Date: August 31, 2005 To: T10 Committee (SCSI)

From: George Penokie (IBM/Tivoli)

Subject: SBC-3, SPC-4: Application ownership of protection 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 the RTO_EN bit be expanded to a 3 bit field to allow different usages of the reference tag field (e.g., setting the RTO_EN field to 010b would have the effect of preventing 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 option is supported.

With this proposal the meaning of a logical unit that is formatted with RTO_EN set to one changes from failing all non-32-byte command except legacy for legacy operations (i.e., those with RDPROTECT, WRPROTECT, and VRPROTECT set to zero) to allowing all commands to execute.

Because the 32-byte commands now have two different ways to define the content of the reference tag so a bit is added to the CDB that indicates if the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG is to used or not.

Changes to SPC-4

1.0.1 Extended INQUIRY Data VPD page

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

Bit Byte	7	6	5	4	3	2	1	0
0	PERI	PHERAL QUAL	IFIER		PERIP	HERAL DEVICE	TYPE	
1				PAGE CODE (8	36h)			
2				Reserved				
3		PAGE LENGTH (3Ch)						
4	Rese	Reserved RTO GRD_CHK APP_CHK REF_CHK						REF_CHK
5		Reserved GROUP_SUP PRIOR_SUP HEADSUP ORDSUP SIMPSUP						SIMPSUP
6		Reserved NV_SUP V_SUP						V_SUP
7	December							
63		Reserved						

Table 1 — 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 2) 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>000b</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>001b</u>	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>010b</u>	Reserved
<u>011b</u>	Indicates that the logical unit supports application client ownership of the LOGICAL BLOCK REFERENCE TAG field (see SBC-3) and, if selected, the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field in the CDB.
<u>100b - 111b</u>	Reserved

Table 2 — <u>RTO</u> field

Changes to SBC-3

1.1 Protection information model

1.1.1 Protection information overview

The protection information model provides for protection of user data while it is being transferred between a sender and a receiver. Protection information is generated at the application layer and may be checked by any object associated with the I_T_L nexus. Once received, protection information is retained (e.g., written to medium, stored in non-volatile memory, or recalculated on read back) by the device server until overwritten. Power loss, hard reset, logical unit reset, and I_T nexus loss shall have no effect on the retention of protection information.

Support for protection information shall be indicated in the PROTECT bit in the standard INQUIRY data (see SPC-3).

For commands that are using protection information, the data in buffer and/or data-out buffer shall consist of logical blocks with both user data and protection information. For commands that are not using protection information, the data in buffer and/or data-out buffer shall consist of logical blocks with only user data.

If the logical unit is formatted with protection information and the EMDP bit is set to one in the Disconnect-Reconnect mode page (see SPC-3), then checking of the logical block reference tag within the service delivery subsystem without accounting for modified data pointers and data alignments may cause false errors when logical blocks are transmitted out of order.

1.1.2 Command processing restrictions

I

If the logical unit is formatted with protection information all commands in which the CDB contains a zero value in the RDPROTECT field, WRPROTECT field, and VRPROTECT field are valid. For those commands, the data-in buffer and/or data-out buffer associated with those commands shall consist of logical blocks with only user data.

If the logical unit is formatted with protection information not all commands in which the CDB contains a non-zero value in the RDPROTECT field, WRPROTECT field, and VRPROTECT field are valid. Commands that are allowed is indicated be the RTO_EN field returned in the READ CAPACITY (16) parameter data (see 1.6.2) as

specified in table 3. For allowed commands using protection information, the data-in buffer and/or data-out buffer shall consist of logical blocks with both user data and protection information.

RTO EN field a	Description
<u>000b</u>	For any command in which the CDB contains an EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field in the CDB shall be terminated with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID COMMAND OPERATION CODE.
<u>001b</u>	 For any command in which the CDB does not contain: a) an EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field; or b) a RDPROTECT field set to zero, WRPROTECT field set to zero, or VRPROTECT field set to zero, shall be terminated with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID COMMAND OPERATION CODE. For any command in which the CDB contains an EXPECTED INITIAL LOGICAL BLOCK. REFERENCE TAG field with the XLB INVALID bit set to one shall be terminated with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID bit set to one shall be terminated with CHECK.
<u>010b</u>	No restrictions placed on any commands as a result of a logical unit being formatted with protection information.
<u>011b - 111b</u>	Reserved
a Specified in the	e READ CAPACITY (16) parameter data (see 1.6.2).

Table 3 — RTO EN field protection information command processing restrictions

Editor's Note 1: I have received requests to eliminate the XLB_INVALID bit from the 32-byte CDBs. This can be done in one of two ways. Option one would be to make 32-byte commands illegal when formatted in the new mode. This is reasonable as the new mode effectively eliminates much of the function used in the 32-byte commands. Option two would be to allow the 32-byte commands but ignore the logical block application tag mask field and the expected logical block application tag field when formatted in the new mode.

1.1.3 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	R DATA			
n	(MSB)		LOGICAL BLOCK GUARD (LSB)					
n + 1		-						(LSB)
n + 2	(MSB)		LOGICAL BLOCK APPLICATION TAG -					
n + 3								
n + 4	(MSB)							
n + 7		-	LOG	JUAL BLUUK	KEFERENCE	TAG		(LSB)

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 <u>when reading from the medium</u>. 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 2: The above change as accepted in 05-101r1.

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)) the LOGICAL BLOCK REFERENCE TAG field of the first logical block in the data-in bufferand/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)) 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 onlyprocessed if the medium was formatted with application client ownership of the logical block referencetag (i.e., with the RTO_REQ bit set to one in the FORMAT UNIT command (see 5.2)).

Each subsequent logical block in the data-in buffer and/or data-out buffer shall contain a logical blockreference tag field with the logical block reference tag of the previous logical block plus one. I

I

I

The LOGICAL BLOCK REFERENCE TAG field of the first logical block in the data-in buffer and/or data-out buffer shall be set as specified in table 5.

