Date: April 21, 2004

To: T10 Committee (SCSI)

From: Jim Coomes (Seagate)

Subject:SBC 32 Byte Commands for End-to-End Data Protection

Revision history

rev 7-

a) Changes to SBC-2 and SPC-3 are incorporated into affected clauses.
b) Text was added to command descriptions in SBC-2 that are disabled when application ownership of the reference tag is enabled (non-LBA locked data protection.
c) References to Protection Information VPD page changed to Extended INQUIRY Data VPD page.

rev 6 -

a) The function of non-LBA locked reference tag is labeled application client reference tag ownership.
b) A bit is added to standard inquiry data to report support of application client ownership of the reference tag.
c) A bit is added to Format to enable application client ownership of the reference tag.
d) A bit is added to Long read capacity data to indicate when application client ownership reference tag is enabled.
e) The new 32 byte commands with reference tag seed are failed if the medium is not formatted with protection information and application client ownership of the reference tag enabled.
f) When the application client owns the reference tag, WRITE (6), VERIFY (10), (12), (16), WRITE AND VERIFY (10), (12), (16), WRITE SAME (10), (16) commands are failed.
g) When the application client owns the reference tag, READ commands with RDPROTECT equal 000b are allowed. READ commands with RDPROTECT 001b, 010b or 011d are failed.
h) When the application client owns the reference tag, Write commands with WRPROTECT equal 000b are allowed. WRITE commands with WRPROTECT 001b, 010b or 011d are failed.
i) Protection information field names DATA BLOCK XXX TAG were changed to LOGICAL BLOCK XXX TAG per SBC-2.

rev 5 - Updated per input at Nov 03 CAP meeting and implementation in SBC-2 r11.

a) Changed the modification of footnotes “c” in SBC-2 r11 to take the “may” out of 32 byte commands providing the application information knowledge.
b) Changed the xxPROTECT fields in all 32 byte commands to 3 bit fields.
c) The LOGICAL BLOCK APPLICATION VALUE field is renamed EXPECTED LOGICAL BLOCK APPLICATION TAG.
d) The descriptions of the new fields in the 32 byte commands are revised.

rev 4 - Added Write Same 32 byte command. Rewrite of the command field descriptions to direct the checking requirements back to the 10 byte version of the commands.

rev 3- Added 32 byte Verify and Write And Verify command.

rev 2 - This revision changed text in proposal 03-176r6 for the RDPROTECT field, Table Footnote c and the WRPROTECT field, Table Footnote c. Editorial convention was corrected for “a bit set to one”

rev 1 -This revision changes the RELADR field in the proposed commands to reserved, RDPROTECT and WRPROTECT field descriptions to reference 03-176r5 and qualifies LOGICAL BLOCK APPLICATION TAG checking with the RDPROTECT and WRPROTECT fields.

Overview

There is a need to provide the initial value of the LOGICAL BLOCK REFERENCE TAG defined in SBC-2 on a command by command basis. One use case is in a configuration where a controller (e.g., a RAID) remaps the LBA to a different LBA space on a physical LUN. By passing the initial value of the LOGICAL BLOCK REFERENCE TAG in the command to the LUN, the original data protection block may be passed through the controller to the LUN and checked. This function provides end to end protection in the remapping case.

To provide the space for the initial LOGICAL BLOCK REFERENCE TAG and maintain 8 byte LBA space, 32 byte formats are proposed for read and write operations.

This proposal additionally provides a mechanism to enable device server checking to the LOGICAL BLOCK APPLICATION TAG in the protection information.

Changes to document SBC-2 r13

4.15.2 Protection information format

~~Figure 3~~ Table 6 defines the placement of protection information in a logical block.

Table 6 — User data and protection information format

Byte\Bit	7	6	5	4	3	2	1	0
0	USER DATA							
n - 1								
n	(MSB)	LOGICAL BLOCK GUARD						
n + 1								(LSB)
n + 2	(MSB)	LOGICAL BLOCK APPLICATION TAG						
n + 3								(LSB)
n + 4	(MSB)	LOGICAL BLOCK REFERENCE TAG						
n + 7								(LSB)

The USER DATA field shall contain user data. The contents of the USER DATA field shall be used to generate and check the CRC contained in the LOGICAL BLOCK GUARD field.

The LOGICAL BLOCK GUARD field contains the CRC (see 4.15.3) of the contents of the USER DATA field.

The LOGICAL BLOCK APPLICATION TAG field is set by the application client. The contents of the logical block application tag are not defined by this standard. The LOGICAL BLOCK APPLICATION TAG field may be modified by a device server if the APP_TAG_OWN bit is set to zero in the Control mode page (see SPC-3). The contents of the LOGICAL BLOCK APPLICATION TAG field shall not be used to generate or check the CRC contained in the LOGICAL BLOCK GUARD field.

The LOGICAL BLOCK REFERENCE TAG field is ~~set to the least significant four bytes of the LBA to which~~ an incrementing value associated with the logical block ~~is associated~~. For commands that do not include an INITIAL LOGICAL BLOCK REFERENCE TAG field, the~~The~~ first logical block in application client data buffer shall contain the least significant four bytes of the LBA contained in the LOGICAL BLOCK ADDRESS field of the command associated with the logical block. For commands that include an INITIAL LOGICAL BLOCK REFERENCE TAG field, the first logical block transferred shall contain the LOGICAL BLOCK REFERENCE TAG equal to the value in the command. Each logical block in the application client data buffer contains a LOGICAL BLOCK REFERENCE TAG field with the logical block reference tag of the previous logical block plus one. The contents of the LOGICAL BLOCK REFERENCE TAG field shall not be used to generate or check the CRC contained in the LOGICAL BLOCK GUARD field.

5.3 FORMAT UNIT command

5.3.1 FORMAT UNIT command overview

The FORMAT UNIT command (see table 12) formats the medium into application client addressable logical blocks per the application client defined options. In addition, the medium may be certified and control

structures may be created for the management of the medium and defects. The degree that the medium is altered by this command is vendor-specific.

Table 12 — FORMAT UNIT command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (04h)							
1	FMTPINFO	RTO_REQ	LONGLIST	FMTDATA	CMPLIST	DEFECT LIST FORMAT		
2	Vendor specific							
3	(MSB)			INTERLEAVE				
4								(LSB)
5	CONTROL							

The simplest mandatory form of the FORMAT UNIT command (i.e., a FORMAT UNIT command with no parameter data) accomplishes medium formatting with little application client control over defect management. The device server implementation determines the degree of defect management that is to be performed. Two additional mandatory forms of this command increase the application client's control over defect management. Several optional forms of this command further increase the application client's control over defect management, by allowing the application client to specify:

a) defect list(s) to be used;
b) defect locations;
c) that logical unit certification be enabled; and
d) exception handling in the event that defect lists are not accessible.

During the format operation, the device server shall respond to commands as follows:

a) In response to all commands except REQUEST SENSE and INQUIRY, the device server shall return CHECK CONDITION status unless a reservation conflict exists, in which case RESERVATION CONFLICT status shall be returned;
b) In response to the INQUIRY command, the device server shall respond as commanded; and
c) In response to the REQUEST SENSE command, unless an error has occurred, the device server shall return a sense key of NOT READY with the additional sense code set to LOGICAL UNIT NOT READY FORMAT IN PROGRESS, with the sense key specific bytes set for progress indication (see SPC-3). See SPC-3 for a description of deferred error handling that may occur during the format operation.

NOTE 1 - The MODE SELECT parameters, if any, should be set prior to issuing the FORMAT UNIT command.

Editor's Note 1: The above list should clarify whether tasks already in the task set are all aborted or not.

During the processing of the FORMAT UNIT command, the device server may perform a medium defect management algorithm. The algorithm may be controlled by the application client, using optional forms of this command. Four sources of defect location information (i.e., defects) are defined as follows:

a) Primary defect list (PLIST). This is the list of defects, that may be supplied by the original manufacturer of the device or medium, that are considered permanent defects. The PLIST is located outside of the application client-accessible logical block space. The PLIST is accessible by the device server (to reference while formatting), but it is not accessible by the application client except through the READ DEFECT DATA command. Once created, the original PLIST shall not be subject to change;

b) Logical unit certification list (CLIST). This list includes defects detected by the device server during an optional certification process performed during the FORMAT UNIT command. This list shall be added to the GLIST;
c) Data defect list (DLIST). This list of defect descriptors may be supplied by the application client to the device server during the the FORMAT UNIT command. This list shall be added to the GLIST. If the DEFECT LIST LENGTH field in the defect list header is set to zero, there is no DLIST; and
d) Grown defect list (GLIST). The GLIST includes all defects sent by the application client or detected by the device server. The GLIST does not include the PLIST. If the CMPLST bit is set to zero, the GLIST shall include DLISTs provided to the device server during the previous and the current FORMAT UNIT commands. The GLIST shall also include:
   A) defects detected by the format operation during medium certification;
   B) defects previously identified with a REASSIGN BLOCKS command (see 5.18); and
   C) defects previously detected by the device server and automatically reallocated.

A format protection information (FMTPINFO) bit set to zero specifies that the device server shall format the medium to the block length specified in the mode parameter block descriptor of the mode parameter header (see SPC-3). A FMTPINFO bit set to one specifies that the device server shall format the medium to the block length specified in the mode parameter block descriptor of the mode parameter header plus eight (e.g., if the block length is 512, then the formatted block length is 520). Following a ~~A~~ successful format, the PROT_EN bit in Long read capacity data (see 5.13) indicates ~~that changes~~ whether protection information (see 4.15) is ~~included shall cause the PROT_EN bit in the READ CAPACITY (16) data to be changed.~~enabled.

