	Description
OPERATION CODE	Set to 9Eh
SERVICE ACTION	Set to 10h
ALLOCATION LENGTH	See 3.1.x
CONTROL	See 5.1.y.3

##### 9.3.2.1 READ CAPACITY (16) parameter data

The SNTL shall return READ CAPACITY (16) parameter data as described in SBC-6.

Table 13 shows the translation for the fields in the READ CAPACITY (16) parameter data.

Table 13 READ CAPACITY (16) parameter data translations

Field	Description										
RETURNED LOGICAL BLOCK ADDRESS	Shall be set to the value of the NVMe Identify Namespace data structure (CNS 00h) NSZE field (see NVMCS) less one.										
LOGICAL BLOCK LENGTH IN BYTES	Shall be set to the Logical Block Length as described in 5.1.x.										
RC_BASIS	Shall be set to 00b.										
P_TYPE	<p>Shall be set to 000b if:</p> <ul style="list-style-type: none"> <li>a) the SNTL does not support protection information;</li> <li>b) the application client does not support the value indicated by the NVMe Identify Namespace data structure (CNS 00h) MTELBA bit,</li> </ul> <p>otherwise this bit shall be set according to the following table based on the NVMe Identify Namespace data structure DPS field:</p> <table border="1"> <thead> <tr> <th>DPS field</th><th>P_TYPE</th></tr> </thead> <tbody> <tr> <td>000b</td><td>000b</td></tr> <tr> <td>001b</td><td>000b</td></tr> <tr> <td>010b</td><td>001b</td></tr> <tr> <td>011b</td><td>010b</td></tr> </tbody> </table>	DPS field	P_TYPE	000b	000b	001b	000b	010b	001b	011b	010b
DPS field	P_TYPE										
000b	000b										
001b	000b										
010b	001b										
011b	010b										
PROT_EN	<p>Shall be set to 0b if:</p> <ul style="list-style-type: none"> <li>a) the SNTL does not support protection information;</li> <li>b) the application client does not support the value indicated by the NVMe Identify Namespace data structure (CNS 00h) MTELBA bit; or</li> <li>c) the NVMe Identify Namespace data structure (CNS 00h) DPS field is set to 000b (i.e., PI disabled),</li> </ul> <p>otherwise this bit shall be set to 1b.</p>										
P_I_EXPONENT	Shall be set to 0h.										
LOGICAL BLOCKS PER PHYSICAL BLOCK EXPONENT	Shall be set to 0h.										
LBPME	<p>Shall be set to 1b if:</p> <ul style="list-style-type: none"> <li>a) DSM Supported (see NCD.2) is true;</li> </ul> <p>or</p> <ul style="list-style-type: none"> <li>a) WZ Supported (see NCD.3) is true; and</li> <li>b) NVMe Identify Namespace data structure (CNS 00h) WZDS bit is set to 1b (i.e., Write Zeroes Deallocate bit supported),</li> </ul> <p>otherwise this bit shall be set to 0b.</p>										
LBPRZ	Shall be set to 1b if the NVMe Identify Namespace data structure (CNS 00h) DRB field is set to 001b (i.e., deallocated logical block returns zeros), otherwise this bit shall be set to 0b.										
LOWEST ALIGNED LOGICAL BLOCK ADDRESS	Shall be set to 0h.										

## 9.4 SANITIZE command

The SANITIZE command specifies that one of several sanitize operations be performed.

The SNTL shall translate a SANITIZE command to an NVMe Sanitize command and an NVMe Get Log Page command and set the Command Specific fields (i.e., Dwords 10-15) in each NVMe command not specified by the translation to zero, including the Protection Information fields.

Table 14 shows the translation for the fields of the SANITIZE CDB.

*Table 14 SANITIZE CDB field translations*

Field	Description
OPERATION CODE	Set to 48h
IMMED	If set to 1b, then the SNTL shall validate all CDB and parameter list fields, if any, and return GOOD status before issuing any commands to the NVMe Controller. If set to 0b, then the SNTL shall wait for completion of all commands issued to the NVMe Controller before returning status.
ZNR	The SNTL shall ignore this field.
AUSE	See 9.4.1.2, 9.4.2, and 9.4.3
SERVICE ACTION	If set to 01h (i.e., OVERWRITE), then see 9.4.1.2. If set to 02h (i.e., BLOCK ERASE), then see 9.4.2. If set to 03h (i.e., CRYPTOGRAPHIC ERASE), then see 9.4.3. If set to 1Fh (i.e., EXIT FAILURE MODE), then see 9.4.4. If set to any other value, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.
PARAMETER LIST LENGTH	See 5.1.y.4 If: <ul style="list-style-type: none"> <li>a) the SERVICE ACTION field is set to 01h and the PARAMETER LIST LENGTH field is not set to 0008h;</li> <li>b) the SERVICE ACTION field is set to 02h and the PARAMETER LIST LENGTH field is not set to 0000h;</li> <li>c) the SERVICE ACTION field is set to 03h and the PARAMETER LIST LENGTH field is not set to 0000h; or</li> <li>d) the SERVICE ACTION field is set to 1Fh and the PARAMETER LIST LENGTH field is not set to 0000h,</li> </ul> then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.
CONTROL	See 5.1.y.3

If an error occurs during the processing of the SANITIZE command and the IMMED bit is set to 1b, then the SNTL shall establish a deferred error condition (see SPC-7) with the sense key set to MEDIUM ERROR, and the additional sense code set to SANITIZE COMMAND FAILED.

## 9.4.1 Sanitize using overwrite method

### 9.4.1.1 OVERWRITE service action parameter list translation

For the SANITIZE command with the overwrite method, the parameter list is translated as shown in Table 15.

