T10/08-116 revision 2

Date: June 03, 2008 To: T10 Committee (SCSI) From: George Penokie (LSI) Subject: SBC-3 SPC-4: Protection Type 3 Reference Tag Clarification

1 Overview

When type 3 data protection is used the logical block reference tag should follow the same rules as the logical block application tag when it comes to the rules on whether or not it may be modified by the device server. To make this clear the following changes should be made to SBC-3 and SPC-4.

2 SBC-3 changes

2.0.1 Protection types

2.0.1.1 Protection types overview

The content of protection information is dependent on the type of protection to which a logical unit has been formatted.

The type of protection supported by the logical unit shall be indicated in the SPT field in the Extended INQUIRY Data VPD page (see SPC-4). The current protection type shall be indicated in the P_TYPE field in the READ CAPACITY(16) command (see 5.13).

An application client may format the logical unit to a specific type of protection using the <u>RTO_REQ bitEMTPINEO</u>field and PROTECTION FIELD USAGE field in the FORMAT UNIT command (see 5.2).

...

2.0.2 Protection information format

Table 1 defines the placement of protection information in a logical block.

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

Table 1 — User data and protection information format

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 4.17.4) of the contents of the USER DATA field.

The LOGICAL BLOCK APPLICATION TAG field is set by the application client. If the device server detects a:

- a) LOGICAL BLOCK APPLICATION TAG field set to FFFFh and type 1 protection (see 4.17.2.3) or type 2 protection (see 4.17.2.4) is enabled; or
- b) LOGICAL BLOCK APPLICATION TAG field set to FFFFh, LOGICAL BLOCK REFERENCE TAG field set to FFFF FFFFh, and type 3 protection (see 4.17.2.5) is enabled,

then the device server disables checking of all protection information for the logical block when reading from the medium. Otherwise, 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 ATO bit is set to zero in the Control mode page (see SPC-4). If the ATO bit is set to one in the Control mode page the device server shall not modify the LOGICAL BLOCK APPLICATION TAG field.

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 of the first logical block in the data-in buffer and/or data-out buffer shall contain the value specified in table 2.

Table 2 — Contents of the LOGICAL BLOCK REFERENCE TAG field of the first logical block in the data-in buffer and/or data-out buffer

Protection Type	Content of the LOGICAL BLOCK REFERENCE TAG field of the first logical block in the data-in buffer and/or data-out buffer
Type 1 protection (see 4.17.2.3)	The least significant four bytes of the LBA contained in the LOGICAL BLOCK ADDRESS field of the command.
Type 2 protection (see 4.17.2.4)	The value in the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field of the command.
Type 3 protection (see 4.17.2.5)	Not defined in this standard. <u>May be modified by a device server if the ATO bit is set to</u> zero in the Control mode page (see SPC-4). If the ATO bit is set to one in the Control mode page the device server shall not modify this field.

The LOGICAL BLOCK REFERENCE TAG field subsequent logical blocks in the data-in buffer and/or data-out buffer shall be set as specified in table 3.

Table 3 — Setting the LOGICAL BLOCK REFERENCE TAG field of the subsequent logical blocks in the data-in buffer and/or data-out buffer

Protection Type The content of the LOGICAL BLOCK REFERENCE TAG field of each subsequence block in the data-in buffer and/or data-out buffer			
Type 1 protection (see 4.17.2.3) and Type 2 protection (see 4.17.2.4)	The logical block reference tag of the previous logical block plus one.		
Type 3 protection (see 4.17.2.5)	Not defined in this standard. May be modified by a device server if the ATO bit is set to zero in the Control mode page (see SPC-4). If the ATO bit is set to one in the Control mode page the device server shall not modify this field.		

T10/08-116 revision 2

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.

2.1 FORMAT UNIT command

2.1.1 FORMAT UNIT command overview

The FORMAT UNIT command (see table 4) requests that the device server format the medium into application client accessible logical blocks as specified in the number of logical blocks and logical block length values received in the last mode parameter block descriptor (see 6.3.2) in a MODE SELECT command (see SPC-4). In addition, the device server may certify the medium and create control structures for the management of the medium and defects. The degree that the medium is altered by this command is vendor-specific.

