Date: April 26, 2005 To: T10 Committee (SCSI) From: George Penokie (IBM/Tivoli) Subject: SBC-3, SPC-4: Application ownership of protetion information Reference Tag

### 1 Overview

Some of the current applications using proprietary end-to-end protection method would like to convert to using the protection defined in the SBC and SPC standards. However, the methods used require a larger application tag field than is currently defined. They also imbed information that performs a similar function to the reference tag field. As a result they would like to have the option to expand the application tag field to include the reference tag field.

The following proposal requests a bit be added to the control mode page that, when set to one, would prevent the device server from modifying the reference tag. A bit is also added to Extended INQUIRY Data VPD page to inform the application client if this options is supported.

### **Changes to SPC-4**

### 1.0.1 Control mode page

The Control mode page (see table 1) provides controls over SCSI features that are applicable to all device types (e.g., task set management and error logging). If a field in this mode page is changed while there is a task already in the task set, it is vendor specific whether the old or new value of the field applies to that task. The mode page policy (see 6.7) for this mode page shall be shared, or per I\_T nexus.

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF (0b)		PAGE CODE (	)Ah)			
1				PAGE LENGTH	l (0Ah)			
2		TST		TMF_ONLY	Reserved	D_SENSE	GLTSD	RLEC
3	(	QUEUE ALGORITHM MODIFIER			Reserved	QERR Obsol		Obsolete
4	VS	RAC	UA_INTL	CK_CTRL	SWP	Obsolete		
5	ATO	TAS	Rese	erved	<u>APOR</u>	AUTOLOAD MODE		
6				Obsolata				
7				Obsolete				
8	(MSB)							
9		BUSY TIMEOUT PERIOD (LSB)					(LSB)	
10	(MSB)							
11			EXTENDED SELF-TEST COMPLETION TIME (LSB)					(LSB)

#### Table 1 — Control mode page

An application owned reference tag (APOR) bit set to one specifies that the contents of the LOGICAL BLOCK REFERENCE TAG field in the protection information, if any, shall not be modified by the device server. An APOR bit set to zero specifics that the contents of a LOGICAL BLOCK REFERENCE TAG field in the protection information, if any, shall be modified as specified in SBP-3.

#### 1.0.2 Extended INQUIRY Data VPD page

The Extended INQUIRY Data VPD page (see table 2) provides the application client with a means to obtain information about the logical unit.

Bit Byte	7	6	5	4	3	2	1	0	
0	PERI	IPHERAL QUAL	IFIER	PERIPHERAL DEVICE TYPE					
1				PAGE CODE (8	36h)				
2		Reserved							
3		PAGE LENGTH (3Ch)							
4		Reserved			RTO GRD_CHK			REF_CHK	
5		Reserved			PRIOR_SUP	HEADSUP	ORDSUP	SIMPSUP	
6		Reserved NV_SUP V_SUP						V_SUP	
7				Deserved					
63	Reserved								

Table 2 — Extended INQUIRY Data VPD page

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. The relationship between the PAGE LENGTH field and the CDB ALLOCATION LENGTH field is defined in 4.3.4.6.

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

A reference tag ownership (RTO) field (see table 3) indicates if the logical unit supports application client ownership of the LOGICAL BLOCK REFERENCE TAG field in the protection information (see SBC-3), if any.

<u>Code</u>	Definition
<u>00b</u>	Indicates that the logical unit does not support application client ownership of the LOGICAL BLOCK REFERENCE TAG field in the protection information (see SBC-3).
<u>01b</u>	If the RTO EN bit (see SBC-3) is set to one, indicates that the logical unit supports use by the application client of the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field in the CDB (see SBC-3).
<u>10b</u>	If the APOR bit (see 1.0.1) is set to one, indicates that the logical unit supports application client ownership of the LOGICAL BLOCK REFERENCE TAG field (see SBC-3).
<u>11b</u>	Reserved

Table 3 — <u>RTO field</u>

# Changes to SBC-3

### **1.0.3 Protection information format**

Table 4 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 (LSB)						
n + 1		-							
n + 2	(MSB)		LOGICAL BLOCK APPLICATION TAG (LSB)						
n + 3		-							
n + 4	(MSB)		LOGICAL BLOCK REFERENCE TAG (LSB)						
n + 7		-							

### Table 4 — 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 4.16.3) of the contents of the USER DATA field.

The LOGICAL BLOCK APPLICATION TAG field is set by the application client. A LOGICAL BLOCK APPLICATION TAG field set to FFFFh disables checking of all protection information for the logical block on read operations (e.g., READ (10) command, READ (12) command, READ (16) command, and READ (32) command, and the read portion of VERIFY (10) command, VERIFY (12) command, VERIFY (16) command, and VERIFY (32) command). 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-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.