Table 5 — Setting the LOGICAL I	OCK REFERENCE TAG field of the first logical block in the data-in buffe	<u>er</u>
	and/or data-out buffer	

RTO EN field a	Description
<u>000b</u>	The content of the LOGICAL BLOCK REFERENCE TAG field of the first logical block in the data-in buffer and/or data-out buffer is set to the least significant four bytes of the LBA contained in the LOGICAL BLOCK ADDRESS field of the command.
<u>001b</u>	For any command in which the CDB contains an EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field with the XLB INVALID bit 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 value in the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field of the command.
<u>010b</u>	 For any command in which the CDB contains: a) does not contain an EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field; or b) contains an EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field with the XLB INVALID bit is set to one, the contents of the LOGICAL BLOCK REFERENCE TAG field of the first the logical block in the data-in buffer and/or data-out buffer is vendor specific. For any command in which the CDB contains an EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field with the XLB INVALID bit is set to zero the LOGICAL BLOCK REFERENCE TAG field with the XLB INVALID bit 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 value in the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field of the command.
<u>011b - 111b</u>	Reserved
^a Specified in the	e READ CAPACITY (16) parameter data (see 1.6.2).

SBC-3, SPC-4: Application ownership of protection in-

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 6.

Table 6 — Setting the LOGICAL BLOCK REFERENCE TAG field of the subsequent logic	al blocks in the data-in
buffer and/or data-out buffer	

RTO EN field a	Description
<u>000b</u>	The content of the LOGICAL BLOCK REFERENCE TAG field of each subsequent logical block in the data-in buffer and/or data-out buffer shall contain the logical block reference tag of the previous logical block plus one.
<u>001b</u>	For any command in which the CDB contains an EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field with the XLB INVALID bit is set to zero the content of the LOGICAL BLOCK REFERENCE TAG field of each subsequent logical block in the data-in buffer and/or data-out buffer shall contain the logical block reference tag of the previous logical block plus one.
<u>010b</u>	For any command in which the CDB contains: a) does not contain an EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field; or b) contains an EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field with the XLB INVALID bit is set to one, the content of the LOGICAL BLOCK REFERENCE TAG field of subsequent logical blocks in the data-in buffer and/or data-out buffer is vendor specific. For any command in which the CDB contains an EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field with the XLB INVALID bit is set to zero the content of the LOGICAL BLOCK REFERENCE TAG field of each subsequent logical block in the data-in buffer and/or data-out buffer shall contain the logical block reference tag of the previous logical block plus one.
<u>011b - 111b</u>	Reserved
^a Specified in the	e READ CAPACITY (16) parameter data (see 1.6.2).

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.

1.2 FORMAT UNIT command

1.2.1 FORMAT UNIT command overview

•••

A format protection information (FMTPINFO) bit (see table 9) specifies if the device server enables or disables the use of protection information. set to zero specifies that the device server shall disable the use of protection information (see 4.16) and format the medium to the block length specified in the mode parameter blockdescriptor of the mode parameter header (see SPC-3). A FMTPINFO bit set to one specifies that the deviceserver shall enable the use of protection information (see 4.16) and format the medium to the block lengthspecified in the mode parameter block descriptor of the mode parameter header plus eight (e.g., if the blocklength is 512, then the formatted block length is 520). Following a successful format, the <u>RTO_EN bit field</u> in the READ CAPACITY (16) parameter data (see 1.6.2) indicates whether protection information (see 4.16) is enabled.

The reference tag own request (RTO_REQ) bit (see table 9) specifies whether the application client or the device server has ownership of the LOGICAL BLOCK REFERENCE TAG field in protection information (see 4.16.2). If the FMTPINFO bit is set to one, and the RTO_REQ bit is set to one, the device server shall enable application client ownership of the LOGICAL BLOCK REFERENCE TAG field. If the FMTPINFO bit set to one and the RTO_REQ bit is set to zero, the device server shall disable application client ownership (i.e., enable device server shall disable application client ownership) of the LOGICAL BLOCK REFERENCE TAG field. If the FMTPINFO bit is set to zero and the RTO_REQ bit is set to one the device server shall terminate the command with CHECK CONDITION status with the sense keyset to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. Following a

successful format, the RTO_EN bit field in the READ CAPACITY (16) parameter data (see 1.6.2) indicates if the application client or the device server owns the LOGICAL BLOCK REFERENCE TAG field.

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

....

1.2.1.1 Parameter list header

The parameter list headers (see table 7 and table 8) provide several optional format control parameters. Device servers that implement these headers provide the application client additional control over the use of the four defect sources, and the format operation. If the application client attempts to select any function not implemented by the device server, 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.

The short parameter list header (see table 7) is used if the LONGLIST bit is set to zero in the FORMAT UNIT CDB.

Byte\Bit	7	6	5	4	3	2	1	0
0			Reserved	PROTE	CTION FIELDS	USAGE		
1	FOV	DPRY	DCRT	STPF	IP	Obsolete	IMMED	Vendor specific
2	(MSB)							
3			DEFECT LIST LENGTH (LSB)					

Table 7 — Short parameter list header

The long parameter list header (see table 8) 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			PROTE	CTION FIELDS	<u>USAGE</u>
1	FOV	DPRY	DCRT	STPF	IP	Obsolete	IMMED	Vendor specific
2		Reserved						
3		Reserved						
4	(MSB)	ASB)						
7		DEFECT LIST LENGTH(LSB)						(LSB)

Table 8 — Long parameter list header

The PROTECTION FIELD USAGE field (see table 9) specifies the requested application tag and reference tag usage when the application client has ownership of the LOGICAL BLOCK REFERENCE TAG field in protection information (see 4.16.2).