If a device server receives a FORMAT UNIT command before receiving a MODE SELECT command with a mode parameter block descriptor the device server shall use the number of logical blocks and logical block length at which the logical unit is currently formatted (i.e., no change is made to the number of logical blocks and the logical block length of the logical unit during the format operation).

If any deferred downloaded code has been received as a result of a WRITE BUFFER command (see SPC-4), then that deferred downloaded code shall replace the current operational code.

Byte\Bit	7	6	5	4	3	2	1	0
0				OPERATION	CODE (04h)			
1	FMTPINFO_ LONGLIST FMTDATA CMPLST DEFECT LIST FORMAT					RMAT		
2		Vendor-specific						
3		Obsolete						
4								
5				CON	rol			

Table 4 — FORMAT UNIT command

•••

A format protection information (FMTPINFO) bitfield (see table 7) in addition with the PROTECTION FIELD USAGE field (see 2.1.2.2) specifies if the device server enables or disables the use of protection information.

The reference tag own request (RTO_REQ) bit (see table 7) specifies whether the application client or the device server has ownership of the LOGICAL BLOCK REFERENCE TAG field in protection information (see 4.17.3).

Following a successful format, the P_TYPE field in the READ CAPACITY (16) parameter data (see 5.13.1) indicates the type of protection currently in effect on the logical unit.

When protection information is written during a FORMAT UNIT command (i.e., the FMTPINFO bitfield is set to onea value greater than zero) protection information shall be written to a default value of FFFFFFF_FFFFFFF.

•••

2.1.2 FORMAT UNIT parameter list

2.1.2.1 FORMAT UNIT parameter list overview

....

2.1.2.2 Parameter list header

....

Byte\Bit	7	6	5	4	3	2	1	0	
0		Reserved					PROTECTION FIELD USAGE		
1	FOV	DPRY	DCRT	STPF	IP	Obsolete	IMMED	Vendor- specific	
2	(MSB)		·	DEFECT LIS					
3					STEENOTT			(LSB)	

Table 5 — Short parameter list header

The long parameter list header (see table 6) is used if the LONGLIST bit is set to one in the FORMAT UNIT CDB.

Byte\Bit	7	6	5	4	3	2	1	0	
0			Reserved	PROTECTION FIELD USAGE					
1	FOV	DPRY	DCRT	STPF	IP	Obsolete	IMMED	Vendor- specific	
2		Reserved							
3				Res	erved				
4	(MSB)	MSB)							
7			DEFECT LIST LENGTH (LSB)						

Table 6 — Long parameter list header

The PROTECTION FIELD USAGE field in combination with the FMTPINFO bit and the RTO_REQ bifieldt (see table 7) specifies the requested protection type (see 4.17.2)

Device server indication		on specification		Description
SPT ^a	PROTECT b	FMTPINFO	PROTECTION FIELD USAGE	Description
xxxb	0	<u>00b</u>	000b	The logical unit shall be formatted to type 0 protection ^c (see 4.17.2.2) resulting in the P_TYPE field ^d being set to 000b.
xxxb	0	<u>00b</u>	>000b	Illegal ^e
xxxb	0	<u>01b</u>	xxxb	Illegal ^f
xxxb	0	<u>1xb</u>	xxxb	Illegal ^f
xxxb	1	<u>00b</u>	000b	The logical unit shall be formatted to type 0 protection ^c (see 4.17.2.2) resulting in the P_TYPE field ^d being set to 000b.
xxxb	1	<u>00b</u>	>000b	Illegal ^e
xxxb	1	<u>01b</u>	xxxb	Illegal ^f
000b 001b 011b	1	<u>1</u> 0 <u>b</u>	000b	The logical unit shall be formatted to type 1 protection ^g (see 4.17.2.3) resulting in the P_TYPE field ^d being set to 000b.
000b 001b 011b	1	<u>1</u> 0 <u>b</u>	>000b	Illegal ^e
000b	1	<u>11b</u>	xxxb	Illegal ^f
001b	1	<u>11b</u>	000b	The logical unit shall be formatted to type 2 protection ^g (see 4.17.2.4) resulting in the P_TYPE field ^d being set to 001b.
001b	1	<u>11b</u>	>000b	Illegal ^e
011b	1	<u>11b</u>	000b	Illegal ^e
011b	1	<u>11b</u>	001b	The logical unit shall be formatted to type 3 protection. ^g (see 4.17.2.5) resulting in the P_TYPE field ^d being set to 010b.
011b	1	<u>11b</u>	>001b	Illegal ^e
010b	1	<u>1xb</u>	xxxb	Reserved
1xxb	1	<u>1</u> x <u>b</u>	xxxb	Reserved

