T10/08-116 revision 1

Date: March 13, 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 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								
n - 1		-	USER DATA					
n	(MSB)		LOGICAL BLOCK GUARD					
n + 1		-						
n + 2	(MSB)		LOGICAL BLOCK APPLICATION TAG					
n + 3								
n + 4	(MSB)							
n + 7		-	LUG	JUAL BLUCK	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. 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.

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 subsequent logical 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.

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 ORWRITE command

••••

Table 4 — ORPROT	TECT field - checking pro	otection 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 information received 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				
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 ^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 4 — ORPROTECT field - 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	lf 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	110b - 111b Reserved					
^a An o forms sens ^b If the with sens ^c The mode field. ^d If an ^e If mu f If typ comp enab of the this s	 ^a An or write 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. This knowledge is obtained 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 If type 1 protection is enabled, the device server shall checks the logical block. If type 3 protection is enabled, 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. 					

2.2 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	grd_chk = 1	LOGICAL BLOCK GUARD CHECK FAILED		
			GUARD	grd_chk = 0	No check performed		
	Yes	No	LOGICAL BLOCK APPLICATION	арр_снк = 1 ^с	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
000b			TAG	арр_снк = 0	No check performed		
			LOGICAL BLOCK REFERENCE TAG	REF_СНК = 1 ^і	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
				REF_CHK = 0	No check performed		
	No		No protection information available to check				
	Yes	Yes ^e	LOGICAL BLOCK GUARD	grd_chk = 1	LOGICAL BLOCK GUARD CHECK FAILED		
				grd_chk = 0	No check performed		
			LOGICAL BLOCK APPLICATION	арр_снк = 1 ^с	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
0016 1016 ^b			140	APP_CHK = 0	No check performed		
			LOGICAL BLOCK REFERENCE	REF_СНК = 1 ^{-і}	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 5 — 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	If check fails ^{df} , additional sense code	
			LOGICAL BLOCK GUARD	No check performed		
	X	V P	LOGICAL BLOCK APPLICATION	арр_снк = 1 ^с	LOGICAL BLOCK APPLICATION TAG CHECK FAILED	
010b b	Yes	Yes ^c	TAG	арр_снк = 0	No check performed	
0100			LOGICAL BLOCK REFERENCE	REF_СНК = 1 ^і	LOGICAL BLOCK REFERENCE TAG CHECK FAILED	
			TAG	REF_CHK = 0	No check performed	
	No ^a	No protection information available to transmit to the data-in buffer or for checking				
b	Yes	Yes ^e	LOGICAL BLOCK GUARD	No check performed		
			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 performed		
			LOGICAL BLOCK REFERENCE TAG	No check performed		
	No ^a	No protection in checking	formation availab	le to transmit to th	e data-in buffer or for	
110b - 111b	Reserved					

Table 5 — RDPROTECT field (part 2 of 3)