If the FMTPINFO bit is set to zero, the reference tag own request (RTO_REQ) bit is ignored. If the FMTPINFO bit is set to one and the RTO_REQ bit is set to one, the device server shall enable application client ownership of the LOGICAL BLOCK REFERENCE TAG field in protection information (see 4.15). If the FMTPINFO bit set to one and the RTO_REQ bit is set to zero, the device server shall disable application client ownership of the LOGICAL BLOCK REFERENCE TAG field. Following a successful format, the RTO_EN bit in Long read capacity data (see 5.13) indicates whether application client ownership of the LOGICAL BLOCK REFERENCE TAG field is enabled.

When protection information is written during a FORMAT UNIT command (i.e., the FMTPINFO bit is set to one) protection information shall be written to a default value of FFFFFFFF FFFFFFFFh.

A LONGLIST bit set to zero specifies that the defect list header follows the short format in table 15. A LONGLIST bit set to one specifies that the defect list header follows the long format in table 16.

A format data (FMTDATA) bit set to zero specifies that no parameter data be transferred from the application client data-out buffer. The source of defect information is not specified.

5.8 READ (6) command

The READ (6) command (see table 29) requests that the device server transfer data, including user data but not including protection information, to the application client. The most recent data value written, or to be written if cached, in the addressed logical block shall be returned.

Table 29 — READ (6) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (08h)							
1	Reserved			(MSB)				
2	LOGICAL BLOCK ADDRESS							
3								(LSB)
4	TRANSFER LENGTH							
5	CONTROL							

The cache control bits (see 5.9) are not provided for this command. Block devices with cache memory may have values for the cache control bits that affect the READ (6) command; however, no default values are defined by this standard. If explicit control is required, the READ (10) command should be used.

The LOGICAL BLOCK ADDRESS field specifies the logical block where the read operation shall begin.

The TRANSFER LENGTH field specifies the number of contiguous logical blocks of data to be transferred. A TRANSFER LENGTH of zero indirectly specifies that 256 logical blocks shall be transferred. Any other value directly specifies the number of logical blocks that shall be transferred. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.4.2).

NOTE 2 - Although the READ (6) command is limited to directly addressing logical blocks up to a capacity of 2 Gigabytes, for block lengths of 512 bytes, this command has been maintained as mandatory since some system initialization routines require that the READ (6) command be used. Application clients should migrate from the READ (6) command to the READ (10) command which may address 2 Terabytes with block lengths of 512 bytes, or the READ (16) command to address more than 2 Terabytes.

NOTE 3 - For the READ (10) command, READ (12) command, and READ (16) command, a transfer length of zero specifies that no logical blocks are transferred.

The device server shall check the protection information read from the medium as described in table 30.

Table 30 — Protection information checking for READ (6)

Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information [e]	Extented INQUIRY Data VPD page bit value [d]	If check fails [b] [c], additional sense code
Yes	No	LOGICAL BLOCK GUARD	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED
			GRD_CHK = 0	No check performed
		LOGICAL BLOCK APPLICATION TAG	APP_CHK = 1 [a]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
			APP_CHK = 0	No check performed
		LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
			REF_CHK = 0	No check performed
No		No protection information available to check		

[a] The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. The method for acquiring this knowledge is not defined by this standard.

[b] If an error is reported the sense key shall be set to ABORTED COMMAND.

[c] If multiple errors occur, the selection of which error to report is not defined by this standard.

[d] See the Extented INQUIRY Data VPD page (see SPC-3) for a description of the GRD_CHK, APP_CHK, and REF_CHK bits.

[e] If the device server detects a LOGICAL BLOCK APPLICATION TAG field set to FFFFh, it shall not check any protection information in the associated logical block.

[f] If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server checks the logical block reference tag by comparing it to the lower 4 bytes of the LBA associated with the data block. If the RTO_EN bit is set to one, the device server checks the logical block reference tag only if it has knowledge of the contents of the LOGICAL BLOCK REFERENCE TAG field. The method for acquiring this knowledge is not defined by this standard.

5.9 READ (10) command

The READ (10) command (see table 31) requests that the device server transfer data to the application client. Data includes user data and protection information, if any. The most recent data value written in the addressed logical block shall be returned.

Table 31 — READ (10) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (28h)							
1	RDPROTECT			DPO	FUA	Reserved		Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	Reserved							
7	(MSB)	TRANSFER LENGTH						
8								(LSB)
9	CONTROL							

See the LOCK UNLOCK CACHE (10) command (see 5.4) for a definition of the LOGICAL BLOCK ADDRESS field.

The device server shall check the protection information read from the medium based on the RDPROTECT field as described in table 32.

Table 32 — RDPROTECT field (part 1 of 4)

Value	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information [h]	Extented INQUIRY Data VPD page bit value [g]	If check fails [d] [f], additional sense code
000b	Yes	No	LOGICAL BLOCK GUARD	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED
				GRD_CHK = 0	No check performed
			LOGICAL BLOCK APPLICATION TAG	APP_CHK = 1 [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
				APP_CHK = 0	No check performed
			LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
				REF_CHK = 0	No check performed
	No		No protection information available to check		

[a] A read operation to a logical unit that supports protection information (see 4.15) and has not been formatted with protection information shall fail with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

[b] If the logical unit does not support protection information the requested command should fail with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

[c] The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. ~~The method for acquiring this knowledge is~~ Knowledge of the LOGICAL BLOCK APPLICATION TAG field contents may be obtained by use of the READ (32) command (see 5.XX) or by a method not defined by this standard.

[d] If an error is reported the sense key shall be set to ABORTED COMMAND.

[e] Transmit protection information to the application client.

[f] If multiple errors occur, the selection of which error to report is not defined by this standard.

[g] See the Extented INQUIRY Data VPD page (see SPC-3) for a description of the GRD_CHK, APP_CHK, and REF_CHK bits.

[h] If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field set to FFFFh, the checking of all protection information in the associated logical block shall be disabled.

[i] If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, READ (10), READ (12),and READ (16) commands with the RDPROTECT field set to 000b may be processed by the device server. If the RTO_EN bit is set to one, the device server shall fail READ (10), READ (12), and READ (16) commands if the RDPROTECT field is set to 001b, 010b, or 011b with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

Table 32 — RDPROTECT **field** (part 2 of 4)

Value	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information [h]	Extented INQUIRY Data VPD page bit value [g]	If check fails [d] [f], additional sense code
[b] [i] 001b	Yes	Yes [e]	LOGICAL BLOCK GUARD	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED
				GRD_CHK = 0	No check performed
			LOGICAL BLOCK APPLICATION TAG	APP_CHK = 1 [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
				APP_CHK = 0	No check performed
			LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
				REF_CHK = 0	No check performed
	No [a]	No protection information available to transmit to the application client or for checking			

[a] A read operation to a logical unit that supports protection information (see 4.15) and has not been formatted with protection information shall fail with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

[b] If the logical unit does not support protection information the requested command should fail with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

[c] The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. ~~The method for acquiring this knowledge is~~ Knowledge of the LOGICAL BLOCK APPLICATION TAG field contents may be obtained by use of the READ (32) command (see 5.XX) or by a method not defined by this standard.

[d] If an error is reported the sense key shall be set to ABORTED COMMAND.

[e] Transmit protection information to the application client.

[f] If multiple errors occur, the selection of which error to report is not defined by this standard.

[g] See the Extented INQUIRY Data VPD page (see SPC-3) for a description of the GRD_CHK, APP_CHK, and REF_CHK bits.

[h] If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field set to FFFFh, the checking of all protection information in the associated logical block shall be disabled.

[i] If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, READ (10), READ (12),and READ (16) commands with the RDPROTECT field set to 000b may be processed by the device server. If the RTO_EN bit is set to one, the device server shall fail READ (10), READ (12), and READ (16) commands if the RDPROTECT field is set to 001b, 010b, or 011b with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

Table 32 — RDPROTECT **field** (part 3 of 4)

Value	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information [h]	Extented INQUIRY Data VPD page bit value [g]	If check fails [d] [f], additional sense code
[b] [i] 010b	Yes	Yes [e]	LOGICAL BLOCK GUARD	Shall not	No check performed
			LOGICAL BLOCK APPLICATION TAG	APP_CHK = 1 [c] [h]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
				APP_CHK = 0	No check performed
			LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 [h]	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
				REF_CHK = 0	No check performed
	No [a]	No protection information available to transmit to the application client or for checking			

[a] A read operation to a logical unit that supports protection information (see 4.15) and has not been formatted with protection information shall fail with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

[b] If the logical unit does not support protection information the requested command should fail with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

[c] The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. ~~The method for acquiring this knowledge is~~ Knowledge of the LOGICAL BLOCK APPLICATION TAG field contents may be obtained by use of the READ (32) command (see 5.XX) or by a method not defined by this standard.

[d] If an error is reported the sense key shall be set to ABORTED COMMAND.

[e] Transmit protection information to the application client.

[f] If multiple errors occur, the selection of which error to report is not defined by this standard.

[g] See the Extented INQUIRY Data VPD page (see SPC-3) for a description of the GRD_CHK, APP_CHK, and REF_CHK bits.

[h] If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field set to FFFFh, the checking of all protection information in the associated logical block shall be disabled.