Editor's Note 1: The above change is not related to this proposal but is an issue that needs to be resolved relating to protection. The issue is that the current wording implies that on both reads and writes if the application tag is set to FFFFh nothing else is checked. But that is only the case on read operations not write operations. If the working group deems this is inappropriate for this proposal it will be placed in a separate proposal.

The LOGICAL BLOCK REFERENCE TAG field is an incrementing value associated with the logical block. The LOGICAL BLOCK REFERENCE TAG field of the first logical block in the data-in buffer and/or data-out buffer depends on the command being processed:

- a) for a command that does not include an EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field (e.g., READ (16)) and the APOR bit (see SPC-4) is set to zero, the LOGICAL BLOCK REFERENCE TAG field of the first logical block in the data-in buffer and/or data-out buffer shall contain the least significant four bytes of the LBA contained in the LOGICAL BLOCK ADDRESS field of the command; and
- b) for a command that does include an EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field (e.g., READ (32)) and the APOR bit (see SPC-4) is set to zero, the LOGICAL BLOCK REFERENCE TAG field of the first logical block shall contain the value in the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field of the

command. These commands are only processed if the medium was formatted with application client ownership of the logical block reference tag (i.e., with the RTO\_REQ bit set to one in the FORMAT UNIT command (see 5.2)).

<u>If the APOR bit (see SPC-4) is set to zero, then each subsequent logical block in the data-in buffer and/or</u> data-out buffer shall contain a LOGICAL BLOCK REFERENCE TAG field with the logical block reference tag of the previous logical block plus one.

If the APOR bit (see SPC-4) is set to one, then the contents of the logical block reference tag are not defined by this standard and shall not be modified by the device server.

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.

### 1.1 READ (6) command

•••

The device server shall check the protection information read from the medium before returning status for the command as described in table 5.

Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information <sup>e</sup>	Extended INQUIRY Data VPD page bit value <sup>d</sup>	lf check fails <sup>bc</sup> , additional sense code		
			grd_chk = 1	LOGICAL BLOCK GUARD CHECK FAILED		
	No	GUARD	$GRD_CHK = 0$	No check performed		
Yes		LOGICAL BLOCK APPLICATION TAG	арр_снк = 1 <sup>а</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
			APP_CHK = $0$	No check performed		
		LOGICAL BLOCK REFERENCE TAG	ref_chk = 1 <sup>f</sup>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
			REF_CHK = 0	No check performed		
No		No protection information available to check				
a The device	sorver checks the	logical block applic	ation tag only if it ha	as knowledge of the contents of the		

Table 5 — Protection information checking for READ (6)

<sup>a</sup> 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.

<sup>b</sup> If an error is reported, the sense key shall be set to ABORTED COMMAND.

<sup>c</sup> If multiple errors occur, the selection of which error to report is not defined by this standard.

<sup>d</sup> See the Extended INQUIRY Data VPD page (see SPC-3) for the definitions of the GRD\_CHK bit, APP\_CHK bit, and REF\_CHK bit.

<sup>e</sup> 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 APOR bit is set to zero in the Control Mode Page (see SPC-3) and the RTO\_EN bit is set to zero in the READ CAPACITY (16) parameter data (see 5.11), the device server checks the logical block reference tag by comparing it to the lower 4 bytes of the LBA associated with the logical block. If the APOR bit is set to zero and 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. If the APOR bit is set to one the LOGICAL BLOCK REFERENCE TAG field. This knowledge may be acquired by a method not defined by this standard.

# 1.2 READ (10) command

•••

The device server shall check the protection information read from the medium before returning status for the command based on the RDPROTECT field as described in table 6.

Code	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information <sup>h</sup>	Extended INQUIRY Data VPD page bit value <sup>g</sup>	lf check fails <sup>df</sup> , additional sense code		
			LOGICAL BLOCK	grd_chk = 1	LOGICAL BLOCK GUARD CHECK FAILED		
			GUARD	$GRD_CHK = 0$	No check performed		
	Yes		LOGICAL BLOCK APPLICATION	арр_снк = 1 <sup>с</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
0006 '		No	TAG	APP_CHK = $0$	No check performed		
			LOGICAL BLOCK REFERENCE	REF_CHK = 1 <sup>j</sup>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
			TAG	$REF_CHK = 0$	No check performed		
	No		No protection information available to check				
		Yes <sup>e</sup>	LOGICAL BLOCK	grd_chk = 1	LOGICAL BLOCK GUARD CHECK FAILED		
			GUARD	$GRD_CHK = 0$	No check performed		
0041	Yes		LOGICAL BLOCK APPLICATION	арр_снк = 1 <sup>с</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
101b <sup>bi</sup>			140	APP_CHK = $0$	No check performed		
			LOGICAL BLOCK REFERENCE	REF_CHK = 1 <sup>j</sup>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
			TAG	$REF_CHK = 0$	No check performed		
	No <sup>a</sup>	No protection in checking	formation availab	le to transmit to th	e data-in buffer or for		