*Table 15 OVERWRITE service action parameter list translation*

Field	Description
INVERT	See 9.4.1.2
TEST	If this field is not set to 00b, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.
OVERWRITE COUNT	If this field is set to 00h or a value greater than 10h, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST. See 9.4.1.2
INITIALIZATION PATTERN LENGTH	If this field is not set to 0004h, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST. See 9.4.1.2
INITIALIZATION PATTERN	See 9.4.1.2

### 9.4.1.2 Sanitize using overwrite method translation details

If the SNTL supports the overwrite method and the NVMe Identify Controller data structure (CNS 01h) OWS bit (see NVMBASE) is set to 1b (i.e., sanitize overwrite supported), then:

- 1) if no CHECK CONDITION status was returned as a result of processing the fields in Table 15 and the IMMED bit is set to 1b, then the SNTL shall return GOOD status;
- 2) if no CHECK CONDITION status was returned as a result of processing the fields in Table 15, then the SNTL shall issue an NVMe Sanitize command with:
  - a. the SANACT field set to 011b (i.e., Start an Overwrite sanitize operation);
  - b. the AUSE bit set to the value of the AUSE bit;
  - c. the OWPASS field set to the value of the OVERWRITE COUNT field;
  - d. the OIPBP bit set to the value of the INVERT bit; and
  - e. the OVRPAT field set to the value of the INITIALIZATION PATTERN field;
- 3) if the NVMe Sanitize command in step 2) completes with an error and the IMMED bit is set to 0b, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to MEDIUM ERROR, and the additional sense code set to SANITIZE COMMAND FAILED;
- 4) if the NVMe Sanitize command in step 2) completes without error, then the SNTL shall periodically issue an NVMe Get Log Page command for the Sanitize Status Log Page (Log Page Identifier 81h) until the SOS field does not indicate 010b (i.e., Sanitizing); and
- 5) if the IMMED bit is set to 0b (i.e., the SNTL has not yet returned status for the SANITIZE command), the NVMe Sanitize command in step 2) completed without error, and:

- a. the most recent SOS field in the NVMe Sanitize Status Log Page in step 4) indicates 001b (i.e., Sanitized), then the SNTL shall complete the SANITIZE command with GOOD status; or
- b. the most recent SOS field in the NVMe Sanitize Status Log Page in step 4) indicates 011b (i.e., Sanitize Failed), then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to MEDIUM ERROR, and the additional sense code set to SANITIZE COMMAND FAILED.

If the SNTL does not support the overwrite method or the NVMe Identify Controller data structure (CNS 01h) OWS bit (see NVMBASE) is set to 0b (i.e., sanitize overwrite not supported), then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

### 9.4.2 Sanitize using block erase method

If the SNTL supports the block erase method and the NVMe Identify Controller data structure (CNS 01h) BES bit (see NVMBASE) is set to 1b (i.e., sanitize block erase supported), then:

- 1) if the IMMED bit is set to 1b, then the SNTL shall return GOOD status;
- 2) the SNTL shall issue an NVMe Sanitize command with:
  - a. the SANACT field set to 010b (i.e., Start a Block Erase sanitize operation);
  - b. the AUSE bit set to the value of the AUSE bit; and
  - c. the No Deallocate After Sanitize bit set to 0b;
- 3) if the NVMe Sanitize command in step 2) completes with an error and the IMMED bit is set to 0b, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to MEDIUM ERROR, and the additional sense code set to SANITIZE COMMAND FAILED;
- 4) if the NVMe Sanitize command in step 2) completes without error, then the SNTL shall periodically issue an NVMe Get Log Page command for the Sanitize Status Log Page (Log Page Identifier 81h) until the SOS field does not indicate 010b (i.e., Sanitizing); and
- 5) if the IMMED bit is set to 0b (i.e., the SNTL has not yet returned status for the SANITIZE command), the NVMe Sanitize command in step 2) completed without error, and:
  - a. the most recent SOS field in the NVMe Sanitize Status Log Page in step 4) indicates 001b (i.e., Sanitized), then the SNTL shall complete the SANITIZE command with GOOD status; or
  - b. the most recent SOS field in the NVMe Sanitize Status Log Page in step 4) indicates 011b (i.e., Sanitize Failed), then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to MEDIUM ERROR, and the additional sense code set to SANITIZE COMMAND FAILED.

If the SNTL does not support the block erase method or the NVMe Identify Controller data structure (CNS 01h) BES bit (see NVMBASE) is set to 0b (i.e., sanitize block erase not supported), then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

### 9.4.3 Sanitize using cryptographic erase method

If the SNTL supports the cryptographic erase method and the NVMe Identify Controller data structure (CNS 01h) CES bit (see NVMBASE) is set to 1b (i.e., sanitize crypto erase supported), then:

- 1) if the IMMED bit is set to 1b, then the SNTL shall return GOOD status;
- 2) the SNTL shall issue an NVMe Sanitize command with:
  - a. the SANACT field set to 100b (i.e., Start a Crypto Erase sanitize operation);
  - b. the AUSE bit set to the value of the AUSE bit; and
  - c. the No Deallocate After Sanitize bit set to 0b;
- 3) if the NVMe Sanitize command in step 2) completes with an error and the IMMED bit is set to 0b, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to MEDIUM ERROR, and the additional sense code set to SANITIZE COMMAND FAILED;
- 4) if the NVMe Sanitize command in step 2) completes without error, then the SNTL shall periodically issue an NVMe Get Log Page command for the Sanitize Status Log Page (Log Page Identifier 81h) until the SOS field does not indicate 010b (i.e., Sanitizing); and
- 5) if the IMMED bit is set to 0b (i.e., the SNTL has not yet returned status for the SANITIZE command), the NVMe Sanitize command in step 2) completed without error, and:
  - a. the most recent SOS field in the NVMe Sanitize Status Log Page in step 4) indicates 001b (i.e., Sanitized), then the SNTL shall complete the SANITIZE command with GOOD status; or
  - b. the most recent SOS field in the NVMe Sanitize Status Log Page in step 4) indicates 011b (i.e., Sanitize Failed), then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to MEDIUM ERROR, and the additional sense code set to SANITIZE COMMAND FAILED.