Table 5 — RDPROTECT	field	(part 3 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 ^{df} , additional sense code
^a A rea form sens ^b If the term addir ^c The LOGI set t LOGI d If an ^e Tran f If mu ^g See h If the a) I b) I then i If typ to th prote the o know	ad operation to a atted with protect is key set to ILLE clogical unit doe inated with CHE tional sense cod device server sh CAL BLOCK APPLIC one in the Con CAL BLOCK APPLIC erwise, this know error is reported smit protection in ultiple errors occu- the Extended IN CHK bit, and the device server d COGICAL BLOCK APPLIC orotection (see 4 LOGICAL BLOCK APPLIC orotection (see 4 LOGICAL BLOCK APPLIC orotection (see 4 LOGICAL BLOCK APPLIC orotection (see 4 LOGICAL BLOCK APPLIC orotection is enabled contents of the LO viedge may be a D (32) command viedge is not def	a logical unit that s stion information s EGAL REQUEST s not support prot CK CONDITION s e set to INVALID I all check the logic CATION TAG field. I trol mode page (s CATION TAG field a ledge may be acc I, the sense key sin formation to the o ur, the selection of QUIRY Data VPD REF_CHK bit. letects a: PPLICATION TAG field d type 3 protectio er shall not check enabled, the device server of the LBA associa the device server DGICAL BLOCK REF cquired through the d (see 5.11). If typ ined by this stand	supports protection hall be terminated and the additional ection information status with the se FIELD IN CDB. (a) block application f the READ (32) of ee SPC-4), this k and the LOGICAL BL quired by a methon hall be set to ABC data-in buffer. f which error to react the set to FFFFh and d; or end set to FFFFh, in (see 4.17.2.5) is any protection infor- ce server checks ated with the logic ERENCE TAG field. The <u>EXPECTED</u> INITIAL e 3 protection is co ard.	n information (see d with CHECK CO al sense code set to n the requested co nse key set to ILLE on tag if it has know command (see 5.1° nowledge is acquir OCK APPLICATION T and to defined by th DRTED COMMANI eport is not defined 4) for the definition and type 1 protection LOGICAL BLOCK REF s enabled, formation in the as the logical block reference cal block. If type 2 cal block reference If type 2 protection AL LOGICAL BLOCK F	4.17) and has not been NDITION status with the o INVALID FIELD IN CDB. mmand should be EGAL REQUEST and the vledge of the contents of the 1) is used and the ATO bit is red from the EXPECTED TAG MASK field in the CDB. his standard. D. by this standard. D. by this standard. as of the GRD_CHK bit, the on (see 4.17.2.3) or type 2 FERENCE TAG field set to sociated logical block. efference tag by comparing it protection or type 3 e tag if it has knowledge of in is enabled, then this REFERENCE TAG field in a method for acquiring this

2.3 VERIFY (10) command

....

Table 6 — 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	No protection information received 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)	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 ^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 6 — 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	lf 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 veri forma sensitive sensitive forma sensitive forma sensitive forma sensitive formation of the field. C The of model field. LOGIO Othe of field field	 111b ^{111b} ¹¹⁰⁰¹¹⁰⁰ ^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 MASK field in the CDB. ^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 If type 1 protection is enabled, the device server shall checks the logical block. If type 2 protection or type 3 protection is enabled, the device server checks the logical block reference tag only if it has knowledge of the contents of the LOGICAL BLOCK REFERENCE TAG field. If type 2 protection is enabled, the device server checks the logical block REFERENCE TAG field in a VERIFY (32) command (see 5.25). If type 3 protection is enabled, then this knowledge may be acquired through the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field in a VERIFY (32) command (see 5.25). If type 3 protection is enabled, then this knowledge may be acquired through the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field in a VERIFY (32) command (see 5.25). If type 3 protection is enabled, then this knowledge ma				

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 7.

Table 7 — VRPROTECT field with BYTCHK set to one	- 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		
0005	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		
010b ^b	Yes	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		
		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				

Table 7 — 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 <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			
	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 veri	fy operation to a	a logical unit that support	ts protection info	rmation (see 4.17) and has not been	
forma	ted with protect	tion information shall be	terminated with	CHECK CONDITION status with the	
^b If the l	ogical unit does	s not support protection i	nformation the re	equested command should be terminated	
with C	HECK CONDIT	FION status with the sense	se key set to ILL	EGAL REQUEST and the additional	
sense	code set to IN	ALID FIELD IN CDB.			
^d If mult	iple errors occu	, the sense key shall be a ir, 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.