Table 7 — FMTPINFO bit, RTO_REQ bit, field and PROTECTION FIELD USAGE field

^a See the Extended INQUIRY Data VPD page (see SPC-4) for the definition of the SPT field.

^b See the standard INQUIRY data (see SPC-4) for the definition of the PROTECT bit.

^c The device server shall format the medium to the logical block length specified in the mode parameter block descriptor of the mode parameter header (see SPC-4).

^d See the READ CAPACITY command (see 5.13.1) for the definition of the P_TYPE field.

^e The device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

^f The device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

^g The device server shall format the medium to the logical block length specified in the mode parameter block descriptor of the mode parameter header plus eight (e.g., if the logical block length is 512, then the formatted logical block length is 520). Following a successful format, the PROT_EN bit in the READ CAPACITY (16) parameter data (see 5.13.1) indicates whether protection information (see 4.17) is enabled.

2.2 ORWRITE command

....

—

...

Table	8 — ORPROTECT	field - checking p	rotection inform	ation from the data-out buffer (part 1 of 2)

Code	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 info	ormation received	d from application client to check		
0000	No	No protection info	ormation received	d from application client to check		
		LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED		
001b ^b	Yes	LOGICAL BLOCK APPLICATION TAG	May ^c	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
		LOGICAL BLOCK REFERENCE TAG	Shall <u>(except</u> for type 3) f	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
	No	Error condition ^a				
		LOGICAL BLOCK GUARD	Shall not	No check performed		
010b ^b	Yes	LOGICAL BLOCK APPLICATION TAG	May ^c	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
		LOGICAL BLOCK REFERENCE TAG	May ^f	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
	No	Error condition ^a				
		LOGICAL BLOCK GUARD	Shall not	No check performed		
011b ^b	Yes	LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed		
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed		
	No	Error condition ^a				
		LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED		
100b ^b	Yes	LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed		
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed		
	No	Error condition ^a				

Table 8 — ORPROTECT field - checking protection information from the data-out buffer (part 2 of 2	<u>')</u>
---	-----------

Code	Logical unit formatted with protection information	Field in Device protection server If check fails ^{d e} , additional ser information check							
		LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED					
101b ^b	Yes	LOGICAL BLOCK APPLICATION TAG	May ^c	LOGICAL BLOCK APPLICATION TAG CHECK FAILED					
		LOGICAL BLOCK REFERENCE TAG	May ^f	LOGICAL BLOCK REFERENCE TAG CHECK FAILED					
	No	Error condition ^a							
110b- 111b	Pacanyad								
forma sense b If the with 0 sense c The 0 mode field. d If an e If mu f If typ comp enab <u>know</u>	atted with protect e key set to ILLE logical unit does CHECK CONDIT e code set to INV device server ma e page (see SPC This knowledge error is reported, Itiple errors occu e 1 protection is baring it to the low led, <u>the ATO bit is</u> <u>ledge of the cont</u> ks the logical blow	ion information sha GAL REQUEST an not support protect ION status with the (ALID FIELD IN CD y check the logical -4) and if it has kno is obtained by a me the sense key sha r, the selection of w enabled, the device ver 4 bytes of the L s set to one in the C tents of the LOGICAL ck reference tag. if	Ill be terminated id the additional tion information is e sense key set to B. block application wledge of the co ethod not defined ll be set to ABOF which error to rep e server shall che BA associated w <u>control mode pag</u> <u>BLOCK REFEREN</u> it has knowledge						

2.3 READ (10) command

...