If the SNTL does not support the cryptographic erase method or the NVMe Identify Controller data structure (CNS 01h) CES bit (see NVMBASE) (i.e., sanitize crypto erase not supported) is set to 0b (i.e., crypto erase sanitize not supported), then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

#### **9.4.4 Exit the sanitize failure mode**

To process the exit failure mode request, the SNTL shall:

- 1) if the NVMe Identify Controller data structure (CNS 01h) SANICAP field (see NVMBASE) is set to 00b (i.e., Sanitize command not supported), then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB; and
- 2) if the NVMe Identify Controller data structure (CNS 01h) SANICAP field (see NVMBASE) is not set to 00b (i.e., Sanitize command supported):
  - 1) if the IMMED bit is set to 1b, then the SNTL shall return GOOD status;
  - 2) the SNTL shall issue an NVMe Sanitize command with the SANACT field set to 001b (i.e., Exit Failure Mode);
  - 3) if the completion of the NVMe Sanitize command in step 2) indicates that a sanitize operation is active, then the SNTL shall periodically issue an NVMe Get Log Page command for the Sanitize Status Log Page (Log Page Identifier 81h) until the SOS field does not indicate 010b (i.e., Sanitizing);

- 4) if the IMMED bit is set to 0b (i.e., the SNTL has not yet returned status for the SANITIZE command), and:
  - a. the most recent SOS field in the NVMe Sanitize Status Log Page in step 3) indicates 001b (i.e., Sanitized), then the SNTL shall return GOOD status; or
  - b. the most recent SOS field in the NVMe Sanitize Status Log Page in step 3) indicates 011b (i.e., Sanitize Failed), then the SANITIZE command shall be terminated with CHECK CONDITION status with the sense key set to MEDIUM ERROR, and the additional sense code set to SANITIZE COMMAND FAILED.

## 9.5 START STOP UNIT command

The START STOP UNIT command provides a method for controlling the power condition of an NVMe Controller.

The SNTL shall translate a START STOP UNIT command to an NVMe Flush command and an NVMe Set Features command, and the SNTL shall set the Command Specific fields (i.e., Dwords 10-15) in each NVMe command not specified by the translation to zero.

Table 16 shows the translation for fields of the START STOP UNIT CDB.

*Table 16 START STOP UNIT CDB field translations*

Field	Description
OPERATION CODE	Set to 1Bh
IMMED	Shall be ignored.
POWER CONDITION MODIFIER	See 9.5.1
POWER CONDITION	See 9.5.1
NO_FLUSH	See 9.5.1
LOEJ	See 9.5.1
START	See 9.5.1
CONTROL	See 5.1.y.3

### 9.5.1 START STOP UNIT command processing

If the POWER CONDITION field is set to 0h (i.e., START\_VALID) and the LOEJ bit is set to 1b, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

If the NO\_FLUSH bit is set to 0b and the NVMe Identify Controller data structure (CNS 01h) VWCP bit (see NVMBASE) is set to 1b (i.e., volatile write cache present), then the SNTL shall issue an NVMe Flush command.

The POWER CONDITION field and the POWER CONDITION MODIFIER fields are processed as described in Table 17. The NVMe Controller may not support the ideal Requested NVMe Power State (e.g., the NVMe Controller may only support Power State 0), so the SNTL shall normalize the Requested NVMe Power State to one that is supported by the NVMe Controller and other vendor specific power limitations (i.e., UBM reported maximum slot power). The SNTL shall issue an NVMe Set Features command with:

- a) the SV bit set to 0b (i.e., No save);

- b) the WH field set to 000b;
- c) the UIDX field set to 00h;
- d) the PS field set to the Normalized Requested NVMe Power State (see *Table 17*); and
- e) the FID field set to 02h (i.e., Power Management).

*Table 17 POWER CONDITION and POWER CONDITION MODIFIER field translation*

POWER CONDITION value	POWER CONDITION name	POWER CONDITION MODIFIER value	START bit	Normalized Requested NVMe Power State <sup>a</sup>
0h	START_VALID	0h	0b	N – 1
		0h	1b	0
1h	ACTIVE	0h	Any	0
2h	IDLE	0h		MIN(1, N-1)
		1h		MIN(2,N-1)
		2h		MIN(3, N-1)
3h	STANDBY	0h		MAX(N – 3, 0)
		1h		MAX(N – 2, 0)
7h	LU_CONTROL	0h	Any	The SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.
Ah	FORCE_IDLE_0	0h		
		1h		
		2h		
Bh	FORCE_STANDBY_0	0h		
		1h		
		2h		
All other combinations				The SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.
<sup>a</sup> N = NVMe Identify Controller data structure (CNS 01h) NPSS field (see NVMeBASE), (i.e., Number of Power States Support) plus one				

If an error occurs during the processing of the START STOP UNIT command, then the SNTL shall terminate the command with the sense key set to HARDWARE ERROR, and the additional sense code set to INTERNAL TARGET FAILURE.

## 9.6 SYNCHRONIZE CACHE commands

This subclause applies to the translation of SCSI SYNCHRONIZE CACHE commands.

### 9.6.1 SYNCHRONIZE CACHE (10) command

The SYNCHRONIZE CACHE (10) command is used to flush the most recent data in the cache of the NVMe Controller to the physical medium.

*Table 18* shows the translation for fields of the SYNCHRONIZE CACHE (10) CDB.

*Table 18 SYNCHRONIZE CACHE (10) CDB field translations*

Field	Description
OPERATION CODE	Set to 35h
IMMED	If set to 1b, then the SNTL shall validate all CDB fields and return GOOD status before issuing the NVMe Flush command. If set to 0b, then the SNTL shall wait for completion of all commands issued to the NVMe Controller before returning status.
LOGICAL BLOCK ADDRESS	The SNTL shall ignore this field and process this command as though this field contained zero (see SBC-6).
GROUP NUMBER	The SNTL shall ignore this field.
NUMBER OF LOGICAL BLOCKS	The SNTL shall ignore this field and process this command as though this field contained zero (see SBC-6).
CONTROL	See 5.1.y.3

The SNTL shall issue an NVMe Flush command.