<u>rto ^b</u>	PROTECT ^a	<u>FMTPINFO</u>	<u>RTO_REQ</u>	PROTECTION FIELD USAGE	Description		
<u>xxxb</u>	<u>0</u>	<u>0 ^d</u>	<u>0</u>	<u>000b</u>	The device server shall disable the use of protection information, if any (see 1.1).		
<u>xxxb</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>>000b</u>	<u>lllegal ^g</u>		
<u>xxxb</u>	<u>0</u>	<u>0</u>	1	<u>xxxb</u>	<u>lllegal ^f</u>		
<u>xxxb</u>	<u>0</u>	<u>1</u>	X	<u>xxxb</u>	<u>Illegal ^f</u>		
<u>xxxb</u>	<u>1</u>	<u>0 </u> <u>d</u>	<u>0</u>	<u>000b</u>	The device server shall disable the use of protection information, if any (see 1.1).		
<u>xxxb</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>>000b</u>	<u>lllegal ^g</u>		
<u>xxxb</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>xxxb</u>	<u>lllegal ^f</u>		
<u>000b</u> <u>001b</u> <u>011b</u>	1	<u>1 ^c ^e</u>	<u>0</u>	<u>000b</u>	The device server shall disable application client ownership (i.e., enable device server ownership) of the LOGICAL BLOCK REFERENCE TAG field in the CDB (see 1.1.3).		
<u>000b</u> <u>001b</u> <u>011b</u>	1	1	<u>0</u>	<u>>000b</u>	<u>lliegal ^g</u>		
<u>000b</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>xxxb</u>	<u>Illegal ^f</u>		
<u>001b</u> 011b	1	<u>1 ^c e</u>	1	<u>000b</u>	The logical unit shall be formatted to allow use by the application client of the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field in the CDB (see 1.1.3).		
<u>001b</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>001b</u>	<u>lllegal ^g</u>		
<u>001b</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>010b - 111b</u>	Reserved		
<u>011b</u>	The logical unit shall be formatted to allow application client ownership of the LOGICAL						
<u>011b</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>010b - 111b</u>	Reserved		
 ^a See the standard INQUIRY data (see SPC-3) for the definition of the PROTECT bit. ^b See the Extended INQUIRY Data VPD page (see SPC-3) for the definition of the RTO field. ^c The device server shall enable the use of protection information (see 4.16). ^d The device server shall format the medium to the block length specified in the mode parameter block descriptor of the mode parameter header (see SPC-3). ^e The device server shall format the medium to the block length specified in the mode parameter block descriptor of the mode parameter header plus eight (e.g., if the block length is 512, then the formatted block length is 520). Following a successful format, the PROT_EN bit in the READ CAPACITY (16) parameter data (see 1.6.2) indicates whether protection information (see 4.16) is enabled. ^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 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. 							

Table 9 — FMTPINFO bit, RTO REQ bit, and PROTECTION FIELD USAGE field

I

I

I

SBC-3, SPC-4: Application ownership of protection information Reference Tag

A format options valid (FOV) bit set to zero specifies that the device server shall use its default settings for the DPRY, DCRT, STPF, and IP bits. If the FOV bit is set to zero, the application client shall set these bits to zero. If the FOV bit is set to zero and any of the other bits listed in this paragraph are not set to zero, 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.

1.3 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 10.

Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information ^f	Extended INQUIRY Data VPD page bit value ^d	lf check fails ^{bc} , additional sense code				
			GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED				
		GUARD	grd_chk = 0	No check performed				
Yes		LOGICAL BLOCK APPLICATION TAG	арр_снк = 1 ^а	LOGICAL BLOCK APPLICATION TAG CHECK FAILED				
	No		$APP_CHK = 0$	No check performed				
		LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 ^g	LOGICAL BLOCK REFERENCE TAG CHECK FAILED				
			$REF_CHK = 0$	No check performed				
No		No protection inf	ormation available	to check				
LOGICAL BLC standard. ^b If an error is ^c If multiple e ^d See the Ext APP_CHK bit	 ^b If an error is reported, the sense key shall be set to ABORTED COMMAND. ^c If multiple errors occur, the selection of which error to report is not defined by this standard. ^d See the Extended INQUIRY Data VPD page (see SPC-3) for the definitions of the GRD_CHK bit, APP_CHK bit, and REF_CHK bit. 							
protection in	If the device server detects a LOCICAL BLOCK APPLICATION TAG field set to FFFFh, it shall not check any protection information in the associated logical block.							
	^f If the device server detects a:							
	a) LOGICAL BLOCK APPLICATION TAG field set to FFFFh and the RTO EN field is set to 000b or 001b in the READ CAPACITY (16) parameter data (see 1.6.2); or							
b) LOGICAL BLOCK APPLICATION TAG field set to FFFFh, LOGICAL BLOCK REFERENCE TAG field set to FFFF FFFFh, and the RTO_EN field is set to 010b,								
then the device server shall not check any protection information in the associated logical block. If the RTO_EN bit field is set to zero in the READ CAPACITY (16) parameter data (see 1.6), the device								
server chec	ks the logical block	k reference tag by a	comparing it to the lo	ower 4 bytes of the LBA associated				
with the logi	ical block. If the RT	O_EN bit field is no	t set to zero one, th	e device server checks the logical				
block refere	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.

Table 10 — Protection information checking for READ (6)

I

1.4 READ (10) command

...

I

I

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 11.

Code	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information ⁱ	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		
			GUARD	grd_chk = 0	No check performed		
	Yes		LOGICAL BLOCK APPLICATION TAG	арр_снк = 1 ^с	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
000b-		No	TAG	АРР_СНК = 0	No check performed		
			LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 ^k	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
				ref_chk = 0	No check performed		
	No		No protection information available to check				
		Yes ^e	LOGICAL BLOCK GUARD	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILEE		
			GUARD	grd_chk = 0	No check performed		
	Yes		LOGICAL BLOCK APPLICATION TAG	арр_снк = 1 ^с	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
001b <u>101b</u> ^{b-j}			TAG	APP_CHK = 0	No check performed		
1010			LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 ^k	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
			TAG	ref_chk = 0	No check performed		
	No ^a	No protection in checking	formation availab	ble to transmit to th	e data-in buffer or for		

Table 11 — RDPROTECT field	(part 1 of 4)
----------------------------	---------------