[i] If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, READ (10), READ (12),and READ (16) commands with the RDPROTECT field set to 000b may be processed by the device server. If the RTO_EN bit is set to one, the device server shall fail READ (10), READ (12), and READ (16) commands if the RDPROTECT field is set to 001b, 010b, or 011b with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

Table 32 — RDPROTECT **field** (part 4 of 4)

Value	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information [h]	Extented INQUIRY Data VPD page bit value [g]	If check fails [d] [f], additional sense code
[b] [i] 011b	Yes	Yes [e]	LOGICAL BLOCK GUARD	Shall not	No check performed
			LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
			LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed
	No [a]	No protection information available to transmit to the application client or for checking			
100b - 111b	Reserved				

[a] A read operation to a logical unit that supports protection information (see 4.15) and has not been formatted with protection information shall fail with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

[b] If the logical unit does not support protection information the requested command should fail with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

[c] The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. ~~The method for acquiring this knowledge is~~ Knowledge of the LOGICAL BLOCK APPLICATION TAG field contents may be obtained by use of the READ (32) command (see 5.XX) or by a method not defined by this standard.

[d] If an error is reported the sense key shall be set to ABORTED COMMAND.

[e] Transmit protection information to the application client.

[f] If multiple errors occur, the selection of which error to report is not defined by this standard.

[g] See the Extented INQUIRY Data VPD page (see SPC-3) for a description of the GRD_CHK, APP_CHK, and REF_CHK bits.

[h] If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field set to FFFFh, the checking of all protection information in the associated logical block shall be disabled.

[i] If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, READ (10), READ (12),and READ (16) commands with the RDPROTECT field set to 000b may be processed by the device server. If the RTO_EN bit is set to one, the device server shall fail READ (10), READ (12), and READ (16) commands if the RDPROTECT field is set to 001b, 010b, or 011b with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

A disable page out (DPO) bit set to zero specifies that the retention priority shall be determined by the RETENTION PRIORITY fields in the Caching mode page (see 6.3.2). A DPO bit set to one specifies that the device server shall assign the logical blocks accessed by this command the lowest retention priority for being fetched into or retained by the cache. A DPO bit set to one overrides any retention priority specified in the Caching mode page. All other aspects of the algorithm implementing the cache memory replacement strategy are not defined by this standard.

NOTE 4 - The DPO bit is used to control replacement of logical blocks in the cache memory when the application client has information on the future usage of the logical blocks. If the DPO bit is set to one, the application client is specifying that the logical blocks accessed by the command are not likely to be accessed

again in the near future and should not be put in the cache memory nor retained by the cache memory. If the DPO bit is set to zero, the application client is specifying that the logical blocks accessed by this command are likely to be accessed again in the near future.

A force unit access (FUA) bit set to zero specifies that the device server may provide the data from cache memory. For read operations, any logical blocks that are contained in the cache memory may be transferred to the application client data-in buffer directly from the cache memory. For write operations, logical blocks may be transferred directly from the application client data-out buffer to the cache memory. GOOD status may be returned to the application client prior to writing the logical blocks to the medium. Any error that occurs after the GOOD status is returned is a deferred error, and information regarding the error is not reported until a subsequent command.

An FUA bit set to one specifies that the device server shall provide the data from the medium, not cache memory. Read commands shall access the specified logical blocks from the medium (i.e., the data is not directly retrieved from the cache). If the cache contains a more recent version of a logical block than the medium, the logical block shall first be written to the medium. Write commands shall not return GOOD status until the logical blocks have actually been written on the medium (i.e., the data is not write cached). Read commands that cause data to be written to the medium from cache and that encounter an error shall cause a deferred error (see SPC-3) to be reported.

The TRANSFER LENGTH field specifies the number of contiguous logical blocks of data that shall be transferred. A TRANSFER LENGTH of zero specifies that no logical blocks shall be transferred. This condition shall not be considered an error. Any other value specifies the number of logical blocks that shall be transferred. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.4.2).

NOTE 5 - For the READ (6) command, a TRANSFER LENGTH of zero specifies that 256 logical blocks are transferred.

5.13 READ CAPACITY (16) command

The READ CAPACITY (16) command (see table 37) provides a means for the application client to request information regarding the capacity of the block device. This command is implemented as a service action of the SERVICE ACTION IN opcode. This command may be processed as if it has a HEAD OF QUEUE task attribute (see 4.7)

Table 37 — READ CAPACITY (16) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (9Eh)							
1	Reserved			SERVICE ACTION (10h)				
2	(MSB)	LOGICAL BLOCK ADDRESS						
9								(LSB)
10	(MSB)	ALLOCATION LENGTH						
13								(LSB)
14	Reserved							PMI
15	CONTROL							

See the LOCK UNLOCK CACHE (10) command (see 5.4) for a definition of the LOGICAL BLOCK ADDRESS field. See the READ CAPACITY (10) command (see 5.12) for a description of the other fields in this command.

The long read capacity data is defined in table 38.

Table 38 — Long read capacity data

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)	RETURNED LOGICAL BLOCK ADDRESS						
7								(LSB)
8	(MSB)	BLOCK LENGTH IN BYTES						
11								(LSB)
12	Reserved						RTO_EN	PROT_EN
13	Reserved							
31								

The RETURNED LOGICAL BLOCK ADDRESS field and BLOCK LENGTH IN BYTES field of the long read capacity data are the same as the in the short read capacity data described in the READ CAPACITY (10) command (see 5.12). The maximum value that shall be returned in the RETURNED LOGICAL BLOCK ADDRESS field is FFFFFFFF FFFFFFFEh.

A PROT_EN bit set to one indicates that the medium was formatted with protection information (see 4.15) enabled. A PROT_EN bit set to zero indicates that the medium was not formatted with protection information enabled.

A reference tag own enable (RTO_EN) bit set to one indicates that application client ownership of the LOGICAL BLOCK REFERENCE TAG field in protection information is enabled (i.e., the medium was formatted with protection information (see 4.15) enabled and the RTO_REQ bit set to one for the format). A RTO_EN bit set to zero indicates that application client ownership of the LOGICAL BLOCK REFERENCE TAG field in protection information is disabled.

5.22 VERIFY (10) command

The VERIFY (10) command (see table 53) requests that the device server verify the data on the medium. Data includes user data and protection information, if any.

Table 53 — VERIFY (10) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (2Fh)							
1	VRPROTECT			DPO	Reserved		BYTCHK	Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	Restricted for MMC-4	Reserved						
7	(MSB)	VERIFICATION LENGTH						
8								(LSB)
9	CONTROL							

See 5.9 for a description of the DPO bit. See the LOCK UNLOCK CACHE (10) command (see 5.4) for a definition of the LOGICAL BLOCK ADDRESS field.

If the MODE SELECT command is implemented, and the Verify Error Recovery mode page (see 6.3.5) is also implemented, then the current settings in that page specifies the verification criteria. If the Verify Error Recovery mode page is not implemented, then the verification criteria is vendor-specific.

If the byte check (BYTCHK) bit is set to zero, the device server shall:

a) perform a medium verification with no data comparison and not transfer any data from the application client data-out buffer; and
b) check protection information read from the medium based on the VRPROTECT field as described in table 54.

If the BYTCHK bit is set to one, the device server shall:

a) perform a byte-by-byte comparison of user data read from the medium and user data transferred from the application client data-out buffer;
b) check protection information read from the medium based on the VRPROTECT field as described in table 55;
c) check protection transferred from the application client data-out buffer based on the VRPROTECT field as described in table 56; and
d) perform a byte-by-byte comparison of protection information read from the medium and transferred from the application client data-out buffer based on the VRPROTECT field as described in table 57.

The order of the user data and protection information checks and comparisons is vendor-specific.

Editor's Note 2: The above one-line paragraph is new. The a)b)c) lists above that are all reformatted from jumbled paragraphs.

If a byte-by-byte comparison is unsuccessful for any reason, the device server shall return CHECK CONDITION status and the sense key shall be set to MISCOMPARE with the appropriate additional sense code for the condition.

The VERIFICATION LENGTH field specifies the number of contiguous logical blocks of data or blanks that shall be verified. A VERIFICATION LENGTH of zero specifies that no logical blocks shall be verified. This condition shall not be considered as an error. Any other value specifies the number of logical blocks that shall be verified. The VERIFICATION LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.4.2).

If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, the device server shall fail a VERIFY(10) with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

If the BYTCHK bit is set to zero, the device server shall check the protection information read from the medium based on the VRPROTECT field as described in table 54.

Table 54 — VRPROTECT **field with** BYTCHK **set to zero - checking protection information read from the medium** (part 1 of 2)

Value	Logical unit formatted with protection information	Field in protection information [g]	Extented INQUIRY Data VPD page bit value [f]	If check fails [d] [e], additional sense code
000b	Yes	LOGICAL BLOCK GUARD	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED
			GRD_CHK = 0	No check performed
		LOGICAL BLOCK APPLICATION TAG	APP_CHK = 1 [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
			APP_CHK = 0	No check performed
		LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
			REF_CHK = 0	No check performed
	No	No protection information on the medium to check. Only user data is checked.		
001b [b]	Yes	LOGICAL BLOCK GUARD	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED
			GRD_CHK = 0	No check performed
		LOGICAL BLOCK APPLICATION TAG	APP_CHK = 1 [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
			APP_CHK = 0	No check performed
		LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
			REF_CHK = 0	No check performed
	No	Error condition [a]		

[a] A verify operation to a logical unit that supports protection information (see 4.15) and has not been formatted with protection information shall fail with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

[b] If the logical unit does not support protection information the requested command should fail with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

[c] The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. ~~The method for acquiring this knowledge is~~ Knowledge of the LOGICAL BLOCK APPLICATION TAG field contents may be obtained by use of the VERIFY (32) command (see 5.XX) or by a method not defined by this standard.