If an error occurs during the processing of the SYNCHRONIZE CACHE (10) command, then the SNTL shall terminate the command with the sense key set to MEDIUM ERROR, and the additional sense code set to INTERNAL TARGET FAILURE.

### 9.6.2 SYNCHRONIZE CACHE (16) command

The SYNCHRONIZE CACHE (16) command is used to flush the most recent data in the cache of the NVMe Controller to the physical medium.

Table 19 shows the translation for fields of the SYNCHRONIZE CACHE (16) CDB.

*Table 19 SYNCHRONIZE CACHE (16) CDB field translations*

Field	Description
OPERATION CODE	Set to 91h
IMMED	If set to 1b, then the SNTL shall validate all CDB fields and return GOOD status before issuing the NVMe Flush command. If set to 0b, then the SNTL shall wait for completion of all commands issued to the NVMe Controller before returning status.
LOGICAL BLOCK ADDRESS	The SNTL shall ignore this field and process this command as though this field contained zero (see SBC-6).
GROUP NUMBER	The SNTL shall ignore this field.
NUMBER OF LOGICAL BLOCKS	The SNTL shall ignore this field and process this command as though this field contained zero (see SBC-6).
CONTROL	See 5.1.y.3

The SNTL shall issue an NVMe Flush command.

If an error occurs during the processing of the SYNCHRONIZE CACHE (16) command, then the SNTL shall terminate the command with the sense key set to MEDIUM ERROR, and the additional sense code set to INTERNAL TARGET FAILURE.

## 9.7 UNMAP command

The UNMAP command requests that the SNTL transfer parameter data from the application client that is used by the SNTL to build a set of LBA range entries that are used in an NVMe Dataset Management command to request the specified LBAs to be deallocated.

If DSM Supported (see NCD.2) is false, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID COMMAND OPERATION CODE.

Table 20 shows the translation for fields of the UNMAP CDB.

*Table 20 UNMAP CDB field translations*

Field	Description
OPERATION CODE	Set to 42h
ANCHOR	The SNTL shall ignore this field.
GROUP NUMBER	The SNTL shall ignore this field.
PARAMETER LIST LENGTH	<p>See 5.1.y.4</p> <p>If this field is set to zero, then the SNTL shall complete the command with GOOD status.</p> <p>If this field is set to one through seven, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.</p> <p>If this field is greater than seven then:</p> <ol style="list-style-type: none"> <li>1) if there are zero complete UNMAP block descriptors (see 9.7.1), then the SNTL shall complete the command with GOOD status; and</li> <li>2) if there are one or more complete UNMAP block descriptors, then the SNTL shall: <ol style="list-style-type: none"> <li>1) create NVMe Dataset Management Range Definition entries that describe the logical blocks represented in all the complete UNMAP block descriptors in the UNMAP parameter list as described in 9.7.2; and</li> <li>2) issue one or more NVMe Dataset Management commands with: <ol style="list-style-type: none"> <li>a. the NR field set to the number of NVMe Dataset Management Range Definition entries less one;</li> <li>b. the AD bit set to 1b (i.e., Attribute Deallocate);</li> <li>c. the IDW bit set to 0b; and</li> <li>d. and the IDR bit set to 0b.</li> </ol> </li> </ol> </li> </ol>
CONTROL	See 5.1.y.3

### 9.7.1 Complete UNMAP block descriptors

The UNMAP command and parameter list specify three different length values. These values should all describe the same number of UNMAP block descriptors; however, when the application did not accurately set the values, the SNTL must determine how many valid UNMAP block descriptors are present. A truncated UNMAP block descriptor (i.e., less than 16 bytes long) is ignored. The SNTL shall

calculate the number of complete UNMAP block descriptors, which is the minimum of the following using integer math (round down):

- $(\text{PARAMETER LIST LENGTH} - 8) / 16$
- $(\text{UNMAP DATA LENGTH} - 6) / 16$
- $\text{UNMAP BLOCK DESCRIPTOR DATA LENGTH} / 16$

### 9.7.2 Creating NVMe Dataset Management Range Definition entries

The UNMAP command parameter list contains UNMAP block descriptors, each defining an LBA range to be unmapped.

The SNTL shall create an NVMe Dataset Management Range Definition entry for each UNMAP block descriptor as follows:

- a) the CATTR field set to zero;
- b) the SLBA field set to the UNMAP LOGICAL BLOCK ADDRESS field; and
- c) the LLB field set to the NUMBER OF LOGICAL BLOCKS field.

## 9.8 VERIFY commands

This subclause applies to the translation of SCSI VERIFY commands.

The SNTL shall translate a VERIFY command to an NVMe Compare or one or more NVMe Verify commands and set the Command Specific fields (i.e., Dwords 10-15) in the NVMe command not specified by the translation to zero, including the Protection Information fields.

The NVMe command used depends on the requested operation as described in 9.8.3.

The SNTL shall set the FUA bit in each NVMe command to 1b.

### 9.8.1 VERIFY (10) command

The VERIFY (10) command is used to verify data on the NVMe controller's medium.

Table 21 shows the translation for the fields of the VERIFY (10) CDB.

*Table 21 VERIFY (10) CDB field translations*

Field	Description
OPERATION CODE	Set to 2Fh
VRPROTECT	See 9.8.4
DPO	The SNTL shall ignore this field.
BYCHK	See 9.8.3
LOGICAL BLOCK ADDRESS	<p>The SNTL shall set the SLBA field in the NVMe command to the value in this field.</p> <p>The SNTL shall set the ELBTL field in the NVMe command to bits 31:0 of the value in this field.</p>
GROUP NUMBER	The SNTL shall ignore this field.