I

I

Code	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information ⁱ	Extended INQUIRY Data VPD page bit value ^g	lf check fails ^d f, additional sense code			
			LOGICAL BLOCK GUARD	No check perforr	ned			
010b ^{bj}		Yes ^e	LOGICAL BLOCK APPLICATION	арр_снк = 1 ^с	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
	Yes		TAG -	АРР_СНК = 0	No check performed			
			LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 ^k	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
				ref_chk = 0	No check performed			
	No ^a	No protection information available to transmit to the data-in buffer or for checking						
			LOGICAL BLOCK GUARD	No check performed				
011b ^{b-j}	Yes	Yes ^e	LOGICAL BLOCK APPLICATION TAG	No check performed				
0110 ~ 1			LOGICAL BLOCK REFERENCE TAG	No check performed				
	No ^a	No protection in checking	formation availab	ble to transmit to th	e data-in buffer or for			

Table 11 — RDPROTECT field	(part 2 of 4)
----------------------------	---------------

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

I

Table 11 — RDPROTECT field (part 3 of 4)

Code	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information ⁱ	Extended INQUIRY Data VPD page bit value ^g	If check fails ^{df} , additional sense code
b If the termi addit c The LOGIO set to LOGIO othe d If an e Trans f If mu g See h APP_ h If the d If an e Trans f If mu g See h If the the c the c d If an f If mu g See k APP_ i If the the c the c d If an f If mu g See k APP_ i If the the c d If an f If mu g See k APP_ i If the the c d If an f If mu g See k APP_ i If the the c d If an f If mu g See k APP_ i If the the c d If an f If mu g See k APP_ i If the the c d If the f If the the c d If the f	atted with protect e key set to ILLE logical unit doe nated with CHE ional sense cod device server sh CAL BLOCK APPLIC o one in the Con CAL BLOCK APPLIC rwise, this know error is reported smit protection in litiple errors occu the Extended IN CHK bit, and the application clien hecking of all pr application clien OGICAL BLOCK APPLIC CAL BLOCK APPLIC CHK bit, and the application clien OGICAL BLOCK APPLIC CAL BLOC	tion information s EGAL REQUEST s not support prot CK CONDITION s e set to INVALID F all check the logic CATION TAG field. If trol mode page (s CATION TAG field and ledge may be acc all check the logic CATION TAG field and ledge may be acc all check the logic CATION TAG field and ledge may be acc all check the logic CATION TAG field and all check the logic CATION TAG field and QUIRY Data VPD REF_CHK bit. Int or device serve PPLICATION TAG field all protection information to zero in the RE and READ (16) COMMAND OPE is set to zero in the gical block referen pgical block. If the ased on the EXPEC s knowledge of the cquired through th	hall be terminate and the additional ection information status with the se FIELD IN CDB. al block application f the READ (32) of ee SPC-3), this kend the LOGICAL BI juired by a method hall be set to ABO data-in buffer. f which error to re- page (see SPC- r detects a LOGIC on in the associat r detects a: end set to FFFFh a eter data (see 1.6 end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to FFFFh, d is set to 010b, rmation in the associat r detects a: end set to 010b, end s	d with CHECK CO al sense code set to in the requested co inse key set to ILLE on tag if it has know command (see 1.5) anowledge is acquir OCK APPLICATION T id not defined by th DRTED COMMANI eport is not defined 3) for the definition AL BLOCK APPLICATION The defined by th DRTED COMMANI eport is not defined 3) for the definition AL BLOCK APPLICATION and the RTO_EN fiel collect block sha and the RTO_EN fiel collect block sha and the RTO_EN fiel collect block sha and the RTO_EN fiel collect block reference to one the device collect block reference and the RDPROTECT fiel conc the device collect block reference collect block reference	EGAL REQUEST and the vledge of the contents of the) 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 HON TAC field set to FFFFh, hall be disabled. Id is set to 000b or 001b in ERENCE TAG field set to CK shall be disabled. - (see 1.6), the device server- mands, READ (12) Db may be processed by the hate READ (10) commands, old not set to 000b with T and the additional sense- tion the CDB (see cks the logical block ERENCE TAG field in the CDB (see cks the logical block ERENCE TAG field. This REFERENCE TAG field in a

1.5 READ (32) command

The READ (32) command (see table 12) requests that the device server read the specified logical block(s) and transfer them to the data-in buffer. Each logical block read includes user data and, if the medium is formatted with protection information enabled, protection information. Each logical block transferred includes user data and may include protection information, based on the RDPROTECT field and the medium format.

If the RTO_EN bit field is set to zero in the READ CAPACITY (16) parameter data (see 1.6.2), the device servershall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST-

I

and the additional sense code set to INVALID COMMAND OPERATION CODE. If the RTO_EN bit field is not set to zero one, the device server may process the command (see 1.1.3).

Byte\Bit	7	6	5	4	3	2	1	0	
0		OPERATION CODE (7Fh)							
1				100	NTROL				
2				Ros	erved				
5		_		i tee					
6		Reserved			I	GROUP NUMB	ER		
7			A	DDITIONAL C	DB LENGTH (1	18h)			
8	(MSB)			SERVICE AC)			
9		_	SERVICE ACTION (0009h)					(LSB)	
10		RDPROTECT	Г	DPO	FUA	Reserved	FUA_NV	Reserved	
11				Reserved				XLB INVALID	
12	(MSB)		- LOGICAL BLOCK ADDRESS						
19				LOGICAL DE		0		(LSB)	
20	(MSB)		- EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG						
23		_			AL BLOOK KL	I ERENCE TA		(LSB)	
24	(MSB)		- EXPECTED LOGICAL BLOCK APPLICATION TAG						
25		_							
26	(MSB)								
27		- LOGICAL BLOCK APPLICATION TAG MASK						(LSB)	
28	(MSB)		TRANSFER LENGTH —						
31				IRANSEI				(LSB)	

Table 12 — READ (32) command

See the READ (10) command (see 1.4) 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.

An expected logical block invalid (XLB_INVALID) bit set to zero specifies that the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field, EXPECTED LOGICAL BLOCK APPLICATION TAG field, and LOGICAL BLOCK APPLICATION TAG MASK field are valid. An XLB_INVALID bit set to one specifies that the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field shall not be used when checking of the LOGICAL BLOCK REFERENCE TAG field is enabled (see table 11 in 1.4). In this case the value expected in the LOGICAL BLOCK REFERENCE TAG field is not defined by this standard.