[d] If an error is reported, the sense key shall be set to ABORTED COMMAND.

[e] If multiple errors occur, the selection of which error to report is not defined by this standard.

[f] See the Extented INQUIRY Data VPD page (see SPC-3) for a description of the GRD_CHK, APP_CHK, and REF_CHK bits.

[g] If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field set to FFFFh, the checking of all protection information shall be disabled for the associated logical block..

Table 54 — VRPROTECT **field with** BYTCHK **set to zero - checking protection information read from the medium** (part 2 of 2)

Value	Logical unit formatted with protection information	Field in protection information [g]	Extented INQUIRY Data VPD page bit value [f]	If check fails [d] [e], additional sense code
010b [b]	Yes	LOGICAL BLOCK GUARD	Shall not	No check performed
		LOGICAL BLOCK APPLICATION TAG	APP_CHK = 1 [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
			APP_CHK = 0	No check performed
		LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
			REF_CHK = 0	No check performed
	No	Error condition [a]		
011b [b]	Yes	LOGICAL BLOCK GUARD	Shall not	No check performed
		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed
	No	Error condition [a]		
100b - 111b	Reserved			

[a] A verify operation to a logical unit that supports protection information (see 4.15) and has not been formatted with protection information shall fail with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

[b] If the logical unit does not support protection information the requested command should fail with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

[c] The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. ~~The method for acquiring this knowledge is~~ Knowledge of the LOGICAL BLOCK APPLICATION TAG field contents may be obtained by use of the VERIFY (32) command (see 5.XX) or by a method not defined by this standard.

[d] If an error is reported, the sense key shall be set to ABORTED COMMAND.

[e] If multiple errors occur, the selection of which error to report is not defined by this standard.

[f] See the Extented INQUIRY Data VPD page (see SPC-3) for a description of the GRD_CHK, APP_CHK, and REF_CHK bits.

[g] If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field set to FFFFh, the checking of all protection information shall be disabled for the associated logical block..

If the BYTCHK bit is set to one, the device server shall check the protection information read from the medium based on the VRPROTECT field as described in table 55.

Table 55 — VRPROTECT field with BYTCHK set to one - checking protection information read from the medium (part 1 of 2)

Value	Logical unit formatted with protection information	Field in protection information	Extented INQUIRY Data VPD page bit value [f]	If check fails [d] [e], additional sense code
000b	Yes	LOGICAL BLOCK GUARD	GRD_CHK = 1 [g]	LOGICAL BLOCK GUARD CHECK FAILED
			GRD_CHK = 0	No check performed
		LOGICAL BLOCK APPLICATION TAG	APP_CHK = 1 [c] [g]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
			APP_CHK = 0	No check performed
		LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 [g]	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
			REF_CHK = 0	No check performed
	No	No protection information on the medium available to check		
001b [b]	Yes	LOGICAL BLOCK GUARD	Shall not	No check performed
		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed
	No	Error condition [a]		

[a] A verify operation to a logical unit that supports protection information (see 4.15) and has not been formatted with protection information shall fail with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

[b] If the logical unit does not support protection information the requested command should fail with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

[c] The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. ~~The method for acquiring this knowledge is~~ Knowledge of the LOGICAL BLOCK APPLICATION TAG field contents may be obtained by use of the VERIFY (32) command (see 5.XX) or by a method not defined by this standard.

[d] If an error is reported, the sense key shall be set to ABORTED COMMAND.

[e] If multiple errors occur, the selection of which error to report is not defined by this standard.

[f] See the Extented INQUIRY Data VPD page (see SPC-3) for a description of the GRD_CHK, APP_CHK, and REF_CHK bits.

[g] If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field set to FFFFh, the checking of all protection information shall be disabled for the associated logical block.

Table 55 — VRPROTECT **field with** BYTCHK **set to one - checking protection information read from the medium** (part 2 of 2)

Value	Logical unit formatted with protection information	Field in protection information	Extented INQUIRY Data VPD page bit value [f]	If check fails [d] [e], additional sense code
010b [b]	Yes	LOGICAL BLOCK GUARD	Shall not	No check performed
		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed
	No	Error condition [a]		
011b [b]	Yes	LOGICAL BLOCK GUARD	Shall not	No check performed
		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed
	No	Error condition [a]		
100b - 111b	Reserved			

[a] A verify operation to a logical unit that supports protection information (see 4.15) and has not been formatted with protection information shall fail with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

[b] If the logical unit does not support protection information the requested command should fail with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

[c] The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. ~~The method for acquiring this knowledge is~~ Knowledge of the LOGICAL BLOCK APPLICATION TAG field contents may be obtained by use of the VERIFY (32) command (see 5.XX) or by a method not defined by this standard.

[d] If an error is reported, the sense key shall be set to ABORTED COMMAND.

[e] If multiple errors occur, the selection of which error to report is not defined by this standard.

[f] See the Extented INQUIRY Data VPD page (see SPC-3) for a description of the GRD_CHK, APP_CHK, and REF_CHK bits.

[g] If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field set to FFFFh, the checking of all protection information shall be disabled for the associated logical block.

If the BYTCHK bit is set to one, the device server shall check the protection information transferred from the application client data-out buffer based on the VRPROTECT field as described in table 56.

Table 56 — VRPROTECT field with BYTCHK set to one - checking protection information from the application client

Value	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails [d] [e], additional sense code
000b	Yes	No protection information received from application client to check		
	No	No protection information received from application client to check		
001b [b]	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
		LOGICAL BLOCK REFERENCE TAG	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	No	Error condition [a]		
010b [b]	Yes	LOGICAL BLOCK GUARD	Shall not	No check performed
		LOGICAL BLOCK APPLICATION TAG	May [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK REFERENCE TAG	May	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	No	Error condition [a]		
011b [b]	Yes	LOGICAL BLOCK GUARD	Shall not	No check performed
		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed
	No	Error condition [a]		
100b - 111b	Reserved			

[a] A verify operation to a logical unit that supports protection information (see 4.15) and has not been formatted with protection information shall fail with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

[b] If the logical unit does not support protection information the requested command should fail with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

[c] The device server may check the logical block application tag if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. ~~The method for acquiring this knowledge is~~ Knowledge of the LOGICAL BLOCK APPLICATION TAG field contents may be obtained by use of the VERIFY (32) command (see 5.XX) or by a method not defined by this standard.

[d] If an error is reported, the sense key shall be set to ABORTED COMMAND.

[e] If multiple errors occur, the selection of which error to report is not defined by this standard.

If the BYTCHK bit is set to one, the device server shall perform a byte-by-byte comparison of protection information transferred from the application client data-out buffer with protection information read from the medium based on the VRPROTECT field as described in table 57.

Table 57 — VRPROTECT **field with** BYTCHK **set to one - byte-by-byte comparison requirements** (part 1 of 2)

Value	Logical unit formatted with protection information	Field	Byte-by-byte Comparison	If compare fails [c] [d], additional sense code
000b	Yes	No protection information received from application client to compare. Only user data is compared within each logical block.		
	No	No protection information or the medium or received from application client to compare. Only user data is compared within each logical block.		
001b [b]	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (APP_TAG_OWN = 1) [e]	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (APP_TAG_OWN = 0) [f]	Shall not	No compare performed
		LOGICAL BLOCK REFERENCE TAG	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	No	Error condition [a]		

[a] A verify operation to a logical unit that supports protection information (see 4.15) and has not been formatted with protection information shall fail with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

[b] If the logical unit does not support protection information the requested command should fail with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

[c] If an error is reported, the sense key shall be set to MISCOMPARE.

[d] If multiple errors occur, the selection of which error to report is not defined by this standard.

[e] If the APP_TAG_OWN bit in the Control mode page (see SPC-3) is set to one, the logical block application tag shall not be modified by a device server.

[f] If the APP_TAG_OWN bit in the Control mode page (see SPC-3) is set to zero, the logical block application tag may be modified by a device server.

Table 57 — VRPROTECT **field with** BYTCHK **set to one - byte-by-byte comparison requirements** (part 2 of 2)

Value	Logical unit formatted with protection information	Field	Byte-by-byte Comparison	If compare fails [c] [d], additional sense code
010b [b]	Yes	LOGICAL BLOCK GUARD	Shall not	No compare performed
		LOGICAL BLOCK APPLICATION TAG (APP_TAG_OWN = 1) [e]	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (APP_TAG_OWN = 0) [f]	Shall not	No compare performed
		LOGICAL BLOCK REFERENCE TAG	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	No	Error condition [a]		
011b [b]	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (APP_TAG_OWN = 1) [e]	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (APP_TAG_OWN = 0) [f]	Shall not	No compare performed
		LOGICAL BLOCK REFERENCE TAG	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	No	Error condition [a]		
100b - 111b	Reserved			

[a] A verify operation to a logical unit that supports protection information (see 4.15) and has not been formatted with protection information shall fail with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

[b] If the logical unit does not support protection information the requested command should fail with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

[c] If an error is reported, the sense key shall be set to MISCOMPARE.

[d] If multiple errors occur, the selection of which error to report is not defined by this standard.

[e] If the APP_TAG_OWN bit in the Control mode page (see SPC-3) is set to one, the logical block application tag shall not be modified by a device server.

[f] If the APP_TAG_OWN bit in the Control mode page (see SPC-3) is set to zero, the logical block application tag may be modified by a device server.