Field	Description
VERIFICATION LENGTH	<p>The SNTL shall:</p> <ol style="list-style-type: none"> <li>1) if this field is set to zero, then the SNTL shall complete the VERIFY (10) command with GOOD status and shall not issue any NVMe commands;</li> <li>2) if this field is greater than FFFFh, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB; or</li> <li>3) set the NLB field in the NVMe command to: <ol style="list-style-type: none"> <li>a. if NVMe Compare command, the value in this field less one; or</li> <li>b. if NVMe Verify command, as described in 9.8.2.</li> </ol> </li> </ol>
CONTROL	See 5.1.y.3

## 9.8.2 Multiple NVMe Verify

The SNTL shall verify the range of logical blocks specified by the LOGICAL BLOCK ADDRESS field and the VERIFICATION LENGTH field, using one or more NVMe Verify commands.

An NVMe device can restrict the maximum size of the NVMe Verify command. If:

- a) the NVMe Identify Controller data structure (CNS 01h) ONCS field (see NVMeBASE) NVMVFYS bit is set to 0b; and
- b) the NVMe I/O Command Set Specific Identify Controller data structure (CNS 06h, CSI 00h) VSL field is non-zero;

then the SNTL shall set each NVMe Verify command NLB field to less than or equal to:

- a)  $(2^{(VSL+12)}) / LBA\_SIZE$ , if IPI Formatted (see NCD.4) is FALSE; or
- b)  $(2^{(VSL+12)}) / (LBA\_SIZE + 8)$ , if IPI Formatted (see NCD.4) is TRUE.

Where:

VSL: NVMe VSL field

LBA\_SIZE: Logical Block Length (see 5.1.x).

## 9.8.3 BYTCHK

The BYTCHK field specifies the contents of the data buffer, if any, used by the VERIFY command.

Table 22 shows the translation of the BYTCHK field.

*Table 22 BYTCHK CDB field translation*

BYTCHK field	Description
00b	<p>The SNTL shall use the NVMe Verify command.</p> <p>If Verify Supported (see NCD.5) is false, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.</p>

BYTCHK field	Description
01b	<p>The SNTL shall use the NVMe Compare command.</p> <p>If:</p> <ul style="list-style-type: none"> <li>a) the NVMe Identify Controller data structure (CNS 01h) NVMCMPS bit (see NVMeBASE), (i.e., Compare command support) is set to 0b; or</li> <li>b) the VRPROTECT field is set to zero and the NVMe namespace is formatted for end-to-end protection information;</li> </ul> <p>then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.</p>
10b	<p>The SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.</p>
11b	

#### 9.8.4 VRPROTECT

If the VRPROTECT field is non-zero, then:

- a) if the SNTL does not support protection information or the NVMe namespace is not formatted for end-to-end protection information, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB;
- b) if the SNTL supports protection information, the NVMe namespace is formatted for end-to-end protection information, and the BYTCHK field is set to 01b, then the SNTL shall set the PRACT bit to 1b and the PRCHK field to 000b.; or
- c) if the SNTL supports protection information, the NVMe namespace is formatted for end-to-end protection information, and the BYTCHK field is set to 00b, then the SNTL shall translate the VRPROTECT field as defined in *Table 23*.

*Table 23 VRPROTECT field translations with BYTCHK = 00b*

Protection Type	VRPROTECT	PRACT	PRCHK		
			Bit 2 (Guard)	Bit 1 (App Tag)	Bit 0 (Ref Tag)
1	000b	0b	0b	0b	0b
	001b	0b	1b	0b	1b
	010b	0b	0b	0b	1b
	011b	0b	0b	0b	0b
	100b	0b	1b	0b	0b
	101b	0b	1b	0b	1b
2	000b	0b	0b	0b	0b
	001b	0b	1b	1b	1b
	010b	0b	0b	1b	1b
	011b	0b	0b	0b	0b
	100b	0b	1b	0b	0b
	101b	0b	1b	1b	1b

Protection Type	VRPROTECT	PRACT	PRCHK		
			Bit 2 (Guard)	Bit 1 (App Tag)	Bit 0 (Ref Tag)
3	000b	0b	0b	0b	0b
	001b	0b	1b	0b	0b
	010b	0b	0b	0b	0b
	011b	0b	0b	0b	0b
	100b	0b	1b	0b	0b
	101b	0b	1b	0b	0b
Any	110b	The SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB			
	111b				

### 9.8.5 VERIFY (12) command

The VERIFY (12) command is used to verify data on the NVMe controller's medium.

Table 21 shows the translation for the fields of the VERIFY (12) CDB.

*Table 24 VERIFY (12) CDB field translations*

Field	Description
OPERATION CODE	Set to AFh
VRPROTECT	See 9.8.4
DPO	The SNTL shall ignore this field.
BYTCHK	See 9.8.3
LOGICAL BLOCK ADDRESS	The SNTL shall set the SLBA field in the NVMe command to the value in this field. The SNTL shall set the ELBTL field in the NVMe command to bits 31:0 of the value in this field.
GROUP NUMBER	The SNTL shall ignore this field.
VERIFICATION LENGTH	The SNTL shall: 4) if this field is set to zero, then the SNTL shall complete the VERIFY (12) command with GOOD status and shall not issue any NVMe commands; 5) if this field is greater than FFFFh, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB; or 6) set the NLB field in the NVMe command to: a. if NVMe Compare command, the value in this field less one; or b. if NVMe Verify command, as described in 9.8.2.
CONTROL	See 5.1.y.3

### 9.8.6 VERIFY (16) command

The VERIFY (16) command is used to verify data on the NVMe controller's medium.

Table 21 shows the translation for the fields of the VERIFY (16) CDB.