Code	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information ^h	Extended INQUIRY Data VPD page bit value ^g	lf check fails ^{df} , additional sense code		
			LOGICAL BLOCK GUARD	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED		
				grd_chk = 0	No check performed		
	Yes		LOGICAL BLOCK APPLICATION TAG	арр_снк = 1 ^с	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
000b		No	TAG	APP_CHK = 0	No check performed		
			LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 ⁱ	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
			TAG	REF_CHK = 0	No check performed		
	No		No protection information available to check				
		s Yes ^e	LOGICAL BLOCK GUARD	grd_chk = 1	LOGICAL BLOCK GUARD CHECK FAILED		
			GUARD	grd_chk = 0	No check performed		
	Yes		LOGICAL BLOCK APPLICATION TAG	арр_снк = 1 ^с	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
001b 101b ^b			TAG	APP_CHK = 0	No check performed		
1010			LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 ⁱ	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
			IAG	REF_CHK = 0	No check performed		
	No ^a	No protection in checking	formation availab	le to transmit to th	e data-in buffer or for		

Table 9 — RDPROTECT field (part 1 of 3)

Code	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information ^h	Extended INQUIRY Data VPD page bit value ^g	lf check fails ^d f, additional sense code	
			LOGICAL BLOCK GUARD	No check perform	ned	
	No.5	Vec 6	LOGICAL BLOCK APPLICATION TAG	арр_снк = 1 ^с	LOGICAL BLOCK APPLICATION TAG CHECK FAILED	
010b ^b	Yes	Yes ^e	140	APP_CHK = 0	No check performed	
0100			LOGICAL BLOCK REFERENCE TAG	REF_СНК = 1 ^і	LOGICAL BLOCK REFERENCE TAG CHECK FAILED	
			140	ref_chk = 0	No check performed	
	No ^a	No protection information available to transmit to the data-in buffer or for checking				
	Yes		LOGICAL BLOCK GUARD	No check performed		
011b ^b		Yes ^e	LOGICAL BLOCK APPLICATION TAG	No check performed		
0110			LOGICAL BLOCK REFERENCE TAG	No check performed		
	No ^a	No protection information available to transmit to the data-in buffer or for checking				
			LOGICAL BLOCK GUARD	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED	
				grd_chk = 0	No check performed	
100b ^b	Yes	Yes ^e	LOGICAL BLOCK APPLICATION TAG	No check perform	ned	
			LOGICAL BLOCK REFERENCE TAG	No check performed		
	No ^a	No protection information available to transmit to the data-in buffer or for checking				
110b - 111b	Reserved					

Table 9 — RDPROTECT fi	ield (part 2 of 3)
------------------------	--------------------

Table 9 — RDPROTECT field (part 3 of 3)

	Table 9 — RDPROTECT Heid (part 3 of 3)							
Code	Logical unit formatted with protection information	unit formattedShall device serverField in protection information 2Extended INQUIRY DataIf check fails d f, additional sense codewith protection information 2information hVPD page bit value gadditional sense code						
forma sens b If the termi addit c The c LOGIC Set to LOGIC Othe d If an e Trans f If mu g See h If the a) L b) L b) L f b) L f then i If typ to the prote the c know REA	atted with protect e key set to ILLE logical unit doe inated with CHE ional sense code device server sh CAL BLOCK APPLIC o one in the Con CAL BLOCK APPLIC rwise, this know error is reported smit protection in ltiple errors occut the Extended IN CHK bit, and the device server d .OGICAL BLOCK APPLIC orotection (see 4 .OGICAL BLOCK APPLIC FFF FFFFh, an the device server e 1 protection is contents of the LC vledge may be a D (32) command	tion information s EGAL REQUEST is not support protect CK CONDITION is e set to INVALID F all check the logic CATION TAG field. If trol mode page (s CATION TAG field and ledge may be acq l, the sense key sh formation to the of ur, the selection of QUIRY Data VPD REF_CHK bit. letects a: PPLICATION TAG field d type 3 protection er shall not check enabled, the device serve DGICAL BLOCK REFF cquired through th	hall be terminated and the additional ection information status with the set FIELD IN CDB. al block application f the READ (32) of ee SPC-4), this k and the LOGICAL BL puired by a methon hall be set to ABC data-in buffer. f which error to real page (see SPC-4) did set to FFFFh and d; or lid set to FFFFh and d; or lid set to FFFFh, if any protection inforce server checks ated with the logic ERENCE TAG field. the <u>EXPECTED</u> INITI- e 3 protection is designed.	d with CHECK COI I sense code set to n the requested con nse key set to ILLE on tag if it has know command (see 5.17 nowledge is acquir .OCK APPLICATION T d not defined by th DRTED COMMANI port is not defined 4) for the definition and type 1 protection LOGICAL BLOCK REF is enabled, formation in the assist the logical block reference al block. If type 2 protection AL LOGICAL BLOCK F	EGAL REQUEST and the vledge of the contents of the 1) is used and the ATO bit is red from the EXPECTED GAG MASK field in the CDB. his standard. D.			