5.23 VERIFY (12) command

The VERIFY (12) command (see table 58) requests that the device server verify the data on the medium. Data includes user data and protection information, if any.

Table 58 — VERIFY (12) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (AFh)							
1	VRPROTECT			DPO	Reserved	BLKVFY	BYTCHK	Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	(MSB)	VERIFICATION LENGTH						
9								(LSB)
10	Reserved							
11	CONTROL							

If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, the device server shall fail a VERIFY(12) with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

See the VERIFY (10) command (see 5.22) for a description of the fields in this command.

5.24 VERIFY (16) command

The VERIFY (16) command (see table 59) requests that the device server verify the data written on the medium. Data includes user data and protection information, if any.

Table 59 — VERIFY (16) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (8Fh)							
1	VRPROTECT			DPO	Reserved	BLKVFY	BYTCHK	Reserved
2	(MSB)	LOGICAL BLOCK ADDRESS						
9								(LSB)
10	(MSB)	VERIFICATION LENGTH						
13								(LSB)
14	Reserved							
15	CONTROL							

If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, the device server shall fail a VERIFY(16) with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

See the VERIFY (10) command (see 5.22) for a description of the fields in this command.

5.25 WRITE (6) command

The WRITE (6) command (see table 60) requests that the device server write the data transferred from the application client to the medium. Data transferred from the application client includes user data but does not include protection information.

Table 60 — WRITE (6) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (0Ah)							
1	Reserved			(MSB)				
2	LOGICAL BLOCK ADDRESS							
3								(LSB)
4	TRANSFER LENGTH							
5	CONTROL							

The cache control bits are not provided for this command. Block devices with cache memory may have values for the cache control bits that may affect the WRITE (6) command, however no default value is defined by this standard. If explicit control is required, the WRITE (10) command should be used.

The LOGICAL BLOCK ADDRESS field specifies the logical block where the write operation shall begin.

The TRANSFER LENGTH field specifies the number of contiguous logical blocks of data that shall be transferred. A TRANSFER LENGTH of zero indirectly specifies that 256 logical blocks shall be transferred. Any other value directly specifies the number of logical blocks that shall be transferred. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.4.2).

NOTE 6 - For the WRITE (10) command, a TRANSFER LENGTH of zero specifies that no logical blocks are transferred.

If a WRITE (6) command is received after protection information is enabled the device server shall set the protection information (see 4.15) as follows as it writes the logical block to the medium:

a) the LOGICAL BLOCK GUARD field to a properly generated CRC (see 4.15.3);
b) the LOGICAL BLOCK REFERENCE TAG field to: ~~a properly calculated logical block reference tag (see 4.15.2)~~
   A) the lower 4 bytes of the LBA if the RTO_EN bit in Long read capacity data (see 5.13) is set to zero; or
   B) FFFFFFFFh if the RTO_EN bit is set to one; and
c) the LOGICAL BLOCK APPLICATION TAG field to:
   A) FFFFh if the APP_TAG_OWN bit in the Control mode page (see SPC-3) is set to one; or
   B) any value if the APP_TAG_OWN bit in the Control mode page (see SPC-3) is set to zero.

5.26 WRITE (10) command

The WRITE (10) command (see table 61) requests that the device server write the data transferred from the application client to the medium. Data transferred from the application client includes user data and includes protection information as required by the WRPROTECT field and the medium format..

Table 61 — WRITE (10) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (2Ah)							
1	WRPROTECT			DPO	FUA	Reserved		Obsolete
2	(MSB)							
3								
4				LOGICAL BLOCK ADDRESS				
5								(LSB)
6	Reserved							
7	(MSB)							
8				TRANSFER LENGTH				(LSB)
9	CONTROL							

See the READ (10) command (see 5.9) for a definition of the DPO bit and the FUA bit. See the LOCK UNLOCK CACHE (10) command (see 5.4) for a definition of the LOGICAL BLOCK ADDRESS field.

The TRANSFER LENGTH field specifies the number of contiguous logical blocks of data that shall be transferred. A TRANSFER LENGTH of zero specifies that no logical blocks shall be transferred. This condition shall not be considered an error and no data shall be written. Any other value specifies the number of logical blocks that shall be transferred. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.4.2).

NOTE 7 - For the WRITE (6) command, a TRANSFER LENGTH of zero specifies that 256 logical blocks are transferred.

The device server shall check the protection information transferred from the application client data-out buffer

based on the WRPROTECT field as described in table 62

Table 62 — WRPROTECT **field** (part 1 of 2)

Value	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails [c] [f], additional sense code
000b	Yes [f]	No protection information received from application client to check		
	No	No protection information received from application client to check		
[b] [g] 001b	Yes [d]	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
		LOGICAL BLOCK REFERENCE TAG	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	No [a]	No protection information available to check		

[a] A write operation to a logical unit that supports protection information (see 4.15) and has not been formatted with protection information shall fail with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

[b] If the logical unit does not support protection information the requested command should fail with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

[c] The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. ~~The method for acquiring this knowledge is~~ Knowledge of the LOGICAL BLOCK APPLICATION TAG field contents may be obtained by use of the WRITE (32) command (see 5.XX) or by a method not defined by this standard.If an error is reported the sense key shall be set to ABORTED COMMAND.

[d] Device server shall preserve the contents of protection information (e.g., write to medium, store in non-volatile memory).

[e] The device server shall write a properly generated CRC (see 4.15.3.2) into the LOGICAL BLOCK GUARD field~~, a properly calculated logical block reference tag into the LOGICAL BLOCK REFERENCE TAG field (see 4.15.2),~~ and if the APP_TAG_OWN bit in the Control mode page (see SPC-3) is set to one, FFFFh into the LOGICAL BLOCK APPLICATION TAG field as it writes the logical block to the medium. If the APP_TAG_OWN bit is set to zero, the device server may set the LOGICAL BLOCK APPLICATION TAG field to any value. If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server shall write the lower 4 bytes of the LBA into the LOGICAL BLOCK REFERENCE TAG field (see 4.15.2). If the RTO_EN bit is set to a one, the device server shall write a value of FFFFFFFFh into the LOGICAL BLOCK REFERENCE TAG field.

[f] If multiple errors occur, the selection of which error to report is not defined by this standard.

[g] If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, WRITE (10), WRITE (12),and WRITE (16) commands with the WRPROTECT field set to 000b may be processed by the device server. If the RTO_EN bit is set to one, the device server shall fail WRITE (10), WRITE (12), and WRITE (16) commands if the WRPROTECT field is not set to 000b with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

Table 62 — WRPROTECT **field** (part 2 of 2)

Value	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails [c] [f], additional sense code
[b] [g] 010b	Yes [d]	LOGICAL BLOCK GUARD	Shall not	No check performed
		LOGICAL BLOCK APPLICATION TAG	May [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK REFERENCE TAG	May	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	No [a]	No protection information available to check		
[b] [g] 011b	Yes [d]	LOGICAL BLOCK GUARD	Shall not	No check performed
		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed
	No [a]	No protection information available to check		
100b - 111b	Reserved			

[a] A write operation to a logical unit that supports protection information (see 4.15) and has not been formatted with protection information shall fail with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

[b] If the logical unit does not support protection information the requested command should fail with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

[c] The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. ~~The method for acquiring this knowledge is~~ Knowledge of the LOGICAL BLOCK APPLICATION TAG field contents may be obtained by use of the WRITE (32) command (see 5.XX) or by a method not defined by this standard.If an error is reported the sense key shall be set to ABORTED COMMAND.

[d] Device server shall preserve the contents of protection information (e.g., write to medium, store in non-volatile memory).

[e] The device server shall write a properly generated CRC (see 4.15.3.2) into the LOGICAL BLOCK GUARD field~~, a properly calculated logical block reference tag into the LOGICAL BLOCK REFERENCE TAG field (see 4.15.2),~~ and if the APP_TAG_OWN bit in the Control mode page (see SPC-3) is set to one, FFFFh into the LOGICAL BLOCK APPLICATION TAG field as it writes the logical block to the medium. If the APP_TAG_OWN bit is set to zero, the device server may set the LOGICAL BLOCK APPLICATION TAG field to any value. If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server shall write the lower 4 bytes of the LBA into the LOGICAL BLOCK REFERENCE TAG field (see 4.15.2). If the RTO_EN bit is set to a one, the device server shall write a value of FFFFFFFFh into the LOGICAL BLOCK REFERENCE TAG field.

[f] If multiple errors occur, the selection of which error to report is not defined by this standard.

[g] If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, WRITE (10), WRITE (12),and WRITE (16) commands with the WRPROTECT field set to 000b may be processed by the device server. If the RTO_EN bit is set to one, the device server shall fail WRITE (10), WRITE (12), and WRITE (16) commands if the WRPROTECT field is not set to 000b with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

Editor's Note 3: Note b changed from proposal to mention the sense key/additional sense code used by a logical unit that checks reserved fields and doesn't know this field is no longer reserved.

Editor's Note 4: should there be a "Include?" column in this table to clarify when the application client is expected to include protection information in the application client data buffer?

5.29 WRITE AND VERIFY (10) command

The WRITE AND VERIFY (10) command (see table 65) requests that the device server write the data transferred from the application client to the medium and then verify that the data is correctly written. Data transferred from the application client includes user data and includes protection information as required by the WRPROTECT field and the medium format. The data is only transferred once from the application client to the device server.