When checking of the LOGICAL BLOCK REFERENCE TAG field is enabled (see table 11 in 1.4) and the XLB INVALID bit is set to zero, 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 ATO bit is set to one in the Control mode page (see SPC-3) <u>the XLB_INVALID bit is set to zero</u>, and checking of the LOGICAL BLOCK APPLICATION TAG field is enabled (see table 11 in 1.4), 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.

The LOGICAL BLOCK APPLICATION TAG MASK field and the EXPECTED LOGICAL BLOCK APPLICATION TAG field shall be ignored if:

- a) the XLB_INVALID bit is set to one;
- b) the ATO bit is set to zero; or
- c) 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 11 in 1.4).

1.6 READ CAPACITY (16) command

1.6.1 READ CAPACITY (16) command overview

The READ CAPACITY (16) command (see table 13) requests that the device server transfer parameter data describing the capacity and medium format of the direct-access block device to the data-in buffer. This command is mandatory if the logical unit supports protection information (see 4.16) and optional otherwise. This command is implemented as a service action of the SERVICE ACTION IN operation code (see A.2). This command may be processed as if it has a HEAD OF QUEUE task attribute (see 4.11).

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

Table 13 — READ CAPACITY (16) command

See the READ CAPACITY (10) command (see 5.10) for definitions of the LOGICAL BLOCK ADDRESS field and the PMI bit.

The ALLOCATION LENGTH field specifies the maximum number of bytes that the application client has allocated for returned parameter data. An allocation length of zero indicates that no data shall be transferred. This condition shall not be considered as an error. The device server shall terminate transfers to the data-in buffer when the number of bytes specified by the ALLOCATION LENGTH field have been transferred or when all available data has been transferred, whichever is less. The contents of the parameter data shall not be altered to reflect the truncation, if any, that results from an insufficient allocation length.

1.6.2 READ CAPACITY (16) parameter data

The READ CAPACITY (16) parameter data is defined in table 14. Any time the READ CAPACITY (16) parameter data changes, the device server should establish a unit attention condition as described in 4.6.

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

Table 14 — READ CAPACITY (16) parameter data

The RETURNED LOGICAL BLOCK ADDRESS field and BLOCK LENGTH IN BYTES field of the READ CAPACITY (16) parameter data are the same as the in the READ CAPACITY (10) parameter data (see 5.10). The maximum value that shall be returned in the RETURNED LOGICAL BLOCK ADDRESS field is FFFFFFF_FFFFFF.

A reference tag own enable (RTO_EN) bit set to one indicates that application client ownership of the LOGICAL-BLOCK REFERENCE TAG field in protection information is enabled (i.e., the medium was formatted withprotection information (see 4.16) enabled and the RTO_REQ bit was set to one). An RTO_EN bit set to zeroindicates that application client ownership of the LOGICAL BLOCK REFERENCE TAG field in protection informationis disabled.

The reference tag owner (RTO_EN) field (see table 15) specifies the logical unit's current allowed application client usage of the LOGICAL BLOCK REFERENCE TAG field in protection information (see 4.16.2).

Code	Description
<u>000b</u>	Application client ownership of the LOGICAL BLOCK REFERENCE TAG field in protection information is disabled.
<u>001b</u>	An EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field; and b) a RDPROTECT field set to zero, WRPROTECT field set to zero, or VRPROTECT field set to zero.
<u>010b</u>	The logical unit supports application client ownership of the LOGICAL BLOCK REFERENCE TAG field and, if selected, the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field in the CDB (see 1.1.3).
<u>011b - 111b</u>	Reserved

Table 15 - RTO EN field

A PROT_EN bit set to one indicates that the medium was formatted with protection information (see 4.16) enabled. A PROT_EN bit set to zero indicates that the medium was not formatted with protection information enabled.

1.7 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 16.

Table 16 — VRPROTECT field with BYTCHK set to zero - checking protection information read from the medium (part 1 of 4)

Code	Logical unit formatted with protection information	Field in protection information ^h	Extended INQUIRY Data VPD page bit value ^f	If check fails ^{de} , 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 ^с	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
000b	Yes	APPLICATION TAG	APP_CHK = 0	No check performed			
		LOGICAL BLOCK	ref_chk = 1 ⁱ	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
		REFERENCE TAG	$REF_CHK = 0$	No check performed			
	No	No protection i	ection information on the medium to check. Only user data is checked.				
		LOGICAL BLOCK GUARD	grd_chk = 1	LOGICAL BLOCK GUARD CHECK FAILED			
			grd_chk = 0	No check performed			
	Yes	LOGICAL BLOCK APPLICATION TAG LOGICAL BLOCK	АРР_СНК = 1 ^с	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
001b <u>101b</u> ^b			APP_CHK = 0	No check performed			
			ref_chk = 1 ⁱ	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
		REFERENCE TAG	ref_chk = 0	No check performed			
	No	Error condition	а				
		LOGICAL BLOCK GUARD	No check perform	ned			
		LOGICAL BLOCK	АРР_СНК = 1 ^с	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
010b ^b	Yes	APPLICATION TAG	APP_CHK = 0	No check performed			
		LOGICAL BLOCK	ref_chk = 1 ⁱ	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
		REFERENCE TAG	$REF_CHK = 0$	No check performed			
	No	Error condition	а				

 Table 16 — VRPROTECT field with BYTCHK set to zero - checking protection information read from the medium (part 2 of 4)

Code	Logical unit formatted with protection information	Field in protection information ^h	Extended INQUIRY Data VPD page bit value ^f	If check fails ^{de} , additional sense code	
	Yes	LOGICAL BLOCK GUARD	No check performed		
011b ^b		LOGICAL BLOCK APPLICATION TAG	No check performed		
		LOGICAL BLOCK REFERENCE TAG	No check perforn	ned	
	No	Error condition	а		