2.4 VERIFY (10) command

....

Table 10 — VRPROTECT field with BYTCHK set to one - checking protection information from the data-out buffer (part 1 of 2)

Code	Logical unit formatted with protection information	Field in protection information	Device server check	lf check fails ^{d e} , additional sense code		
000b	Yes	No protection info	ormation received	d from application client to check		
0000	No	No protection information received from application client to check				
		LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED		
001b ^b	Yes	LOGICAL BLOCK APPLICATION TAG	May ^c	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
		LOGICAL BLOCK REFERENCE TAG	Shall <u>(except</u> for type 3) f	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
	No	Error condition ^a				
	Yes	LOGICAL BLOCK GUARD	Shall not	No check performed		
010b ^b		LOGICAL BLOCK APPLICATION TAG	May ^c	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
		LOGICAL BLOCK REFERENCE TAG	May ^f	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
	No	Error condition ^a				
	Yes	LOGICAL BLOCK GUARD	Shall not	No check performed		
011b ^b		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed		
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed		
	No	Error condition ^a				
	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED		
100b ^b		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed		
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed		
	No	Error condition ^a		·		

 Table 10 — VRPROTECT field with BYTCHK set to one - checking protection information from the data-out buffer (part 2 of 2)

Code	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails ^{d e} , additional sense code	
		LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED	
101b ^b	Yes	LOGICAL BLOCK APPLICATION TAG	May ^c	LOGICAL BLOCK APPLICATION TAG CHECK FAILED	
		LOGICAL BLOCK REFERENCE TAG	May ^f	LOGICAL BLOCK REFERENCE TAG CHECK FAILED	
	No	Error condition ^a			
110b- 111b	Reserved				
 ^a A verify operation to a logical unit that supports protection information (see 4.17) and has not been formatted with protection information shall be terminated with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. ^b If the logical unit does not support protection information the requested command should be terminated with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. ^c The device server may check the logical block application tag if the ATO bit is set to one in the Control mode page (see SPC-4) and if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. If the VERIFY (32) command (see 5.25) is used, this knowledge is obtained from the EXPECTED LOGICAL BLOCK APPLICATION TAG field and the LOGICAL BLOCK APPLICATION TAG field and the LOGICAL BLOCK APPLICATION TAG field in the CDB. ^c If multiple errors occur, the selection of which error to report is not defined by this standard. ^f If type 1 protection is enabled, the device server shall checks the logical block reference tag by comparing it to the lower 4 bytes of the LBA associated with the logical block reference tag if it has knowledge of the contents of the LOGICAL BLOCK REFERENCE TAG field, then₁₇ the device server checks the logical block reference tag if it has knowledge of the contents of the LOGICAL BLOCK REFERENCE TAG field, then₁₇ the device server checks the logical block reference tag if it has knowledge of the contents of the LOGICAL BLOCK REFERENCE TAG field, then₁₇ the device server checks the logical block reference tag if it has knowledge of the contents of the LOGICAL BLOCK REFERENCE TAG field, then₁₇ the device server checks the logical block reference tag if it has knowledge of the contents of the LOGICAL BLOCK REFERENCE TAG field, then₁₇ the device server checks the logical block refe					

If the BYTCHK bit is set to one, the device server shall perform a byte-by-byte comparison of protection information transferred from the data-out buffer with protection information read from the medium based on the VRPROTECT field as described in table 11.