Table 6 — RDPROTECT field	(part 1 of 4)
---------------------------	---------------

Code	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information <sup>h</sup>	Extended INQUIRY Data VPD page bit value <sup>g</sup>	If check fails <sup>df</sup> , additional sense code		
			LOGICAL BLOCK GUARD	No check perform	ned		
010b <sup>b i</sup>		Yes <sup>e</sup>	LOGICAL BLOCK APPLICATION	арр_снк = 1 <sup>с</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
	Yes		TAG	APP_CHK = 0	No check performed		
			LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 <sup>j</sup>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
				ref_chk = 0	No check performed		
	No <sup>a</sup>	No protection information available to transmit to the data-in buffer or for checking					
			LOGICAL BLOCK GUARD	No check performed			
011b <sup>b i</sup>	Yes	Yes <sup>e</sup>	LOGICAL BLOCK APPLICATION TAG	No check performed			
			LOGICAL BLOCK REFERENCE TAG	No check performed			
	No <sup>a</sup>	No protection in checking	formation availab	le to transmit to th	e data-in buffer or for		

 Table 6 — RDPROTECT field (part 2 of 4)

Code	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information <sup>h</sup>	Extended INQUIRY Data VPD page bit value <sup>g</sup>	If check fails <sup>df</sup> , additional sense code	
	Yes	Yes <sup>e</sup>	LOGICAL BLOCK	grd_chk = 1	LOGICAL BLOCK GUARD CHECK FAILED	
			GUARD	grd_chk = 0	No check performed	
100b <sup>b i</sup>			LOGICAL BLOCK APPLICATION TAG	No check performed		
			LOGICAL BLOCK REFERENCE TAG	No check performed		
	No <sup>a</sup> No protection information available to transmit to the data-in buffer or for checking					
1 <u>10</u> b - 111b	Reserved					

Table 6 — RDPROTECT field (part 3 of 4)

Table 6 — RDPROTECT field	(part 4 of 4)
---------------------------	---------------

Code	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information <sup>h</sup>	Extended INQUIRY Data VPD page bit value <sup>g</sup>	If check fails <sup>df</sup> , additional sense code
<ul> <li>A real form sense</li> <li>b If the term addit</li> <li>c The LOGI set term LOGI other of the other other</li></ul>	ad operation to a atted with protect e key set to ILLE e logical unit doe inated with CHE tional sense code device server sh CAL BLOCK APPLIC o one in the Con CAL BLOCK APPLIC or one in the Con CAL BLOCK APPLIC erwise, this know error is reported smit protection in altiple errors occu- the Extended IN CHK bit, and the e application client the Extended IN CHK bit, and the e application client the Extended IN CHK bit, and the e application client the Extended IN CHK bit, and REA ce server. If the F D (12) command CK CONDITION e set to INVALID e set to INVALID e APOR bit is set to READ CAPACITY ence tag by com a bit is set to zero beignored.	I logical unit that s tion information s EGAL REQUEST s not support prot CK CONDITION s e set to INVALID I all check the logic CATION TAG field. If trol mode page (s CATION TAG field a ledge may be acc I, the sense key sin formation to the our, the selection of QUIRY Data VPD REF_CHK bit. Int or device serve otection information to zero in the R the command. If the AD (16) commands RTO_EN bit is set to ds, and READ (16) I status with the sec COMMAND OPE o zero in the Conf Y (16) parameter of paring it to the low D and the RTO_EN ecks the logical blow in the CDB (see 4 rence tag if it has I a may be acquired of the EXPECTED IN	supports protection hall be terminated and the additional ection information status with the se FIELD IN CDB. al block application f the READ (32) of ee SPC-3), this k and the LOGICAL BL puired by a methon hall be set to ABC data-in buffer. f which error to re- page (see SPC- on in the associated EAD CAPACITY the RTO_EN bit is se s with the RDPROT to one, the device ) commands with ense key set to IL RATION CODE. trol Mode Page (se data (see 5.11), the wer 4 bytes of the bit is set to one ( to ck reference tag .16.2). If the APOI knowledge of the by a method not ITIAL LOGICAL BLO	in information (see d with CHECK CO al sense code set to in the requested co nse key set to ILLE on tag if it has know command (see 1.3) anowledge is acquir _OCK APPLICATION T od not defined by th ORTED COMMANI eport is not defined 3) for the definition AL BLOCK APPLICATION ted logical block sh (16) parameter dat set to one, READ ( TECT field set to 000 server shall termin the RDPROTECT field LEGAL REQUES see SPC-3) and the he device server che a LBA associated w i.e., the command based on the EXPER- the bit is set to one the contents of the LOO adefined by this stat CK REFERENCE TAG	4.16) 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 ) is used and the ATO bit is red from the EXPECTED 'AG MASK field in the CDB. his standard. D. by this standard. D. by this standard. a (see 5.11), the device 10) commands, READ (12) bb may be processed by the nate READ (10) commands, eld not set to 000b with T and the additional sense e RTO_EN bit is set to zero in necks the logical block with the logical block. If the is a READ (32) command), ECTED INITIAL LOGICAL BLOCK the device server shall check DICAL BLOCK REFERENCE TAG andard. If the APOR bit is set field (see 1.3) in the CDB