Table 65 — WRITE AND VERIFY (10) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (2Eh)							
1	WRPROTECT			DPO	Reserved	EBP	BYTCHK	Obsolete
2	(MSB)							
3								
4				LOGICAL BLOCK ADDRESS				
5								(LSB)
6	Reserved							
7	(MSB)							
8				TRANSFER LENGTH				(LSB)
9	CONTROL							

See the LOCK UNLOCK CACHE (10) command (see 5.4) for a definition of the LOGICAL BLOCK ADDRESS field. See the WRITE (10) command (see 5.26) for a definition of the TRANSFER LENGTH field and the WRPROTECT field. See the READ (10) command (see 5.9) for a description of the DPO bit. See the WRITE (10) command (see 5.26) for a description of the EBP bit.

If the MODE SELECT command is implemented, and the Verify Error Recovery mode page (see 6.3.5) is also implemented, then the current settings in that mode page along with the AWRE bit in the Read-Write Error Recovery mode page (see 6.3.4) specify the verification error criteria. If these mode pages are not implemented, then the verification criteria is vendor-specific.

A byte check (BYTCHK) bit set to zero specifies that the device server perform a medium verification with no data comparison. A BYTCHK bit set to one specifies that the device server perform a byte-by-byte comparison of data written on the medium with the data transferred from the application client data-out buffer. If the comparison is unsuccessful for any reason, the device server shall return CHECK CONDITION status with the sense key set to MISCOMPARE with the appropriate additional sense code for the condition.

If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, the device server shall fail a WRITE AND VERIFY(10) with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

5.30 WRITE AND VERIFY (12) command

The WRITE AND VERIFY (12) command (see table 66) requests that the device server write the data transferred from the application client to the medium and then verify that the data is correctly written. Data includes user data and protection information, if any.

Table 66 — WRITE AND VERIFY (12) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (AEh)							
1	WRPROTECT			DPO	Reserved	EBP	BYTCHK	Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	(MSB)	TRANSFER LENGTH						
9								(LSB)
10	Reserved							
11	CONTROL							

If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, the device server shall fail a WRITE AND VERIFY(12) with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

See the WRITE AND VERIFY (10) command (see 5.29) for a description of the bits in this command.

5.31 WRITE AND VERIFY (16) command

The WRITE AND VERIFY (16) command (see table 67) requests that the device server write the data transferred from the application client to the medium and then verify that the data is correctly written. Data transferred from the application client includes user data and includes protection information as required by the WRPROTECT field and the medium format. The data is only transferred once from the application client to the device server.

Table 67 — WRITE AND VERIFY (16) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (8Eh)							
1	Reserved			DPO	Reserved	EBP	BYTCHK	Reserved
2	(MSB)	LOGICAL BLOCK ADDRESS						
9								(LSB)
10	(MSB)	TRANSFER LENGTH						
13								(LSB)
14	Reserved							
15	CONTROL							

If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, the device server shall fail a WRITE AND VERIFY(16) with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

See the WRITE AND VERIFY (10) command (see 5.29) for a description of the fields in this command.

5.34 WRITE SAME (10) command

The WRITE SAME (10) command (see table 70) requests that the device server write the single block of data transferred from the application client to the medium multiple times to consecutive multiple logical blocks. Data transferred from the application client includes user data and includes protection information as required by the WRPROTECT field and the medium format.

NOTE 8 - This command may be useful if large areas of the medium need to be written, prepared for certification, or otherwise initialized without the application client having to transfer all the data.

Table 70 — WRITE SAME (10) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (41h)							
1	WRPROTECT			Reserved		PBDATA	LBDATA	Obsolete
2	(MSB)							
5	LOGICAL BLOCK ADDRESS							(LSB)
6	Reserved							
7	(MSB)							
8	NUMBER OF BLOCKS							(LSB)
9	CONTROL							

See the LOCK UNLOCK CACHE (10) command (see 5.4) for a definition of the LOGICAL BLOCK ADDRESS field.

If the medium is formatted with protection information the value in the LOGICAL BLOCK REFERENCE TAG field received from the application client shall be placed into the LOGICAL BLOCK REFERENCE TAG field (see 4.5.2) of the first logical block written to the medium. Into each of the following logical blocks the logical block reference tag received in the data transferred from the application client, incremented by one, shall be placed into the LOGICAL BLOCK REFERENCE TAG field of that logical block (i.e., each logical block written to the medium has a logical block reference tag value of one greater than the previous logical block).

If the APP_TAG_OWN bit in the Control mode page (see SPC-3) is set to one, the logical block application tag received in the single block of data shall be placed in the LOGICAL BLOCK APPLICATION TAG field of each logical block. If the APP_TAG_OWN bit is set to zero, the logical block application tag received in the single block of data may be placed in the LOGICAL BLOCK APPLICATION TAG field of each logical block.

A logical block data (LBDATA) bit set to zero and a physical block data (PBDATA) bit set to zero specifies that the single block of data transferred from the application client shall be used without modification. A LBDATA bit set to one specifies that the device server replace the first four bytes of the data to be written to the current logical block with the LBA of the block currently being written.

A PBDATA bit set to one specifies that the device server replace the first eight bytes of the data to be written to the current physical sector with the physical address of the sector currently being written using the physical sector format (see 5.3.3).

If PBDATA and LBDATA are one the command shall be terminated with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with the appropriate additional sense code for the condition.

If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, the device server shall fail a WRITE SAME(10) with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

Editor's Note 5: 04-012 creates a new table to clarify the interaction of LBDATA and PBDATA, describes how LBDATA and PBDATA are handled if protection information is present (device server must recalculate the CRC, or refuse the whole command), and describe how they work if the LBA is 8 bytes long (from WRITE SAME (16))

The NUMBER OF BLOCKS field specifies the number of contiguous logical blocks to be written. A NUMBER OF BLOCKS field set to 0000h specifies that the device server write all the remaining logical blocks on the medium.

5.35 WRITE SAME (16) command

The WRITE SAME (16) command (see table 71) requests that the device server write the single block of data transferred from the application client to the medium multiple times to consecutive multiple logical blocks. Data transferred from the application client includes user data and includes protection information as required by the WRPROTECT field and the medium format.

Table 71 — WRITE SAME (16) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (93h)							
1	WRPROTECT			Reserved		PBDATA	LBDATA	Reserved
2	(MSB)			LOGICAL BLOCK ADDRESS				
9								(LSB)
10	(MSB)			NUMBER OF BLOCKS				
13								(LSB)
14	Reserved							
15	CONTROL							

If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, the device server shall fail a WRITE SAME(16) with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

See the WRITE SAME (10) command (see 5.34) for a description of the fields in this command.

Editor's Note 6: 04-012 describes how LBDATA and PBDATA fields work with an LBA that is 8 bytes long

5.38 XDWRITE (10) command

The XDWRITE (10) command (see table 74) requests that the target XOR the data transferred from the application client with the data on the medium. Data transferred from the application client includes user data

and includes protection information as required by the WRPROTECT field and the medium format. The resulting XOR data is stored by the target until it is retrieved by an XDREAD (10) command.

Table 74 — XDWRITE (10) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (50h)							
1	WRPROTECT			DPO	FUA	DISABLE WRITE	Reserved	
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	Reserved							
7	(MSB)	TRANSFER LENGTH						
8								(LSB)
9	CONTROL							

See the READ (10) command (see 5.9) for a definition of the DPO bit and the FUA bit. See the WRITE (10) command (see 5.26) for a definition of the WRPROTECT field.

If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, the device server shall fail a XDWRITE (10) with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

A DISABLE WRITE bit set to zero specifies that the data transferred from the application client shall be written to the medium after the XOR operation is complete. A DISABLE WRITE bit set to one specifies that the data shall not be written to the medium.

The LOGICAL BLOCK ADDRESS specifies the starting LBA of the data on which an XOR operation shall be performed with the data from the medium.

The TRANSFER LENGTH field specifies the number of logical blocks that shall be transferred from the application client and the number of logical blocks on which an XOR operation shall be performed with the data from the medium. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.4.2).

The resulting XOR data is retrieved by an XDREAD command with starting LOGICAL BLOCK ADDRESS field and TRANSFER LENGTH field that match, or are a subset of, the starting LOGICAL BLOCK ADDRESS field and TRANSFER LENGTH field of this command.

5.39 XDWRITE (32) command

The XDWRITE (32) command (see table 75) requests that the target XOR the data transferred from the application client with the data on the medium. Data transferred from the application client includes user data

and includes protection information as required by the WRPROTECT field and the medium format. The resulting XOR data is stored by the target until it is retrieved by an XDREAD (32) command.

Table 75 — XDWRITE (32) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Reserved							
6								
7	ADDITIONAL CDB LENGTH (18h)							
8	(MSB)							
9	SERVICE ACTION (0004h)							(LSB)
10	WRPROTECT			DPO	FUA	DISABLE WRITE	Reserved	
11	Reserved							
12	(MSB)							
19	LOGICAL BLOCK ADDRESS							(LSB)
20	Reserved							
27								
28	(MSB)							
31	TRANSFER LENGTH							(LSB)

If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, the device server shall fail a XDWRITE (32) with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

See the XDWRITE (10) command (see 5.38) and SPC-3 for a description of the fields in this command.

5.40 XDWRITEREAD (10) command

The XDWRITEREAD (10) command (see table 76) requests that the target XOR the data transferred from the application client with the data on the medium and return the resulting XOR data to the application client. Data transferred to and from the application client includes user data and includes protection information as required by the WRPROTECT field, the XORPINFO bit, and the medium format. This is the equivalent to an

XDWRITE (10) followed by an XDREAD (10) with the same LBA and transfer length. This command is only available on transport protocols supporting bidirectional commands.