Table 11 — VRPROTECT field with BYTCHK set to one	e - byte-by-byte comparison requirements (part 1 of 2)
---	--

Code	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.					
0000	No	No protection information or the medium or received from application client to compare. Only user data is compared within each logical block.					
		LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED			
		LOGICAL BLOCK APPLICATION TAG (ATO = 1) ^e	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
		LOGICAL BLOCK APPLICATION TAG (ATO = 0) ^f	Shall not	No compare performed			
001b ^b	Yes	LOGICAL BLOCK REFERENCE TAG <u>(not type 3)</u>	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
		LOGICAL BLOCK REFERENCE TAG (type 3 and ATO = 0)	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
		LOGICAL BLOCK REFERENCE TAG (type 3 and ATO = 1)	Shall not	No compare performed			
	No	Error condition ^a					
		LOGICAL BLOCK GUARD	Shall not	No compare performed			
		LOGICAL BLOCK APPLICATION TAG (ATO = 1) ^e	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
	Yes	LOGICAL BLOCK APPLICATION TAG (ATO = 0) ^f	Shall not	No compare performed			
010b ^b		LOGICAL BLOCK REFERENCE TAG <u>(not type 3)</u>	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
		LOGICAL BLOCK REFERENCE TAG (type 3 and ATO = 0)	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
		LOGICAL BLOCK REFERENCE TAG (type 3 and ATO = 1)	Shall not	No compare performed			
	No	Error condition ^a					

Table 11 — VRPROTECT field with BYTCHK set to one - byte-by-byte comparison requirements (part 2 of 2)

Code	Logical unit formatted with protection information	Field	Byte-by-byte Comparison	If compare fails ^{c d} , additional sense code			
		LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED			
		LOGICAL BLOCK APPLICATION TAG (ATO = 1) ^e	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
		LOGICAL BLOCK APPLICATION TAG (ATO = 0) ^f	Shall not	No compare performed			
011b 100b ^b	Yes	LOGICAL BLOCK REFERENCE TAG (not type 3)	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
		LOGICAL BLOCK REFERENCE TAG (type 3 and ATO = 0)	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
		LOGICAL BLOCK REFERENCE TAG (type 3 and ATO = 1)	Shall not	No compare performed			
	No	Error condition ^a					
	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED			
		LOGICAL BLOCK APPLICATION TAG (ATO = 1) ^e	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
101b ^b		LOGICAL BLOCK APPLICATION TAG (ATO = 0) ^f	Shall not	No compare performed			
		LOGICAL BLOCK REFERENCE TAG	Shall not	No compare performed			
	No	Error condition ^a					
110b - 111b	Reserved						
 ^a A verify operation to a logical unit that supports protection information (see 4.17) and has not been formatted with protection information shall be terminated with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. ^b If the logical unit does not support protection information the requested command should be terminated with CHECK CONDITION status with the sense key set to 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 ATO bit is set to one in the Control mode page (see SPC-4), the logical block application tag shall not be modified by a device server. 							

^f If the ATO bit is set to zero in the Control mode page (see SPC-4), the logical block application tag may be modified by a device server.