# 1.3 READ (32) command

••••

See the READ (10) command (see 1.2) for the definitions of the GROUP NUMBER field, the RDPROTECT field, the DPO bit, the FUA bit, the FUA\_NV bit, the LOGICAL BLOCK ADDRESS field, and the TRANSFER LENGTH field.

If the APOR bit is set to zero in the Control Mode Page (see SPC-3) and When-checking of the LOGICAL BLOCK REFERENCE TAG field is enabled (see table 6 in 1.2), the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field contains the value of the LOGICAL BLOCK REFERENCE TAG field expected in the protection information of the first logical block accessed by the command instead of a value based on the LBA (see 4.16.2).

If the APOR bit is set to one and checking of the LOGICAL BLOCK REFERENCE TAG field is enabled see table 6 in 1.2.

If the ATO bit is set to one in the Control mode page (see SPC-3) and checking of the LOGICAL BLOCK APPLICATION TAG field is enabled (see table 6 in 1.2), the LOGICAL BLOCK APPLICATION TAG MASK field contains a value that is a bit mask for enabling the checking of the LOGICAL BLOCK APPLICATION TAG field in the protection information for each logical block accessed by the command. A LOGICAL BLOCK APPLICATION TAG MASK field bit set to one enables the checking of the corresponding bit of the EXPECTED LOGICAL BLOCK APPLICATION TAG field with the corresponding bit of the LOGICAL BLOCK APPLICATION TAG field in the protection information.

If the ATO bit is set to one in the Control mode page (see SPC-3) and checking of the LOGICAL BLOCK APPLICATION TAG field is disabled (see table 6 in 1.2), or if the ATO bit is set to zero, the LOGICAL BLOCK APPLICATION TAG MASK field and the EXPECTED LOGICAL BLOCK APPLICATION TAG field shall be ignored.

# 1.4 VERIFY (10) command

...

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 7.

# Table 7 — VRPROTECT field with BYTCHK set to zero - checking protection information read from the medium (part 1 of 3)

Code	Logical unit formatted with protection information	Field in protection information <sup>g</sup>	Extended INQUIRY Data VPD page bit value <sup>f</sup>	If check fails <sup>de</sup> , additional sense code
		LOGICAL	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED
		BLOCK GUARD	grd_chk = 0	No check performed
		LOGICAL BLOCK	арр_снк = 1 <sup>с</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
000b	Yes	APPLICATION TAG	APP_CHK = 0	No check performed
		LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 <sup>h</sup>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
			ref_chk = 0	No check performed
	No	No protection i	medium to check. Only user data is checked.	
		LOGICAL	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED
		BLOCK GUARD	grd_chk = 0	No check performed
		LOGICAL BLOCK	АРР_СНК = 1 <sup>с</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
001b 101b <sup>b</sup>	Yes	APPLICATION TAG	APP_CHK = 0	No check performed
		LOGICAL BLOCK	REF_CHK = 1 <sup>h</sup>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
		REFERENCE TAG	REF_CHK = 0	No check performed
	No	Error condition	а	

# Table 7 — VRPROTECT field with BYTCHK set to zero - checking protection information read from the medium (part 2 of 3)

Code	Logical unit formatted with protection information	Field in protection information <sup>g</sup>	Extended INQUIRY Data VPD page bit value <sup>f</sup>	If check fails <sup>de</sup> , additional sense code			
		LOGICAL BLOCK GUARD	No check performed				
010b <sup>b</sup>		LOGICAL BLOCK	APP_CHK = 1 <sup>c</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
	Yes	APPLICATION TAG	APP_CHK = 0	No check performed			
		LOGICAL BLOCK	REF_CHK = 1 <sup>h</sup>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
		REFERENCE TAG	$REF_CHK = 0$	No check performed			
	No	Error condition	а	a			
	Yes	LOGICAL BLOCK GUARD	No check perforn	ned			
011b <sup>b</sup>		LOGICAL BLOCK APPLICATION TAG	No check performed				
		LOGICAL BLOCK REFERENCE TAG	No check performed				
	No	Error condition	а				