Table 16 — VRPROTECT field with BYTCHK set to zero - checking protection information read from the medium (part 3 of 4)

Code	Logical unit formatted with protection information	Field in protection information ^h	Extended INQUIRY Data VPD page bit value ^f	If check fails ^{de} , additional sense code		
		LOGICAL	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED		
		BLOCK GUARD	grd_chk = 0	No check performed		
100b ^b	Yes	LOGICAL BLOCK APPLICATION TAG	No check performed			
		LOGICAL BLOCK REFERENCE TAG	No check performed			
	No	Error condition	а			
1 <u>10</u> b - 111b	Reserved					

 Table 16 — VRPROTECT field with BYTCHK set to zero - checking protection information read from the medium (part 4 of 4)

		medium (part 4 of 4)								
Code	Logical unit formatted with protection information	Field in protection information ^h	Extended INQUIRY Data VPD page bit value ^f	If check fails ^{de} , additional sense code						
form sens b If the with sens c The LOGI set to LOGI d If an e If mu f See <u>9 If the</u> <u>a) 1</u> <u>b) 1</u> <u>b) 1</u> <u>i</u> i If the chect i If the com device the L G	atted with prote e key set to ILL e logical unit do CHECK COND e code set to II device server s CAL BLOCK APPL o one in the Co CAL BLOCK APPL erwise, this know error is reported litiple errors occ the Extended II CHK bit, and the error is reported litiple errors occ the Extended II CHK bit, and the error is reported litiple errors occ the Extended II CHK bit, and the error is reported litiple errors occ the Extended II CHK bit, and the error is reported litiple errors occ the Extended II CHK bit, and the error is reported litiple errors occ the Extended II CHK bit, and the error is reported litiple errors occ the Extended II CHK bit, and the error is reported to concal BLOCK of the checking erro_EN bit fiel mand is a VER command), the the LOGICAL BLOCK of OGICAL BLOCK of OGICAL BLOCK of OGICAL BLOCK of	ection information LEGAL REQUES es not support pro- VITION status with NVALID FIELD II shall check the lo LICATION TAG field introl mode page LICATION TAG field wiedge may be of ed, the sense key cur, the selection NQUIRY Data VI e REF_CHK bits. Ent or device server section information ent or device server section information ent or device server APPLICATION TAG TY (16) paramet APPLICATION TAG TY (16) paramet APPLICATION TAG is set to zero- HFY (10) comma check checks the with the logical to device server section check checks the server checks the REFERENCE TAG field is set to check set to server section the server checks the server checks the server checks the check checks the server checks the server checks the server checks the server checks the server checks the	n shall be terminate T and the addition rotection informatio h the sense key se N CDB. gical block applicat d. If the VERIFY (32 (see SPC-3), this l d and the LOGICAL B obtained by a methor y shall be set to AB n of which error to re PD page (see SPC wer detects a LOCICA n shall be disabled wer detects a: field set to FFFFh set to 010b, information in the a n the READ CAPAC nd, a VERIFY (12) e logical block refer block. If the RTO_EN hall check the logical c field in the CDB he logical block refer ield. This knowledg	ion information (see 4.16) and has not been ad with CHECK CONDITION status with the al sense code set to INVALID FIELD IN CDB. In the requested command should be terminated t to ILLEGAL REQUEST and the additional tion tag if it has knowledge of the contents of the 2) command (see 1.8) is used and the ATO bit is knowledge is acquired from the EXPECTED LOCK APPLICATION TAG MASK field in the CDB. od not defined by this standard. ORTED COMMAND. eport is not defined by this standard. -3) for the definitions of the GRD_CHK bit, the AL BLOCK APPLICATION TAG field set to FFFFh, the for the associated logical block. and the RTO_EN field is set to 000b or 001b in the _OT						

I

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 17.

Table 17 — 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 ^h	Extended INQUIRY Data VPD page bit value ^f	If check fails ^{d e} , additional sense code			
		LOGICAL BLOCK GUARD	grd_chk = 1	LOGICAL BLOCK GUARD CHECK FAILED			
		GOARD	GRD_CHK = 0	No check performed			
000b	Yes	LOGICAL BLOCK	АРР_СНК = 1 ^{с g}	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
		AFFEICATION TAG	APP_CHK = 0	No check performed			
		LOGICAL BLOCK	REF_CHK = 1 ⁱ	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
		REFERENCE TAG	REF_CHK = 0	No check performed			
	No	No protection inform	No protection information on the medium available to check				
001b		LOGICAL BLOCK GUARD	No check perform	ned			
010b 011b 100b	Yes	LOGICAL BLOCK APPLICATION TAG	No check performed				
<u>101b</u> b		LOGICAL BLOCK REFERENCE TAG	No check perforr	ned			
	No	Error condition ^a					

Table 17 — VRPROTECT field with BYTCHK set to one - checking protection information read from the medium (part 2 of 2)

				· ·			
Code	Logical unit formattedField in protection informationExtended INQUIRY Data VPD page bit value f		If check fails ^{d e} , additional sense code				
110b - 111b Reserved							
formatted with protection information Field in protection information Extended INQUIRY Data VPD page bit value f If check fails ^{d e} , additional sense code 110b - Reserved Reserved							

I

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 18.

Table 18 — 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 ^{d e} , additional sense code		
000b	Yes	No protection info	ormation received	d from application client to check		
0000	No	No protection info	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 ^f	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
	No	Error condition ^a	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				
		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 18 — 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	<u>Shall</u>	LOGICAL BLOCK GUARD CHECK FAILED					
<u>101b ^b</u>	Yes	LOGICAL BLOCK APPLICATION TAG	<u>May ^c</u>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED					
		LOGICAL BLOCK REFERENCE TAG	<u>May ^f</u>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED					
	No	Error condition ^a							
1 <u>10</u> b - 111b	Reserved								
 ^a 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. ^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 and the LOGICAL BLOCK APPLICATION TAG field and the LOGICAL BLOCK APPLICATION TAG field and the LOGICAL BLOCK APPLICATION TAG MASK field in the CDB. ^c Otherwise, 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 the RTO_EN bit field is set to zero in the READ CAPACITY (16) parameter data (see 1.6.2)(i.e., the command is a VERIFY (10) command, a VERIFY (12) command, or a VERIFY (16) command), the device server shall 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 Zero one the device server checks the logical block reference tag based on the EXPECTED INITAL LOGICAL BLOCK REFERENCE TAG field in the CDB (see 4.16.2). If the RTO_EN bit field is not set to zero one the device server checks the logical block reference tag if it has knowledge of the contents of the LOGICAL BLOCK REFERENCE TAG field in a VERIFY (32) command (see 1.8) or by a method not defined by this standard. 									