Table 76 — XDWRITEREAD (10) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (53h)							
1	WRPROTECT			DPO	FUA	DISABLE WRITE	Reserved	XORPINFO
2	(MSB)							
5	LOGICAL BLOCK ADDRESS							(LSB)
6	Reserved							
7	(MSB)							
8	TRANSFER LENGTH							(LSB)
9	CONTROL							

If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, the device server shall fail a XDWRITEREAD (10) with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

See the XDWRITE (10) command (see 5.38) and XDREAD (10) command (see 5.36) for a description of the fields in this command.

5.41 XDWRITEREAD (32) command

The XDWRITEREAD (32) command (see table 77) requests that the target XOR the data transferred from the application client with the data on the medium and return the resulting XOR data to the application client. Data transferred to and from the application client includes user data and includes protection information as required by the WRPROTECT field, the XORPINFO bit, and the medium format. This is the equivalent to an

XDWRITE (32) followed by an XDREAD (32) with the same LBA and transfer length. This command is only available on transport protocols supporting bidirectional commands.

Table 77 — XDWRITEREAD (32) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Reserved							
6								
7	ADDITIONAL CDB LENGTH (18h)							
8	(MSB)							
9	SERVICE ACTION (0007h)							(LSB)
10	WRPROTECT			DPO	FUA	DISABLE WRITE	Reserved	XORPINFO
11	Reserved							
12	(MSB)							
19	LOGICAL BLOCK ADDRESS							(LSB)
20	Reserved							
27								
28	(MSB)							
31	TRANSFER LENGTH							(LSB)

If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, the device server shall fail a XDWRITEREAD (32) with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

See the XDWRITEREAD (10) command (see 5.40) and SPC-3 for a description of the fields in this command.

Additions to document SBC-2 r13

5.xx READ (32) Command

The READ (32) command (see table Table xx —) requests that the device server transfer data to the application client. The most recent data value written in the addressed logical block shall be returned.

Table xx — READ (32) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Reserved							
6								
7	ADDITIONAL CDB LENGTH (18h)							
8	(MSB)			SERVICE ACTION (TBDh)				
9								(LSB)
10	RDPROTECT			DPO	FUA	Reserved		
11	Reserved							
12	(MSB)			LOGICAL BLOCK ADDRESS				
19								(LSB)
20	(MSB)			INITIAL LOGICAL BLOCK REFERENCE TAG				
23								(LSB)
24	(MSB)			EXPECTED LOGICAL BLOCK APPLICATION TAG				
25								(LSB)
26	(MSB)			LOGICAL BLOCK APPLICATION TAG MASK				
27								(LSB)
28	(MSB)			TRANSFER LENGTH				
31								(LSB)

If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server shall fail the READ (32) command with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE. If the RTO_EN bit is set to one, the device server may process the command.

The INITIAL LOGICAL BLOCK REFERENCE TAG field contains the value of the LOGICAL BLOCK REFERENCE TAG expected on the first logical block of the range of logical blocks for this command. The checking enables and requirements are controlled by the RDPROTECT field. See the READ (10) command (see ~~5.2.8~~ 5.9) for a definition ~~description~~ of the RDPROTECT field. ~~checking enables and requirements.~~

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled ~~by~~ as defined in the RDPROTECT field (see ~~5.2.8~~ 5.9) and the APP_CHK bit in the Extented INQUIRY Data VPD page (see SPC-3), the EXPECTED LOGICAL BLOCK APPLICATION TAG field contains a value that is expected in the LOGICAL BLOCK APPLICATION TAG with the LOGICAL BLOCK APPLICATION TAG MASK applied in the protection information of logical blocks for this command.

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled ~~by~~ as defined in the RDPROTECT field (see ~~5.2.8~~ 5.9) and the APP_CHK bit in the Extented INQUIRY Data VPD page (see SPC-3), the LOGICAL BLOCK APPLICATION MASK field contains a value that is a bit mask for enabling the checking of the LOGICAL BLOCK APPLICATION TAG in the protection information for each logical block of the range of logical blocks for this command. A LOGICAL BLOCK APPLICATION TAG MASK bit set to one enables the checking of the corresponding bit in the EXPECTED LOGICAL BLOCK APPLICATION TAG field with the LOGICAL BLOCK APPLICATION TAG.

~~See 4.2.1.8 for reservation requirements for this command.~~ See the READ (10) command (see ~~5.2.8~~ 5.9) for a description of the other fields in this command.

5.xx WRITE (32) command

The WRITE (32) command (see table Table xx —) requests that the device server write the data transferred from the application client to the medium

Table xx — WRITE (32) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Reserved							
6								
7	ADDITIONAL CDB LENGTH (18h)							
8	(MSB)			SERVICE ACTION (TBDh)				
9								(LSB)
10	WRPROTECT			DPO	FUA	Reserved		
11	Reserved							
12	(MSB)			LOGICAL BLOCK ADDRESS				
19								(LSB)
20	(MSB)			INITIAL LOGICAL BLOCK REFERENCE TAG				
23								(LSB)
24	(MSB)			EXPECTED LOGICAL BLOCK APPLICATION TAG				
25								(LSB)
26	(MSB)			LOGICAL BLOCK APPLICATION TAG MASK				
27								(LSB)
28	(MSB)			TRANSFER LENGTH				
31								(LSB)

If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server shall fail the WRITE (32) command with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE. If the RTO_EN bit is set to one, the device server may process the command.

The INITIAL LOGICAL BLOCK REFERENCE TAG field contains the value of the LOGICAL BLOCK REFERENCE TAG expected on the first logical block of the range of logical blocks for this command. The checking enables and requirements are controlled by the WRPROTECT field. See the WRITE (10) command (see ~~5.2.31~~ 5.26) for a definition ~~description~~ of the WRPROTECT field. ~~checking enables and requirements.~~

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled ~~by~~ as defined in the WRPROTECT field (see ~~5.2.31~~ 5.26) and the APP_CHK bit in the Extented INQUIRY Data VPD page (see SPC-3), the EXPECTED LOGICAL BLOCK APPLICATION TAG field contains a value that is expected in the LOGICAL BLOCK APPLICATION TAG with the LOGICAL BLOCK APPLICATION TAG MASK applied in the protection information of logical blocks for this command.

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled ~~by~~ as defined in the WRPROTECT field (see ~~5.2.31~~ 5.26) and the APP_CHK bit in the Extented INQUIRY Data VPD page (see SPC-3), the LOGICAL BLOCK

APPLICATION MASK field contains a value that is a bit mask for enabling the checking of the LOGICAL BLOCK APPLICATION TAG in the protection information for each logical block of the range of logical blocks for this command. A LOGICAL BLOCK APPLICATION TAG MASK bit set to one enables the checking of the corresponding bit in the EXPECTED LOGICAL BLOCK APPLICATION TAG field with the LOGICAL BLOCK APPLICATION TAG.

~~See 4.2.1.8 for reservation requirements for this command.~~ See the WRITE (10) command (see 5.2.31) for a description of the other fields in this command.

5.xx WRITE AND VERIFY (32) Command

The WRITE AND VERIFY (32) command (see table Table xx —) requests that the device server write the data transferred from the application client to the medium and then verify that the data and protection information, if any, is correctly written. The data is only transferred once from the application client to the device server.

Table xx — WRITE AND VERIFY (32) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Reserved							
6								
7	ADDITIONAL CDB LENGTH (18h)							
8	(MSB)							
9	SERVICE ACTION (TBDh)							(LSB)
10	WRPROTECT			DPO	Reserved	EBP	BYTCHK	Reserved
11	Reserved							
12	(MSB)							
19	LOGICAL BLOCK ADDRESS							(LSB)
20	(MSB)							
23	INITIAL LOGICAL BLOCK REFERENCE TAG							(LSB)
24	(MSB)							
25	EXPECTED LOGICAL BLOCK APPLICATION TAG							(LSB)
26	(MSB)							
27	LOGICAL BLOCK APPLICATION TAG MASK							(LSB)
28	(MSB)							
31	TRANSFER LENGTH							(LSB)

If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server shall fail the WRITE and VERIFY (32) command with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE. If the RTO_EN bit is set to one, the device server may process the command.

The INITIAL LOGICAL BLOCK REFERENCE TAG field contains the value of the LOGICAL BLOCK REFERENCE TAG expected on the first logical block of the range of logical blocks for this command. See the WRITE AND VERIFY (10) command (see ~~5.2.34~~ 5.29) for a description of the checking enables and requirements.

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled ~~by~~ as defined in the WRPROTECT field (see ~~5.2.31~~ 5.26) and the APP_CHK bit in the Extented INQUIRY Data VPD page (see SPC-3), the EXPECTED LOGICAL BLOCK APPLICATION TAG field contains a value that is expected in the LOGICAL BLOCK APPLICATION TAG with the LOGICAL BLOCK APPLICATION TAG MASK applied in the protection information of logical blocks for this command.

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled ~~by~~ as defined in the WRPROTECT field (see ~~5.2.31~~ 5.26) and the APP_CHK bit in the Extented INQUIRY Data VPD page (see SPC-3), the LOGICAL BLOCK APPLICATION MASK field contains a value that is a bit mask for enabling the checking of the LOGICAL BLOCK APPLICATION TAG in the protection information for each logical block of the range of logical blocks for this command. A LOGICAL BLOCK APPLICATION TAG MASK bit set to one enables the checking of the corresponding bit in the EXPECTED LOGICAL BLOCK APPLICATION TAG field with the LOGICAL BLOCK APPLICATION TAG.