 Table 7 — VRPROTECT field with BYTCHK set to zero - checking protection information read from the medium (part 3 of 3)

Code	Logical unit formatted with protection information	Field in protection information <sup>g</sup>	Extended INQUIRY Data VPD page bit value <sup>f</sup>	If check fails <sup>de</sup> , additional sense code		
		LOGICAL	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED		
		BLOCK GUARD	$GRD_CHK = 0$	No check performed		
100b <sup>b</sup>	Yes	LOGICAL BLOCK APPLICATION TAG	No check perform	ned		
		LOGICAL BLOCK REFERENCE TAG	No check perforn	ned		
	No	Error condition	n <sup>a</sup>			
1 <u>10</u> b - 111b	Reserved					
<ul> <li><sup>111b</sup></li> <li><sup>a</sup> A verify operation to a logical unit that supports protection information (see 4.16) 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.</li> <li><sup>b</sup> 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.</li> <li><sup>c</sup> The device server shall check the logical block application tag if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. If the VERIFY (32) command (see 5.23) is used and the ATO bit is set to one in the Control mode page (see SPC-3), this knowledge is acquired from the EXPECTED LOGICAL BLOCK APPLICATION TAG field and the LOGICAL BLOCK APPLICATION TAG MASK field in the CDB. Otherwise, this knowledge may be obtained by a method not defined by this standard.</li> <li><sup>d</sup> If an error is reported, the sense key shall be set to ABORTED COMMAND.</li> <li><sup>e</sup> If multiple errors occur, the selection of which error to report is not defined by this standard.</li> <li><sup>f</sup> See the Extended INQUIRY Data VPD page (see SPC-3) for the definitions of the GRD_CHK bit, the APP_CHK bit, and the REF_CHK bits.</li> <li><sup>g</sup> 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.</li> <li><sup>h</sup> If the APOR bit is set to zero in the Control Mode Page (see SPC-3) and the RTO_EN bit is set to zero in the READ CAPACITY (16) parameter data (see 5.11)(i.e., the command is a VERIFY (10) command, a VERIFY (12) command, or a VERIFY (16) command), the device server shall check the logical block reference tag based on the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field in the CDB (s</li></ul>						

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 8.

# Table 8 — VRPROTECT field with BYTCHK set to one - checking protection information read from the medium (part 1 of 2)

Code	Logical unit formatted with protection information	Field in protection information <sup>g</sup>	Extended INQUIRY Data VPD page bit value <sup>f</sup>	If check fails <sup>d e</sup> , additional sense code	
			GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED	
000b		GUARD	GRD_CHK = 0	No check performed	
	Yes	LOGICAL BLOCK	APP_CHK = 1 <sup>c</sup> <sup>g</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED	
			APP_CHK = 0	No check performed	
		LOGICAL BLOCK	REF_CHK = 1 <sup>h</sup>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED	
		REFERENCE TAG	REF_CHK = 0	No check performed	
	No	No protection inform	nation on the medi	um available to check	
001b		LOGICAL BLOCK GUARD	No check performed		
010b 011b	Yes	LOGICAL BLOCK APPLICATION TAG	No check performed		
<u>101b</u> b		LOGICAL BLOCK REFERENCE TAG	No check perforr	ned	
	No	Error condition <sup>a</sup>	•		

 Table 8 — VRPROTECT field with BYTCHK set to one - checking protection information read from the medium (part 2 of 2)