*Table 25 VERIFY (16) CDB field translations*

Field	Description
OPERATION CODE	Set to 8Fh
VRPROTECT	See 9.8.4
DPO	The SNTL shall ignore this field.
BYCHK	See 9.8.3
LOGICAL BLOCK ADDRESS	The SNTL shall set the SLBA field in the NVMe command to the value in this field. The SNTL shall set the ELBTL field in the NVMe command to bits 31:0 of the value in this field.
GROUP NUMBER	The SNTL shall ignore this field.
VERIFICATION LENGTH	The SNTL shall: <ul style="list-style-type: none"> <li>7) if this field is set to zero, then the SNTL shall complete the VERIFY (16) command with GOOD status and shall not issue any NVMe commands;</li> <li>8) if this field is greater than FFFFh, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB; or</li> <li>9) set the NLB field in the NVMe command to: <ul style="list-style-type: none"> <li>a. if NVMe Compare command, the value in this field less one; or</li> <li>b. if NVMe Verify command, as described in 9.8.2.</li> </ul> </li> </ul>
CONTROL	See 5.1.y.3

### 9.8.7 VERIFY (32) command

The VERIFY (32) command requests that the device perform the actions defined for the VERIFY (10) command with extensions for type 2 protection.

If the SNTL does not support protection information or the NVMe namespace is not formatted for type 2 end-to-end protection information, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID COMMAND OPERATION CODE.

Table 26 shows the translation for the fields of the VERIFY (32) CDB.

*Table 26 VERIFY (32) CDB field translations*

Field	Description
OPERATION CODE	Set to 7Fh
CONTROL	See 5.1.y.3
GROUP NUMBER	The SNTL shall ignore this field.
ADDITIONAL CDB LENGTH	If this field is not set to 18h, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.
SERVICE ACTION	Set to 000Ah
VRPROTECT	See 9.8.4

Field	Description
DPO	The SNTL shall ignore this field.
BYCHK	See 9.8.3
LOGICAL BLOCK ADDRESS	The SNTL shall set the SLBA field in the NVMe command to the value in this field.
EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG	The SNTL shall set the ELBTL field in the NVMe command based on the value in this field.
EXPECTED LOGICAL BLOCK APPLICATION TAG	The SNTL shall set the ELBAT field in the NVMe command to the value in this field.
LOGICAL BLOCK APPLICATION TAG MASK	The SNTL shall set the LBATM field in the NVMe command to the value in this field.
VERIFICATION LENGTH	<p>The SNTL shall:</p> <ol style="list-style-type: none"> <li>1) if this field is set to zero, then the SNTL shall complete the VERIFY (32) command with GOOD status and shall not issue any NVMe commands;</li> <li>2) if this field is greater than FFFFh, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB; or</li> <li>3) set the NLB field in the NVMe command to: <ol style="list-style-type: none"> <li>a. if NVMe Compare command, the value in this field less one; or</li> <li>b. if NVMe Verify command, as described in 9.8.2.</li> </ol> </li> </ol>

## 9.9 WRITE commands

This subclause applies to the translation of SCSI WRITE commands.

The SNTL shall translate WRITE commands to an NVMe Write command and set the Command Specific fields (i.e., Dwords 10-15) in the NVMe Write command not specified by the translation to zero, including the Protection Information fields.

### 9.9.1 WRITE (10) command

Table 27 shows the translation for the fields of the WRITE (10) CDB.

*Table 27 WRITE (10) CDB field translations*

Field	Description
OPERATION CODE	Set to 2Ah
WRPROTECT	See 9.9.2
DPO	The SNTL shall ignore this field.
FUA	The SNTL shall set the FUA bit in the NVMe Write command to the value in this field.

Field	Description
LOGICAL BLOCK ADDRESS	The SNTL shall set the SLBA field in the NVMe Write command to the value in this field. The SNTL shall set the LBTL field in the NVMe Write command to bits 31:0 of the value in this field.
GROUP NUMBER	The SNTL shall ignore this field.
TRANSFER LENGTH	If this field is set to zero, then the SNTL shall complete the WRITE (10) command with GOOD status and shall not issue an NVMe Write command. If this field is set to non-zero, then the SNTL shall set the NLB field in the NVMe Write command based on the value in this field less one.
CONTROL	See 5.1.y.3

### 9.9.2 WRPROTECT

If the WRPROTECT field is non-zero, then:

- if the SNTL does not support protection information or the NVMe namespace is not formatted for end-to-end protection information, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB; or
- if the SNTL supports protection information and the NVMe namespace is formatted for end-to-end protection information, then the SNTL shall translate the WRPROTECT field as defined in *Table 28*.

*Table 28 WRPROTECT field translations*

Protection Type	WRPROTECT	PRACT	PRCHK		
			Bit 2 (Guard)	Bit 1 (App Tag)	Bit 0 (Ref Tag)
1	000b	1b (i.e., insert)	0b	0b	0b
	001b	0b (i.e., passed)	1b	0b	1b
	010b	0b (i.e., passed)	0b	0b	1b
	011b	0b (i.e., passed)	0b	0b	0b
	100b	0b (i.e., passed)	1b	0b	0b
	101b	0b (i.e., passed)	1b	0b	1b
2	000b	1b (i.e., insert)	0b	0b	0b
	001b	0b (i.e., passed)	1b	1b	1b
	010b	0b (i.e., passed)	0b	1b	1b
	011b	0b (i.e., passed)	0b	0b	0b
	100b	0b (i.e., passed)	1b	0b	0b
	101b	0b (i.e., passed)	1b	1b	1b
3	000b	1b (i.e., insert)	0b	0b	0b
	001b	0b (i.e., passed)	1b	0b	0b
	010b	0b (i.e., passed)	0b	0b	0b
	011b	0b (i.e., passed)	0b	0b	0b

Protection Type	WRPROTECT	PRACT	PRCHK		
			Bit 2 (Guard)	Bit 1 (App Tag)	Bit 0 (Ref Tag)
	100b	0b (i.e., passed)	1b	0b	0b
	101b	0b (i.e., passed)	1b	0b	0b
Any	110b	The SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB			
	111b				

### 9.9.3 WRITE (12) command

Table 27 shows the translation for the fields of the WRITE (12) CDB.