If the BYTCHK bit is set to one, the device server shall perform a byte-by-byte comparison of protection information transferred from the data-out buffer with protection information read from the medium based on the VRPROTECT field as described in table 19.

Code	Logical unit formatted with protection information	Field	Byte-by-byte Comparison	If compare fails ^{c d} , additional sense code				
000b	Yes	No protection informati data is compared withi		n application client to compare. Only user lock.				
0005	No	No protection information or the medium or received from application client to compare. Only user data is compared within each logical block.						
	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED				
001b		LOGICAL BLOCK APPLICATION TAG (ATO = 1) ^e	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED				
011b 100b ^b		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 ^a						

Table 19 — 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 not	No compare performed		
		LOGICAL BLOCK APPLICATION TAG (ATO = 1) ^e	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
010b ^b	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 ^a				
	<u>Yes</u>	LOGICAL BLOCK GUARD	<u>Shall</u>	LOGICAL BLOCK GUARD CHECK FAILED		
		LOGICAL BLOCK APPLICATION TAG (ATO = 1) [©]	<u>Shall</u>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
<u>101b ^b</u>		$\frac{\text{LOGICAL BLOCK}}{\text{APPLICATION TAG}}$ $\frac{\text{(ATO = 0)}^{\text{f}}}{\text{(ATO = 0)}^{\text{f}}}$	<u>Shall not</u>	No compare performed		
		LOGICAL BLOCK REFERENCE TAG	Shall not	No compare performed		
	<u>No</u>	Error condition a				
1 <u>10</u> b - 111b	Reserved					
 ^a 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. ^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 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. 						

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

1.8 VERIFY (32) command

The VERIFY (32) command (see table 20) requests that the device server verify the specified logical block(s) on the medium. Each logical block includes user data and may include protection information, based on the VRPROTECT field and the medium format.

If the RTO_EN bit field is set to zero in the READ CAPACITY (16) parameter data (see 1.6.2), the device servershall terminate this command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST- and the additional sense code set to INVALID COMMAND OPERATION CODE. <u>If the_RTO_EN bit field is not</u> set to zero one, the device server may process the command (see 1.1.3).

Byte\Bit	7	6	5	4	3	2	1	0
0				OPERATIO	N CODE (7Fh)		
1		CONTROL						
2		Reserved						
5		_						
6		Reserved GROUP NUMBER						
7			A	DDITIONAL C	DB LENGTH (1	8h)		
8	(MSB)				TION (000Ah	<u>۱</u>		
9		_		SERVICE AC)		(LSB)
10		VRPROTECT DPO Reserved BYTCHK					Reserved	
11				Reserved				XLB INVALID
12	(MSB)				OCK ADDRES	2		
19		_		LUGICAL BL	OCK ADDRES	5		(LSB)
20	(MSB)		EXDECTED		AL BLOCK RE		`	
23		_	EXPECTED	INTIAL LOGIC	AL BLOCK RE	FERENCE TA	3	(LSB)
24	(MSB)		EVDECT					
25		_	– EXPECTED LOGICAL BLOCK APPLICATION TAG (LS					
26	(MSB)							
27		– LOGICAL BLOCK APPLICATION TAG MASK (LS						(LSB)
28	(MSB)				ION LENGTH			
31		_		VERIFICAT				(LSB)

Table 20 — VERIFY (32) command

See the VERIFY (10) command (see 1.7) for the definitions of the GROUP NUMBER field, VRPROTECT field, DPO bit, BYTCHK bit, LOGICAL BLOCK ADDRESS field, and VERIFICATION LENGTH field.

An expected logical block invalid (XLB_INVALID) bit set to zero specifies that the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field, EXPECTED LOGICAL BLOCK APPLICATION TAG field, and LOGICAL BLOCK APPLICATION TAG MASK field are valid. An XLB_INVALID bit set to one specifies that the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field shall not be used when checking of the LOGICAL BLOCK REFERENCE TAG field is enabled (see table 16, table 17, table 18, and table 19 in 1.7). In this case the value expected in the LOGICAL BLOCK REFERENCE TAG field is not defined by this standard.

When checking of the LOGICAL BLOCK REFERENCE TAG field is enabled (see table 16, table 17, table 18, and table 19 in 1.7) and the XLB_INVALID bit is set to zero, 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 ATO bit is set to one in the Control mode page (see SPC-3), the XLB INVALID bit is set to zero, and checking of the LOGICAL BLOCK APPLICATION TAG field is enabled (see table 16, table 17, table 18, and table 19 in 1.7), 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.

The LOGICAL BLOCK APPLICATION TAG MASK field and the EXPECTED LOGICAL BLOCK APPLICATION TAG field shall be ignored if:

- a) the XLB_INVALID bit is set to one;
- b) the ATO bit is set to zero; or
- c) 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 16, table 17, table 18, and table 19 in 1.7).

1.9 WRITE (6) command

The WRITE (6) command (see table 21) 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	LOGICAL BLOCK ADDRESS (LSB)							(LSB)
4		TRANSFER LENGTH						
5	CONTROL							

Table 21 — 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 the RTO_EN bit field is set to zero in the READ CAPACITY (16) parameter data (see 1.6.2); or
 - B) FFFFFFFh, if the RTO_EN bit field is not set to zero one;

and

I

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).

1.10 WRITE (10) command

•••

I

I

I

The device server shall check the protection information transferred from the data-out buffer based on the WRPROTECT field as described in table 22.