Code	Logical unit formatted with protection information	Field in protection information <sup>g</sup>	Extended INQUIRY Data VPD page bit value <sup>f</sup>	lf check fails <sup>d e</sup> , additional sense code
1 <u>10</u> b - 111b	Reserved			
a A v for set b If ti wit set c Th LOO ofti d If a e If n f Se g If ti che h If <u>ti</u> the BLO che RE API	verify operation matted with pro- matted with pro- nse key set to IL he logical unit de h CHECK CON nse code set to e device server GICAL BLOCK APF to one in the C GICAL BLOCK APF nerwise, this kno n error is report nultiple errors of e the Extended P_CHK bit, and the application cl ecking of all pro- he APOR bit is set e READ CAPAC RIFY (12) comr erence tag by co DCK REFERENCE eck the logical b FERENCE TAG fie DR bit is set to o the CDB shall be	to a logical unit that s tection information sh LEGAL REQUEST a oes not support protect DITION status with th INVALID FIELD IN Cl shall check the logica PLICATION TAG field. If to ontrol mode page (se PLICATION TAG field an owledge may be obta ted, the sense key sh ccur, the selection of INQUIRY Data VPD ( he REF_CHK bit. ient or device server of tection information sh et to zero in the Contr ITY (16) parameter da mand, or a VERIFY (10 omparing it to the low <u>ero and the RTO_EN b</u> shall check the logica TAG field in the CDB ( lock reference tag if i ald. This knowledge m ne the contents of the e ignored.	upports protection all be terminated v and the additional s ction information the sense key set to DB. al block application the VERIFY (32) co es SPC-3), this know d the LOGICAL BLOC ined by a method v all be set to ABOR which error to repor page (see SPC-3) detects a LOGICAL E vall be disabled for to Mode Page (see ata (see 5.11)(i.e., 16) command), the er 4 bytes of the L block reference ta (see 4.16.2). If the t has knowledge o hay be acquired by e EXPECTED INITIAL I	information (see 4.16) and has not been with CHECK CONDITION status with the bense code set to INVALID FIELD IN CDB. he requested command should be terminated ILLEGAL REQUEST and the additional tag if it has knowledge of the contents of the formmand (see 5.23) is used and the ATO bit is weldge is acquired from the EXPECTED CK APPLICATION TAG MASK field in the CDB. not defined by this standard. TED COMMAND. ort is not defined by this standard. for the definitions of the GRD_CHK bit, the BLOCK APPLICATION TAG field set to FFFFh, the the associated logical block. <b>a</b> SPC-3) and the RTO_EN bit is set to zero in the command is a VERIFY (10) command, a device server shall check the logical block BA associated with the logical block. If the , the command is a VERIFY (32) command), ag based on the EXPECTED INITIAL LOGICAL APOR bit is set to one the device server shall f the contents of the LOGICAL BLOCK a method not defined by this standard. If the LOGICAL BLOCK REFERENCE TAG field (see 1.3)

If the BYTCHK bit is set to one, the device server shall check the protection information transferred from the data-out buffer based on the VRPROTECT field as described in table 9.

# Table 9 — 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	If check fails <sup>d e</sup> , 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 <sup>b</sup>	Yes	LOGICAL BLOCK APPLICATION TAG	May <sup>c</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
		LOGICAL BLOCK REFERENCE TAG	Shall <sup>f</sup>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
	No	Error condition <sup>a</sup>	Error condition <sup>a</sup>				
	Yes	LOGICAL BLOCK GUARD	Shall not	No check performed			
010b <sup>b</sup>		LOGICAL BLOCK APPLICATION TAG	May <sup>c</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
		LOGICAL BLOCK REFERENCE TAG	May <sup>f</sup>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
	No	Error condition <sup>a</sup>	Error condition <sup>a</sup>				
		LOGICAL BLOCK GUARD	Shall not	No check performed			
011b <sup>b</sup>	Yes	LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed			
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed			
	No	Error condition <sup>a</sup>					
		LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED			
100b <sup>b</sup>	Yes	LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed			
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed			
	No	Error condition <sup>a</sup>					

 Table 9 — 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 <sup>d e</sup> , additional sense code		
		LOGICAL BLOCK GUARD	<u>Shall</u>	LOGICAL BLOCK GUARD CHECK FAILED		
<u>101b <sup>b</sup></u>	Yes	LOGICAL BLOCK APPLICATION TAG	<u>May <sup>c</sup></u>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
		LOGICAL BLOCK REFERENCE TAG	<u>May <sup>f</sup></u>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
	<u>No</u>	Error condition <sup>a</sup>				
1 <u>10</u> b - 111b	Reserved					
<ul> <li><sup>a</sup> A verify operation to a logical unit that supports protection information (see 4.16) 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.</li> <li><sup>b</sup> 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.</li> <li><sup>c</sup> 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-3) and if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field and the LOGICAL BLOCK APPLICATION TAG MASK field in the CDB. Otherwise, this knowledge is obtained by a method not defined by this standard.</li> <li><sup>d</sup> If an error is reported, the sense key shall be set to ABORTED COMMAND.</li> <li><sup>e</sup> If multiple errors occur, the selection of which error to report is not defined by this standard.</li> <li><sup>f</sup> If the APOR bit is set to zero in the Control Mode Page (see SPC-3) and the RTO_EN bit is set to zero in the READ CAPACITY (16) parameter data (see 5.11)(i.e., the command is a VERIFY (10) command, a VERIFY (12) command, or a VERIFY (16) command), the device server estall check checks the logical block reference tag by comparing it to the lower 4 bytes of the LBA associated with the logical block. If the RTO_EN bit is set to one (i.e., the command is a VERIFY (32) command), the device server has knowledge of the contents of the LOGICAL BLOCK REFERENCE TAG field, then the device server has knowledge of the contents of the LOGICAL BLOCK REFERENCE TAG field, then the device server has knowledge of the contents of the LOGICAL BLOCK REFERENCE TAG field, then the d</li></ul>						

I

I

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 10.