2.5 WRITE (10) command

••••

Code	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails ^{d i} , additional sense code			
000b	Yes ^{fgh}	No protection info	No protection information received from application client to check				
0005	No	No protection information received from application client to check					
		LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED			
001b ^b	Yes ^e	LOGICAL BLOCK APPLICATION TAG	May ^c	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
		LOGICAL BLOCK REFERENCE TAG	Shall <u>(except</u> for type 3) ^j	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
	No ^a	No protection info	ormation availat	ble to check			
		LOGICAL BLOCK GUARD	Shall not	No check performed			
010b ^b	Yes ^e	LOGICAL BLOCK APPLICATION TAG	May ^c	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
		LOGICAL BLOCK REFERENCE TAG	May ^j	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
	No ^a	No protection information available to check					
	Yes ^e	LOGICAL BLOCK GUARD	Shall not	No check performed			
011b ^b		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					
	Yes ^e	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED			
100b ^b		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed			
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed			
	No ^a	No protection info	ormation availat	ble to check			
		LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED			
101b ^b	Yes ^e	LOGICAL BLOCK APPLICATION TAG	May ^c	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
		LOGICAL BLOCK REFERENCE TAG	May ^j	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
	No ^a	No protection info	ormation availat	ble to check			
110b - 111b	Reserved						

Table 12 — WRPROTECT field (part 1 of 2)

Table 12 — WRPROTECT field (part 2 of 2)
--

Logical u formatte Code with protectic informati	d Field in protection on information	Device server check	lf check fails ^{d i} , additional sense code
 formatted with p key set to ILLEG If the logical unit with CHECK CC code set to INVA The device serv mode page (see field. If the WRIT LOGICAL BLOCK A Otherwise, this I If an error is rep Device server sl non-volatile mer If an error is rep Device server sl non-volatile mer If the P_TYPE fie server shall write of each of the w value of FFFFF If the ATO bit is s each LOGICAL BL value into each If multiple errors If type 1 protecti the lower 4 byte enabled and the then_the device LOGICAL BLOCK F through the EXPE type 3 protection device server has server may ched 	rotection information sl GAL REQUEST and the t does not support prot DNDITION status with t ALID FIELD IN CDB. er may check the logic SPC-4) and if it has k TE (32) command (see APPLICATION TAG field and knowledge is obtained orted, the sense key sl nall preserve the content orted, the logical blocks. If FFFh into the LOGICAL B et to one in the COGICAL B COCK APPLICATION TAG f LOGICAL BLOCK APPLICA is occur, the selection of on is enabled, the devi server checks the logic cererence tag field. If ECTED INITIAL LOGICAL B is enabled, the ATO b as knowledge of the co	hall be terminate e additional sen ection informati he sense key se al block applica nowledge of the 5.30) is used, t nd the LOGICAL I by a method no hall be set to AE nts of protection y generated CR READ CAPACI our bytes of eac the P_TYPE field BLOCK REFERENC I mode page (se ield. If the ATO b NTION TAG field. f which error to ce server check d with the logica wledge of the c cal block referent type 2 protectio LOCK REFERENC it is set to one in ntents of the LO erence tag. If typ	n information (e.g., write to medium, store in C (see 4.17.4.2) into each LOGICAL BLOCK GUARD TY (16) parameter data (see 5.13), the device h LBA into the LOGICAL BLOCK REFERENCE TAG field is not set to 000b, the device server shall write a CE TAG field of each of the written logical blocks. ee SPC-4), the device server shall write FFFFh into bit is set to zero, the device server may write any report is not defined by this standard. ts the logical block reference tag by comparing it to al block. If type 2 protection or type 3 protection is ontents of the LOGICAL BLOCK REFERENCE TAG field, nee tag if it has knowledge of the contents of the- n is enabled, then this knowledge may be acquired E TAG field in a WRITE (32) command (see 5.30). If n the Control mode page (see SPC-4), and the GICAL BLOCK REFERENCE TAG field, then the device be 3 protection is enabled, then the method for

2.6 WRITE (6) command

•••

If a WRITE (6) command is received after protection information is enabled the device server shall set the protection information (see 4.17) as follows as it writes each logical block to the medium:

- a) the LOGICAL BLOCK GUARD field set to a properly generated CRC (see 4.17.4);
- b) the LOGICAL BLOCK REFERENCE TAG field set to:
 - A) the least significant four bytes of the LBA, if type 1 protection (see 4.17.2.3) is enabled;-or
 - B) FFFFFFFh, if type 2 protection (see 4.17.2.4): or
 - C) <u>FFFFFFFh, if the ATO bit is set to one in the Control mode page (see SPC-4) and type 3</u> protection (see 4.17.2.5) is enabled; or
 - D) any value, if the ATO bit is set to zero in the Control mode page (see SPC-4) and type 3 protection (see 4.17.2.5) is enabled;

and

- c) the LOGICAL BLOCK APPLICATION TAG field set to:
 - A) FFFFh, if the ATO bit is set to one in the Control mode page (see SPC-4); or
 - B) any value, if the ATO bit is set to zero in the Control mode page (see SPC-4).