*Table 29 WRITE (12) CDB field translations*

Field	Description
OPERATION CODE	Set to AAh
WRPROTECT	See 9.9.2
DPO	The SNTL shall ignore this field.
FUA	The SNTL shall set the FUA bit in the NVMe Write command to the value in this field.
LOGICAL BLOCK ADDRESS	The SNTL shall set the SLBA field in the NVMe Write command to the value in this field. The SNTL shall set the LBTL field in the NVMe Write command to bits 31:0 of the value in this field.
GROUP NUMBER	The SNTL shall ignore this field.
TRANSFER LENGTH	If this field is set to zero, then the SNTL shall complete the WRITE (12) command with GOOD status and shall not issue an NVMe Write command. If this field is set to non-zero, then the SNTL shall set the NLB field in the NVMe Write command based on the value in this field less one.
CONTROL	See 5.1.y.3

### 9.9.4 WRITE (16) command

Table 27 shows the translation for the fields of the WRITE (16) CDB.

*Table 30 WRITE (16) CDB field translations*

Field	Description
OPERATION CODE	Set to 8Ah
WRPROTECT	See 9.9.2
DPO	The SNTL shall ignore this field.
FUA	The SNTL shall set the FUA bit in the NVMe Write command to the value in this field.
DLD2	The SNTL shall ignore this bit.

Field	Description
LOGICAL BLOCK ADDRESS	The SNTL shall set the SLBA field in the NVMe Write command to the value in this field. The SNTL shall set the LBTL field in the NVMe Write command to bits 31:0 of the value in this field.
DLD1	The SNTL shall ignore this bit.
DLD0	The SNTL shall ignore this bit.
GROUP NUMBER	The SNTL shall ignore this field.
TRANSFER LENGTH	If this field is set to zero, then the SNTL shall complete the WRITE (16) command with GOOD status and shall not issue an NVMe Write command. If this field is set to non-zero, then the SNTL shall set the NLB field in the NVMe Write command based on the value in this field less one.
CONTROL	See 5.1.y.3

### 9.9.5 WRITE (32) command

The WRITE (32) command requests that the device perform the actions defined for the WRITE (10) command with extensions for type 2 protection.

If the SNTL does not support protection information or the NVMe namespace is not formatted for type 2 end-to-end protection information, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID COMMAND OPERATION CODE.

Table 31 shows the translation for the fields of the WRITE (32) CDB.

*Table 31 WRITE (32) CDB field translations*

Field	Description
OPERATION CODE	Set to 7Fh
CONTROL	See 5.1.y.3
GROUP NUMBER	The SNTL shall ignore this field.
ADDITIONAL CDB LENGTH	If this field is not set to 18h, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.
SERVICE ACTION	Set to 000Bh
WRPROTECT	See 9.9.2
DPO	The SNTL shall ignore this field.
FUA	The SNTL shall set the FUA bit in the NVMe Write command to the value in this field.
DLD2	The SNTL shall ignore this bit.
DLD1	The SNTL shall ignore this bit.
DLD0	The SNTL shall ignore this bit.
LOGICAL BLOCK ADDRESS	The SNTL shall set the SLBA field in the NVMe Write command to the value in this field.

Field	Description
EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG	The SNTL shall set the LBTL field in the NVMe Write command based on the value in this field.
EXPECTED LOGICAL BLOCK APPLICATION TAG	The SNTL shall set the LBAT field in the NVMe Write command to the value in this field.
LOGICAL BLOCK APPLICATION TAG MASK	The SNTL shall set the LBATM field in the NVMe Write command to the value in this field.
TRANSFER LENGTH	<p>If this field is set to zero, then the SNTL shall complete the WRITE (32) command with GOOD status and shall not issue an NVMe Write command.</p> <p>If this field is set to non-zero, then the SNTL shall set the NLB field in the NVMe Write command based on the value in this field less one.</p>

## 9.10 WRITE LONG commands

This subclause applies to the translation of SCSI WRITE LONG commands.

The WRITE LONG command requests that the device mark a logical block as containing an error.

If WU Supported (see NCD.6) is false, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID COMMAND OPERATION CODE.

The SNTL shall translate a WRITE LONG command to an NVMe Write Uncorrectable command and set the Command Specific fields (i.e., Dwords 10-15) in the NVMe Write Uncorrectable command not specified by the translation to zero, including the Protection Information fields.

The SNTL shall set the NLB field in the NVMe Write Uncorrectable command to zero.

### 9.10.1 WRITE LONG (10) command

Table 32 shows the translation for the fields of the WRITE LONG (10) CDB.

*Table 32 WRITE LONG (10) CDB field translations*

Field	Description
OPERATION CODE	Set to 3Fh
WR_UNCOR	If this bit is set to 0b, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.
LOGICAL BLOCK ADDRESS	The SNTL shall set the SLBA field in the NVMe Write Uncorrectable command to the value specified in this field.
CONTROL	See 5.1.y.3

### 9.10.2 WRITE LONG (16) command

Table 32 shows the translation for the fields of the WRITE LONG (16) CDB.

*Table 33 WRITE LONG (16) CDB field translations*

Field	Description
OPERATION CODE	Set to 9Fh

Field	Description
SERVICE ACTION	Set to 11h
WR_UNCOR	If this bit is set to 0b, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.
LOGICAL BLOCK ADDRESS	The SNTL shall set the SLBA field in the NVMe Write Uncorrectable command to the value specified in this field.
CONTROL	See 5.1.y.3

## 9.11 WRITE SAME commands

This subclause applies to the translation of SCSI WRITE SAME commands.

The SNTL shall translate a WRITE SAME command to NVMe Write commands or NVMe Write Zeros commands and set the Command Specific fields (i.e., Dwords 10-15) in the NVMe command not specified by the translation to zero, including the Protection Information fields.

The NVMe command used depends on the requested operation as described in 9.11.3.

### 9.11.1 WRITE SAME (10) command

Table 34 shows the translation for the fields of the WRITE SAME (10) CDB.