Table 10 — VRPROTECT field with BYTCHK set to one	- byte-by-byte comparison requ	uirements (part 1 of 2)
---	--------------------------------	-------------------------

Code	Logical unit formatted with protection information	Field	Byte-by-byte Comparison	If compare fails <sup>c d</sup> , additional sense code		
000b	Yes	No protection information received from application client to compare. Only data is compared within each logical block.				
0005	No	No protection information or the medium or received from application client compare. Only user data is compared within each logical block.				
	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED		
		LOGICAL BLOCK APPLICATION TAG (ATO = 1) <sup>e</sup>	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
001b <sup>b</sup>		LOGICAL BLOCK APPLICATION TAG (ATO = 0) <sup>f</sup>	Shall not No compare performed			
		LOGICAL BLOCK REFERENCE TAG	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
	No	Error condition <sup>a</sup>				

Table 10 — 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 <sup>c d</sup> , additional sense code
		LOGICAL BLOCK GUARD	Shall not	No compare performed
		LOGICAL BLOCK APPLICATION TAG (ATO = 1) <sup>e</sup>	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
010b <sup>b</sup>	Yes	LOGICAL BLOCK APPLICATION TAG (ATO = 0) $f$	Shall not	No compare performed
		LOGICAL BLOCK REFERENCE TAG	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
No Error condition <sup>a</sup>				
		LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
011b 1 <u>01</u> b b	Yes	LOGICAL BLOCK APPLICATION TAG (ATO = 1) <sup>e</sup>	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (ATO = 0) $f$	Shall not	No compare performed
		LOGICAL BLOCK REFERENCE TAG (APOR = 0)	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
		LOGICAL BLOCK REFERENCE TAG (APOR = 1)	<u>Shall not</u>	No compare performed
	No	Error condition <sup>a</sup>		
1 <u>10</u> b - 111b	Reserved			
<ul> <li><sup>a</sup> A verify operation to a logical unit that supports protection information (see 4.16) 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.</li> <li><sup>b</sup> 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.</li> <li><sup>c</sup> If an error is reported, the sense key shall be set to MISCOMPARE.</li> <li><sup>d</sup> If multiple errors occur, the selection of which error to report is not defined by this standard.</li> <li><sup>e</sup> If the ATO bit is set to one in the Control mode page (see SPC-3), the logical block application tag shall not be modified by a device server.</li> </ul>				

# 1.5 WRITE (6) command

The WRITE (6) command (see table 11) requests that the device server transfer the specified logical block(s) from the data-out buffer and write them. Each logical block transferred includes user data but does not include

protection information. Each logical block written includes user data and, if the medium is formatted with protection information enabled, protection information.

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

#### Table 11 — WRITE (6) command

The cache control bits are not provided for this command. Direct-access block devices with cache 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.

See the PRE-FETCH (10) command (see 5.3) for the definition of the LOGICAL BLOCK ADDRESS field.

The TRANSFER LENGTH field specifies the number of contiguous logical blocks of data that shall be transferred from the data-out buffer and written, starting with the logical block specified by the LOGICAL BLOCK ADDRESS field. A TRANSFER LENGTH field set to zero specifies that 256 logical blocks shall be written. Any other value specifies the number of logical blocks that shall be written. If the logical block address plus the transfer length exceeds the capacity of the medium, 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 LOGICAL BLOCK ADDRESS OUT OF RANGE. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.4.2).

NOTE 1 - For the WRITE (10) command, WRITE (12) command, WRITE (16) command, and WRITE (32) command, a TRANSFER LENGTH field set to 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 4.16) as follows as it writes each logical block to the medium:

- a) the LOGICAL BLOCK GUARD field set to a properly generated CRC (see 4.16.3);
- b) the LOGICAL BLOCK REFERENCE TAG field set to:
  - A) the least significant four bytes of the LBA, if <u>the APOR bit is set to zero in the Control Mode Page</u> (see SPC-3) and the RTO\_EN bit is set to zero in the READ CAPACITY (16) parameter data (see 5.11); or

B) FFFFFFFh, if <u>the APOR bit is set to one or</u> the RTO\_EN bit is set to one;

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-3); or
  - B) any value, if the ATO bit is set to zero in the Control mode page (see SPC-3).

### 1.6 WRITE (10) command

•••

The device server shall check the protection information transferred from the data-out buffer based on the WRPROTECT field as described in table 12.