2.7 WRITE SAME (10) command

...

LBDATA	PBDATA	Description
0	0	 The device server shall write the single block of user data received from the data-out buffer to each logical block without modification. If the medium is formatted with type 1 protection information: a) the value in the LOGICAL BLOCK REFERENCE TAG field received in the single block of data from the data-out buffer shall be placed into the LOGICAL BLOCK REFERENCE TAG field of the first logical block written to the medium. Into each of the subsequent logical blocks, the device server shall place into the LOGICAL BLOCK REFERENCE TAG field the value of the previous logical block's LOGICAL BLOCK REFERENCE TAG field the value of the previous logical block's LOGICAL BLOCK REFERENCE TAG field plus one; b) If the ATO bit is set to one in the Control mode page (see SPC-4), 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 ATO bit is set to zero, the device server may write any value into the LOGICAL BLOCK APPLICATION TAG field received in the single block of data from the data-out buffer shall be placed in the single block of data field of each logical block.
		 If the medium is formatted with type 3 protection information: a) If the ATO bit is set to one in the Control mode page (see SPC-4), the logical block reference tag received in the single block of data shall be placed in the LOGICAL BLOCK REFERENCE TAG field of each logical block. If the ATO bit is set to zero, the device server may write any value into the LOGICAL BLOCK REFERENCE TAG field of each logical block; b) If the ATO bit is set to one in the Control mode page (see SPC-4), 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 ATO bit is set to zero, the device server may write any value into the LOGICAL BLOCK APPLICATION TAG field of each logical block. If the ATO bit is set to zero, the device server may write any value into the LOGICAL BLOCK APPLICATION TAG field of each logical block; and c) The value in the LOGICAL BLOCK GUARD field received in the single block of data from the data-out buffer shall be placed in the LOGICAL BLOCK GUARD field of each logical block;
0	1 ^a	The device server shall replace the first eight bytes of the block received from the data-out buffer to each physical sector with the physical address of the sector being written using the physical sector format (see 5.2.2.4.5).
		matted with protection information then the protection information shall be written to FFFFFF_FFFFFFFFFFFFFFFFFFFFFFFFFFFFF

LBDATA	PBDATA	Description
1 ^a	0	The device server shall replace the first four bytes of the block received from the data-out buffer with the least significant four bytes of the LBA of the block being written, ending with the least significant byte (e.g., if the LBA is 77665544_33221100h, 33221100h is written with 33h written first and 00h written last).
1	1	The device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
^a If the medium is formatted with protection information then the protection information shall be written to a default value of FFFFFFF_FFFFFFFF in each of the written logical blocks.		

3 SPC-4 changes

7.4.6 Control mode page

••••

An application tag owner (ATO) bit set to one specifies that the contents of the LOGICAL BLOCK APPLICATION TAGfield in the protection information (see SBC-2), if any, shall not be modified by the device server. An ATO bit set to zero specifies that the contents of the LOGICAL BLOCK APPLICATION TAG field in the protection information, if any, may be modified by the device server. If the ATO bit is set to zero, the device server shall ignore thecontents of the LOGICAL BLOCK APPLICATION TAG field in the protection when received from the application client.

If the ATO bit is set to zero the device server may modify the contents of the LOGICAL BLOCK APPLICATION TAG field and, depending on the protection type, may modify the contents of the LOGICAL BLOCK REFERENCE TAG field (see SBC-3). If the ATO bit is set to one the device server shall not modify the LOGICAL BLOCK APPLICATION TAG field and, depending on the protection type, shall not modify the contents of the LOGICAL BLOCK REFERENCE TAG field (see SBC-3)