*Table 34 WRITE SAME (10) CDB field translations*

Field	Description
OPERATION CODE	Set to 41h
WRPROTECT	See 9.11.2
ANCHOR	See 9.11.3
UNMAP	See 9.11.3
LOGICAL BLOCK ADDRESS	The SNTL shall set the NVMe SLBA field and the LBTL field in the initial NVMe command to the value of the LOGICAL BLOCK ADDRESS field.
NUMBER OF LOGICAL BLOCKS	<p>If this field is set to zero or greater than the MAXIMUM WRITE SAME LENGTH field in the Block Limits VPD page (See 10.5.1.2), then the SNTL should terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.</p> <p>If the NUMBER OF LOGICAL BLOCKS field is set to a value other than zero, then the SNTL shall repeatedly write the data block to the number of logical blocks specified by this field starting at the LBA specified by the LOGICAL BLOCK ADDRESS field.</p>
GROUP NUMBER	The SNTL shall ignore this field.
CONTROL	See 5.1.y.3

### 9.11.2 WRPROTECT

If the WRPROTECT field is non-zero, then:

- a) if the SNTL does not support protection information or the NVMe namespace is not formatted for end-to-end protection information, then the SNTL shall terminate the command with CHECK

CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB; or

- b) if the SNTL supports protection information and the NVMe namespace is formatted for end-to-end protection information, then the SNTL shall translate the WRPROTECT field as defined in *Table 28*.

### 9.11.3 ANCHOR bit, UNMAP bit, and NDOB bit

*Table 35* shows the interactions of the UNMAP bit, the ANCHOR bit, and the NDOB bit. The NDOB bit for the WRITE SAME (10) command is assumed to have a value of 0b.

*Table 35 UNMAP bit, ANCHOR bit, and NDOB bit interactions*

UNMAP bit	ANCHOR bit	NDOB bit	Description
0b	0b	0b	See 9.11.4 (Write the logical block)
		1b	See 9.11.5 (Write zeros)
	1b	0b	The SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.
		1b	
1b	0b or 1b	0b	If the LBPRZ bit in the READ CAPACITY (16) parameter data is set to 1b and the data block transferred from the application client to the SNTL for the WRITE SAME command has all bits set to 0b, then the SNTL shall write zeros with deallocate as described in 9.11.6 (Write zeros with deallocate). Otherwise, the SNTL shall write the data block as described in 9.11.4 (Writing the data block).
	0b or 1b	1b	See 9.11.6 (Write zeros with deallocate)

### 9.11.4 Write the logical blocks

The SNTL shall write the data block to the range of logical blocks specified by the LOGICAL BLOCK ADDRESS field and the NUMBER OF LOGICAL BLOCKS field, repeatedly, to the NVMe Controller using the NVMe Write command, with:

- a) the LR bit set to 0b; and
- b) the FUA bit set to 0b.

If the Logical Block Length (see 5.1.x) is 512 bytes, the SNTL shall set the NLB field in each of the NVMe Write commands to zero (i.e., single logical block writes).

The NVMe device restricts the size of the NVMe Write command. If the Logical Block Length (see 5.1.x) is not 512 bytes, the SNTL shall set the NLB field in each of the NVMe Write commands to less than:

- a)  $(2^{(MDTS+12)}) / LBA\_SIZE$ , if IPI Formatted (see NCD.4) is false;
- b)  $(2^{(MDTS+12)}) / (LBA\_SIZE + 8)$ , if IPI Formatted (see NCD.4) is true.

Where:

MDTS: NVMe Identify Controller data structure MDTS field

LBA\_SIZE: Logical Block Length (see 5.1.x)

### 9.11.5 Write zeros

If WZ Supported (see NCD.3) is true, the SNTL shall write zeros to the range of logical blocks specified by the LOGICAL BLOCK ADDRESS field and the NUMBER OF LOGICAL BLOCKS field, repeatedly, to the NVMe Controller using the NVMe Write Zeroes command, with:

- a) the LR bit set to 0b;
- b) the FUA bit set to 0b;
- c) the DEAC bit set to 0b.

If WZ Supported (see NCD.3) is false, the SNTL shall:

- 1) allocate and zero fill a data block; and
- 2) write the data block to the range of logical blocks as described in 9.11.4.

### 9.11.6 Write zeros with deallocate

To write zeros with deallocate:

If:

- a) the LBRPZ field is set to xx1b in the Logical Block Provisioning VPD page (see NCD.4); and
  - b) WZ Supported (see NCD.3) is true;
- then the SNTL shall request the NVMe Controller deallocate the range of logical blocks specified by the LOGICAL BLOCK ADDRESS field and the NUMBER OF LOGICAL BLOCKS field, repeatedly, on the medium of the NVMe Controller using the NVMe Write Zeroes command, with:
- a) the LR bit set to 0b;
  - b) the FUA bit set to 0b;
  - c) the DEAC bit set to 1b;

otherwise the SNTL shall write zeros as described in 9.11.5.

### 9.11.7 WRITE SAME (16) command

Table 34 shows the translation for the fields of the WRITE SAME (16) CDB.

*Table 36 WRITE SAME (16) CDB field translations*

Field	Description
OPERATION CODE	Set to 93h
WRPROTECT	See 9.11.2
ANCHOR	See 9.11.3
UNMAP	See 9.11.3
NDOB	See 9.11.3
LOGICAL BLOCK ADDRESS	The SNTL shall set the NVMe SLBA field and the LBTL field in the initial NVMe command to the value of this field.

Field	Description
NUMBER OF LOGICAL BLOCKS	<p>If this field is set to zero or greater than the MAXIMUM WRITE SAME LENGTH field in the Block Limits VPD page (See 10.5.1.2), then the SNTL should terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.</p> <p>If the NUMBER OF LOGICAL BLOCKS field is set to a value other than zero, then the SNTL shall repeatedly write the data block to the number of logical blocks specified by this field starting at the LBA specified by the LOGICAL BLOCK ADDRESS field.</p>
GROUP NUMBER	The SNTL shall ignore this field.
CONTROL	See 5.1.y.3

EdNote: Add headings for cross references

## **10.4.X Mode parameter block descriptor**