Code	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails <sup>d i</sup> , additional sense code	
000b	Yes <sup>fgh</sup>	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 <sup>b j</sup>	Yes <sup>e</sup>	LOGICAL BLOCK APPLICATION TAG	May <sup>c</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED	
		LOGICAL BLOCK REFERENCE TAG	Shall <sup>k</sup>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED	
	No <sup>a</sup>	No protection information available to check			
	Yes <sup>e</sup>	LOGICAL BLOCK GUARD	Shall not	No check performed	
010b <sup>b j</sup>		LOGICAL BLOCK APPLICATION TAG	May <sup>c</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED	
		LOGICAL BLOCK REFERENCE TAG	May <sup>k</sup>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED	
	No <sup>a</sup>	No protection information available to check			
		LOGICAL BLOCK GUARD	Shall not	No check performed	
011b <sup>b j</sup>	Yes <sup>e</sup>	LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed	
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed	
	No <sup>a</sup>	No protection info	ormation ava	ilable to check	
		LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED	
100b <sup>b j</sup>	Yes <sup>e</sup>	LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed	
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed	
	No <sup>a</sup>	No protection info	ormation ava	ilable to check	

Table 12 –	- WRPROTECT	field	(part 1	of 3)
------------	-------------	-------	---------	-------

Code	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails <sup>d i</sup> , additional sense code
<u>101b</u> <sup>b j</sup>	<u>Yes</u>	LOGICAL BLOCK GUARD	<u>Shall</u>	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG	<u>May <sup>c</sup></u>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK REFERENCE TAG	<u>May_<sup>k</sup></u>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	<u>No </u> a	No protection information available to check		

Table 12 — WRPROTECT field (part 2 of 3)

Table 12 — WRPROTECT field (part 3 of 3)

Code	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails <sup>d i</sup> , additional sense code			
1 <u>10</u> b - 111b	1 <u>10</u> b - 111b Reserved						
110b- 111b         Reserved           a         A write operation to a logical unit that supports protection information (see 4.16) 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-3) and if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. If the WRITE (32) command (see 1.7) 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. Otherwise, this knowledge is obtained by a method not defined by this standard.           e         If an error is reported, the sense key shall be set to ABORTED COMMAND.           e         Device server shall write a properly generated CRC (see 4.16.3.2) into each LOGICAL BLOCK GUARD field.           g         If the APOR bit is set to zero in the CONTOI Mode Page (see SPC-3) and the RTO_EN bit is set to zero in the READ CAPACITY (16) parameter data (see 5.11), the device server shall write heleast significant four bytes of each LBA into the LOGICAL BLOCK REFERENCE TAG field of each of the written logical blocks.           ft the APOR bit is set to zero in the CONTOI mode page (see SPC-3) and the exore shall write FFFFh into each LOGICAL BLOCK APPLICATION TA							
the Lo base obtai	the LOGICAL BLOCK REFERENCE TAG field, then the device server checks the logical block reference tag based on its knowledge of the contents of the LOGICAL BLOCK REFERENCE TAG field. This knowledge is obtained by a method not defined by this standard. If the APOR bit is set to one the contents of the						

# 1.7 WRITE (32) command

...

See the WRITE (10) command (see 1.6) for the definitions of the GROUP NUMBER field, the WRPROTECT field, the DPO bit, the FUA bit, the FUA\_NV bit, the LOGICAL BLOCK ADDRESS field, and the TRANSFER LENGTH field.

If the APOR bit is set to zero in the Control Mode Page (see SPC-3) and When checking of the LOGICAL BLOCK REFERENCE TAG field is enabled (see table 12 in 1.6), then the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field contains the value of the LOGICAL BLOCK REFERENCE TAG field expected in the protection information of the first logical block accessed by the command instead of a value based on the LBA (see 4.16.2).

If the APOR bit is set to one and checking of the LOGICAL BLOCK REFERENCE TAG field is enabled see table 12 in 1.6.

If the ATO bit is set to one in the Control mode page (see SPC-3) and checking of the LOGICAL BLOCK APPLICATION TAG field is enabled (see table 12 in 1.6), the LOGICAL BLOCK APPLICATION TAG MASK field contains a value that is a bit mask for enabling the checking of the LOGICAL BLOCK APPLICATION TAG field in the protection information for each logical block accessed by the command. A LOGICAL BLOCK APPLICATION TAG MASK bit set to one enables the checking of the corresponding bit of the EXPECTED LOGICAL BLOCK APPLICATION TAG field with the corresponding bit of the LOGICAL BLOCK APPLICATION TAG field in the protection information.

If the ATO bit is set to one in the Control mode page (see SPC-3) and checking of the LOGICAL BLOCK APPLICATION TAG field is disabled (see table 12 in 1.6), or if the ATO bit is set to zero, the LOGICAL BLOCK APPLICATION TAG MASK field and the EXPECTED LOGICAL BLOCK APPLICATION TAG field shall be ignored.