2.4 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	ormation rec	eived from application client to check	
0000	No	No protection info	ormation rec	eived 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 ^j	LOGICAL BLOCK REFERENCE TAG CHECK FAILED	
	No ^a	No protection info	ormation ava	ilable 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 info	ormation ava	ilable 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	No protection information available to check		
	Yes ^e	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED	
101b ^b		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 ava	ilable to check	
110b - 111b	Reserved				

Table 8 —	WRPROTECT	field	(part 1	of 2)
-----------	-----------	-------	---------	-------

Code	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails ^{d i} , additional sense code
 ^a A wrift form sense sense bill the with sense contemportal sense sense bill the with sense correct The model field LOGI Other diftered field of the field of the field writter blocc hill the server field writter blocc hill the into any i if the into any i if the into the intot the into the into the into the into the into the into the int	ite operation to atted with prote se key set to ILL e logical unit doe CHECK COND se code set to IN device server m e page (see SP . If the WRITE (CAL BLOCK APPL erwise, this know error is reporte ce server shall volatile memory device server shall volatile set to se so server shall volatile set to set	a logical unit that s ction information s EGAL REQUEST es not support prote ITION status with t IVALID FIELD IN (hay check the logic C-4) and if it has k 32) command (see ICATION TAG field a wledge is obtained d, the sense key sl preserve the conte (). hall write a property set to 000h in the e least significant for written logical block FFFFh into the LO o one in the Contro LOGICAL BLOCK APP cur, the selection of s enabled, the devi of the LBA associa ce server checks the CAL BLOCK REFEREN rough the EXPECTED). If type 3 protection standard.	L supports prof hall be termi and the addi ection inform he sense ke CDB. al block app nowledge of 5.30) is use nowledge of 5.30) is use nowledge of 5.30) is use nowledge of 5.30) is use nowledge of support by a method hall be set to not the LOGIC by a method hall be set to not protect y generated READ CAPA our bytes of ts. If the P_T DGICAL BLOCH DI mode page tag field. If the PLICATION TAG f which error ice server ch the logical blo NCE TAG field D INITIAL LOGIO	tection information (see 4.17) and has not been nated with CHECK CONDITION status with the tional sense code set to INVALID FIELD IN CDB. nation the requested command should be terminated y set to ILLEGAL REQUEST and the additional lication tag if the ATO bit is set to one in the Control the contents of the LOGICAL BLOCK APPLICATION TAG ed, this knowledge is obtained from the EXPECTED AL BLOCK APPLICATION TAG MASK field in the CDB. I not defined by this standard. ABORTED COMMAND. tion information (e.g., write to medium, store in CRC (see 4.17.4.2) into each LOGICAL BLOCK GUARD ACITY (16) parameter data (see 5.13), the device each LBA into the LOGICAL BLOCK REFERENCE TAG YPE field is not set to 000b, the device server shall K REFERENCE TAG field of each of the written logical e (see SPC-4), the device server shall write FFFFh he ATO bit is set to zero, the device server may write 3 field. to report is not defined by this standard. Hecks the logical block reference tag by comparing it logical block. If type 2 protection or type 3 protection hock reference tag- <u>only</u> if it has knowledge of the . If type 2 protection is enabled, then this knowledge ICAL BLOCK REFERENCE TAG field in a WRITE (32) d, then the method for acquiring this knowledge is

2.5 WRITE SAME (10) command

...

LBDATA	PBDATA	Description		
0	<u>РВДАТА</u> 0	 Description 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 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 of each logical block of data from the data-out buffer shall be placed in the single block of user data from the data-out buffer shall be placed in the LOGICAL BLOCK AUDICAL BLOCK GUARD field received in the single block of data shall be placed in the LOGICAL BLOCK REFERENCE TAG field of each logical block. If the medium is formatted with type 3 protection information: a) 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. 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. 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. 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 T		
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).		
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 n a defau	nedium is for It value of F	matted with protection information then the protection information shall be written to FFFFFF_FFFFFFFFFF in each of the written logical blocks.		

Table 9 — LBDATA bit and PBDATA bi	it
------------------------------------	----

3 SPC-4 changes

T10/08-116 revision 1

7.4.6 Control mode page

....

An application tag owner (ATO) bit set to one specifies that the contents of the LOCICAL 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 LOCICAL 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 LOCICAL BLOCK APPLICATION TAG field in the protection when received from theapplication client.

If the ATO bit is set to zero the device server may modify the contents of the LOGICAL BLOCK APPLICATION 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 BLOCK REFERENCE TAG field (see SBC-3)