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 information received from application client to check					
0000	No	No protection information received from application client to check					
001b ^{bj}	Yes ^e	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED			
		LOGICAL BLOCK APPLICATION TAG	May ^c	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
		LOGICAL BLOCK REFERENCE TAG	Shall ^k	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
	No ^a	No protection information available to check					
010b ^{b_j}		LOGICAL BLOCK GUARD	Shall not	No check performed			
	Yes ^e	LOGICAL BLOCK APPLICATION TAG	May ^c	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
		LOGICAL BLOCK REFERENCE TAG	May ^k	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					

Code	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails ^{d i} , additional sense code		
100b ^{b_j}	Yes ^e	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED		
		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				
ن ^{ز ع} <u>101b</u>	<u>Yes</u> ª	LOGICAL BLOCK GUARD	<u>Shall</u>	LOGICAL BLOCK GUARD CHECK FAILED		
		LOGICAL BLOCK APPLICATION TAG	<u>May ^c</u>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
		LOGICAL BLOCK REFERENCE TAG	<u>May ^k</u>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
	<u>No ^a</u>	No protection information available to check				

I

I

Table 22 — WRPROTECT field (part 2 of 3)

Table 22 — WRPROTECT field (part 3 of 3)
--

Code	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails ^{d i} , additional sense code					
1 <u>10</u> b - 111b									
 form sense lf the with sense The mod field. LOGI Othe field. If an e Devi non-f The field. g If the serv field shall logic h If the into any i If mu i If the code lift the serv with devi k If the serv with devi REFE 	atted with prote e key set to ILL e logical unit doe CHECK COND e 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 memory device server shall volatile memory device server shall er shall write the of each of the v write a value of al blocks. e ATO bit is set to each LOGICAL BL value into each litiple errors occ RTO_EN bit is set ce server. If the TE (12) comma CK CONDITION e set to INVALIE e RTO_EN bit field er checks the lo the logical block ce server check rence TAC field is the logical block ce server check rence TAC field is the logical block	ction information s EGAL REQUEST es not support prote ITION status with t VALID FIELD IN C hay check the logic C-3) and if it has k 32) command (see ICATION TAG field at wledge is obtained d, the sense key sl preserve the conte d). hall write a properly d is set to zero in the e least significant for written logical block f FFFFFFFh into c one in the Contro LOCK APPLICATION T LOGICAL BLOCK APP cur, the selection of the to zero in the RE mmand. If the RTO- COMMAND OPE d is set to zero in the RTO_EN bit is set to c COMMAND OPE d is set to zero in the sthe logical block f in the CDB (see 4 lock reference tag d. This knowledge r	hall be termi and the addi ection inform he sense ke CDB. al block app nowledge of 1.11) is use nd the LOGIC by a method hall be set to the LOGICAL of generated ne READ CA our bytes of the LOGICAL of mode page rag field. If the LICATION TAG f which error EAD CAPAC EN bit is set ls with the w oon, the do the comman ense key set RATION CO he READ CA ce tag by cor is set to one reference ta the court is set to one reference ta the court and the court court by tes of the comman ense key set reference ta the court is set to one reference ta the court and the court court by tes of the court by tes of tes of tes o	to report is not defined by this standard. ITY (16) parameter data (see 1.6), the device server to one, WRITE (10) commands, WRITE (12) RPROTECT field set to 000b may be processed by the evice server shall terminate WRITE (10) commands, ds with the WRPROTECT field not set to 000b with- to ILLEGAL REQUEST and the additional sense					

1.11 WRITE (32) command

The WRITE (32) command (see table 23) 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 and may include protection information, based on the WRPROTECT field and the medium format. Each logical block

I

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

If the RTO_EN bit field is set to zero in the READ CAPACITY (16) parameter data (see 1.6.2), the device servershall terminate the WRITE (32) command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID COMMAND OPERATION CODE. If the RTO_EN bitfield is not set to zero one, the device server may process the command (see 1.1.3).

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Paganyad							
5	Reserved							
6		Reserved GROUP NUMBER						
7	ADDITIONAL CDB LENGTH (18h)							
8	(MSB)	SERVICE ACTION (000Bb)						
9		_	- SERVICE ACTION (000Bh)					
10		WRPROTEC	Т	DPO	FUA	Reserved	FUA_NV	Reserved
11		Reserved						XLB INVALID
12	(MSB)	– LOGICAL BLOCK ADDRESS						·
19								(LSB)
20	(MSB)	- EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG						
23		_	(LSB)					
24	(MSB)	EXPECTED LOGICAL BLOCK APPLICATION TAG						
25								(LSB)
26	(MSB)	- LOGICAL BLOCK APPLICATION TAG MASK						
27								(LSB)
28	(MSB)							
31								(LSB)

Table 23 — WRITE (32) command

See the WRITE (10) command (see 1.10) 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.

An expected logical block invalid (XLB_INVALID) bit set to zero specifies that the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field, EXPECTED LOGICAL BLOCK APPLICATION TAG field, and LOGICAL BLOCK APPLICATION TAG MASK field are valid. An XLB_INVALID bit set to one specifies that the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field shall not be used when checking of the LOGICAL BLOCK REFERENCE TAG field is enabled (see table 22 in 1.10). In this case the value expected in the LOGICAL BLOCK REFERENCE TAG field is not defined by this standard.

When checking of the LOGICAL BLOCK REFERENCE TAG field is enabled (see table 22 in 1.10) and the <u>XLB_INVALID bit is set to zero</u>, 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 ATO bit is set to one in the Control mode page (see SPC-3), <u>the XLB_INVALID bit is set to zero</u>, and checking of the LOGICAL BLOCK APPLICATION TAG field is enabled (see table 22 in 1.10), 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.

The LOGICAL BLOCK APPLICATION TAG MASK field and the EXPECTED LOGICAL BLOCK APPLICATION TAG field shall be ignored if:

- a) the XLB_INVALID bit is set to one;
- b) the ATO bit is set to zero; or
- c) 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 22 in 1.10).