~~See 4.2.1.8 for reservation requirements for this command.~~ See the WRITE AND VERIFY(10) command (see ~~5.2.34~~ 5.29) for a description of the other fields in this command.

5.xx VERIFY (32) Command

The VERIFY (32) command (see table Table xx —) requests that the device server verify the data on the medium.

Table xx — VERIFY (32) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Reserved							
6								
7	ADDITIONAL CDB LENGTH (18h)							
8	(MSB)	SERVICE ACTION (TBD)						
9								(LSB)
10	VRPROTECT			DPO	Reserved	EBP	BYTCHK	Reserved
11	Reserved							
12	(MSB)	LOGICAL BLOCK ADDRESS						
19								(LSB)
20	(MSB)	INITIAL LOGICAL BLOCK REFERENCE TAG						
23								(LSB)
24	(MSB)	EXPECTED LOGICAL BLOCK APPLICATION TAG						
25								(LSB)
26	(MSB)	LOGICAL BLOCK APPLICATION TAG MASK						
27								(LSB)
28	(MSB)	TRANSFER LENGTH						
31								(LSB)

If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server shall fail the VERIFY (32) command with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE. If the RTO_EN bit is set to one, the device server may process the command.

The INITIAL LOGICAL BLOCK REFERENCE TAG field contains the value of the LOGICAL BLOCK REFERENCE TAG expected on the first logical block of the range of logical blocks for this command. The checking enables and requirements are controlled by the VRPROTECT field. See the VERIFY (10) command (see ~~5.2.27~~ 5.22) for a definition ~~description~~ of the VRPROTECT field. ~~checking enables and requirements.~~

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled ~~by~~ as defined in the VRPROTECT field (see ~~5.2.27~~ 5.22) and the APP_CHK bit in the Extented INQUIRY Data VPD page (see SPC-3), the EXPECTED LOGICAL BLOCK APPLICATION TAG field contains a value that is expected in the LOGICAL BLOCK APPLICATION TAG with the LOGICAL BLOCK APPLICATION TAG MASK applied in the protection information of logical blocks for this command.

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled ~~by~~ as defined in the VRPROTECT field (see ~~5.2.27~~ 5.22) and the APP_CHK bit in the Extented INQUIRY Data VPD page (see SPC-3), the LOGICAL BLOCK APPLICATION MASK field contains a value that is a bit mask for enabling the checking of the LOGICAL BLOCK APPLICATION TAG in the protection information for each logical block of the range of logical blocks for this command. A LOGICAL BLOCK APPLICATION TAG MASK bit set to one enables the checking of the corresponding bit in the EXPECTED LOGICAL BLOCK APPLICATION TAG field with the LOGICAL BLOCK APPLICATION TAG.

~~See 4.2.1.8 for reservation requirements for this command.~~ See the VERIFY (10) command (see ~~5.2.27~~ 5.22) for a description of the other fields in this command.

5.xx WRITE SAME (32) Command

The WRITE SAME (32) command (see table Table xx —) requests that the device server write the single block of data transferred by the application client to the medium multiple times to consecutive multiple logical blocks.

Table xx — WRITE SAME (32) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1		CONTROL						
2	Reserved							
6								
7		ADDITIONAL CDB LENGTH (18h)						
8	(MSB)	SERVICE ACTION (TBD)						
9								(LSB)
10	WRPROTECT			Reserved		PBDATA	LBDATA	Reserved
11	Reserved							
12	(MSB)	LOGICAL BLOCK ADDRESS						
19								(LSB)
20	(MSB)	INITIAL LOGICAL BLOCK REFERENCE TAG						
23								(LSB)
24	(MSB)	EXPECTED LOGICAL BLOCK APPLICATION TAG						
25								(LSB)
26	(MSB)	Reserved						
27								(LSB)
28	(MSB)	TRANSFER LENGTH						
31								(LSB)

If the RTO_EN bit in Long read capacity data (see 5.13) is set to zero, the device server shall fail the WRITE SAME (32) command with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE. If the RTO_EN bit is set to one, the device server may process the command.

The INITIAL LOGICAL BLOCK REFERENCE TAG field contains the value of the LOGICAL BLOCK REFERENCE TAG expected in the ~~block of data~~ transfer~~red by~~ from the application client for this command and the LOGICAL BLOCK REFERENCE TAG for the first logical block written to the medium. The checking enables and requirements are controlled by the WRPROTECT field. See the WRITE (10) command (see ~~5.2.31~~ 5.26) for a definition ~~description~~ of the WRPROTECT field. ~~checking enables and requirements.~~

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled ~~by~~ as defined in the WRPROTECT field (see ~~5.2.31~~ 5.26) and the APP_CHK bit in the Extented INQUIRY Data VPD page (see SPC-3), the EXPECTED LOGICAL BLOCK APPLICATION TAG field contains a value that is expected in the LOGICAL BLOCK APPLICATION TAG with the LOGICAL BLOCK APPLICATION TAG MASK applied in the protection information ~~of~~ in the ~~block of data~~ transfer~~red by~~ from the application client for this command.

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled ~~by~~ as defined in the WRPROTECT field (see ~~5.2.31~~ 5.26) and the APP_CHK bit in the Extented INQUIRY Data VPD page (see SPC-3), the LOGICAL BLOCK APPLICATION MASK field contains a value that is a bit mask for enabling the checking of the LOGICAL BLOCK APPLICATION TAG in the protection information ~~of~~ in the ~~block of data~~ transfer~~red by~~ from the application client for this command. A LOGICAL BLOCK APPLICATION TAG MASK bit set to one enables the checking of the corresponding bit in the EXPECTED LOGICAL BLOCK APPLICATION TAG field with the LOGICAL BLOCK APPLICATION TAG.

~~See 4.2.1.8 for reservation requirements for this command.~~ See the WRITE SAME (10) command (see ~~5.2.38~~ 5.34) for a description of the other fields in this command.

Changes to document SPC-3 r17

7.6.5 Extended INQUIRY Data VPD page

The Extended INQUIRY Data VPD page (see table 1) provides the application client with a means to obtain information about the logical unit.

Table 1 — Extended INQUIRY Data VPD page

Bit / Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (86h)							
2	Reserved							
3	PAGE LENGTH (3Ch)							
4	Reserved				RFTG_OWN	GRD_CHK	APTG_CHK	RFTG_CHK
5	Reserved					HEADSUP	ORDSUP	SIMPSUP
6	Reserved							
63								

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field are as defined in 6.4.2.

The PAGE LENGTH field specifies the length of the following VPD page data and shall be set to 60. If the allocation length is less than the length of the data to be returned, the page length shall not be adjusted to reflect the truncation.

A guard check (GRD_CHK) bit set to zero indicates the device server does not check the LOGICAL BLOCK GUARD field in the protection information (see SBC-2) before transmitting it to an application client. A GRD_CHK bit set

to one indicates the device server checks the LOGICAL BLOCK GUARD field in the protection information before transmitting it to an application client. If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field containing FFFFh, the checking of the LOGICAL BLOCK GUARD field in the protection information shall not be performed for the associated logical block.

An application tag check (APTG_CHK) bit set to zero indicates the device server does not check the LOGICAL BLOCK APPLICATION TAG field in the protection information (see SBC-2) before transmitting it to an application client. An APTG_CHK bit set to one indicates the device server checks the LOGICAL BLOCK APPLICATION TAG field in the protection information before transmitting it to an application client. If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field containing FFFFh, the checking of the LOGICAL BLOCK APPLICATION TAG field in the protection information shall not be performed for the associated logical block.

A reference tag check (RFTG_CHK) bit set to zero indicates the device server does not check the LOGICAL BLOCK REFERENCE TAG field in the protection information (see SBC-2) before transmitting it to an application client. A RFTG_CHK bit set to one indicates the device server checks the LOGICAL BLOCK REFERENCE TAG field in the protection information before transmitting it to an application client. If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field containing FFFFh, the checking of the LOGICAL BLOCK REFERENCE TAG field in the protection information shall not be performed for the associated logical block.

A reference tag ownership (RFTG_OWN) bit set to zero indicates that the logical unit does not support application client ownership of the LOGICAL BLOCK REFERENCE TAG field in the protected information (see SBC-2). A RFTG_OWN bit set to one indicates that the logical unit supports application client ownership of the LOGICAL BLOCK REFERENCE TAG field.

A head of queue supported (HEADSUP) bit set to one shall indicate that the HEAD OF QUEUE task attribute (see SAM-3) is supported by the logical unit. A HEADSUP bit set to zero shall indicate that the HEAD OF QUEUE task attribute is not supported. If the HEADSUP bit is set to zero application clients should not specify the HEAD OF QUEUE task attribute as an Execute Command (see 4.2) procedure call argument.

An ordered supported (ORDSUP) bit set to one shall indicate that the ORDERED task attribute (see SAM-3) is supported by the logical unit. An ORDSUP bit set to zero shall indicate that the ORDERED task attribute is not supported. If the ORDSUP bit is set to zero application clients should not specify the ORDERED task attribute as an Execute Command procedure call argument.

A simple supported (SIMPSUP) bit set to one shall indicate that the SIMPLE task attribute (see SAM-3) is supported by the logical unit. Logical units that support the full task management model (see SAM-3) shall set the SIMPSUP bit to one. A SIMPSUP bit set to zero shall indicate that the SIMPLE task attribute is not supported. If the SIMPSUP bit is set to zero application clients should not specify the SIMPLE task attribute as an Execute Command procedure call argument.

SAM-3 defines how unsupported task attributes are processed.