#### T10/05-156 revision 4

Date: August 03, 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	PERIPHERAL QUALIFIER PERIPHERAL DEVICE TYPE							
1				PAGE CODE (8	36h)			
2		Reserved						
3		PAGE LENGTH (3Ch)						
4	Rese	erved		<u>RTO</u>		GRD_CHK	APP_CHK	REF_CHK
5		Reserved			PRIOR_SUP	HEADSUP	ORDSUP	SIMPSUP
6		Reserved NV_SUP V_SUP						V_SUP
7				Deserved				
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>	<u>Definition</u>
<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 field</u>

### Changes to SBC-3

#### **1.0.2 CDB processing restrictions**

If the logical unit is formatted with protection information not all CDBs with non-zero values in the <u>RDPROTECT</u> field, <u>WRPROTECT</u> field, and <u>VRPROTECT</u> field are valid. CDBs that are allowed is based on the <u>RTO</u> EN field returned in the <u>READ</u> CAPACITY (16) parameter data (see 1.5.2) as specified in table 3.

#### Table 3 — Protection information CDB processing restrictions

RTO EN field a	Description
<u>000b</u>	Any CDB that contains an EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field 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>	<ul> <li>Any CDB that does not contain:         <ul> <li>a) an EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field; or</li> <li>b) a RDPROTECT field set to zero, WRPROTECT field set to zero, or VRPROTECT field set to zero,</li> </ul> </li> <li>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.</li> <li>Any CDB that 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 ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.</li> </ul>
<u>010b</u>	No restrictions placed on any CDBs as a result of a logical unit being formatted with protection information.
<u>011b - 111b</u>	Reserved
<sup>a</sup> Specified in the	e READ CAPACITY (16) parameter data (see 1.5.2).

#### 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			USER DATA							
n - 1										
n	(MSB)		LOGICAL BLOCK GUARD (LSB)							
n + 1										
n + 2	(MSB)		LOGICAL BLOCK APPLICATION TAG (LSB)							
n + 3										
n + 4	(MSB)									
n + 7			LOGICAL BLOCK REFERENCE 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 1: 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. 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 BLOCK REFERENCE TAG fin	eld of the first logical block in the data-in buffer
and/or data-ou	ut 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>	CDBs that contain an EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field with the XLB_INVALID bit is set to zero shall set the LOGICAL BLOCK REFERENCE TAG field of the first logical block in the data-in buffer and/or data-out buffer to the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field of the command.
<u>010b</u>	For any CDB that:         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.         CDBs that contain an EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field with the         XLB       INVALID bit is set to zero shall set the LOGICAL BLOCK REFERENCE TAG field of the first         logical block in the data-in buffer and/or data-out buffer to 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 1.5.2).

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 logical 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 CDBs that contain 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 CDB that:         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 CDBs that contain 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 1.5.2).

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.

## **1.1 FORMAT UNIT command**

### 1.1.1 FORMAT UNIT command overview

•••

A format protection information (FMTPINFO) bit 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 block descriptor of the mode parameter header (see SPC-3). A FMTPINFO bit set to one specifies that the device server shall enable the use of protection information (see 4.16) and format the medium to the block length specified in the medium to the block length specified in the mode parameter block descriptor of 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 1.5.2) indicates whether protection information (see 4.16) is enabled.

The reference tag own request (RTO\_REQ) bit 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, and the RTO field in the Extended INQUIRY Data VPD page (see x.x.x). is not set to 000b, 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 ownership) of the LOGICAL BLOCK REFERENCE TAG field. If the:

- a) FMTPINFO bit is set to zero and the RTO\_REQ bit is set to one; or
- b) FMTPINFO bit is set to one, the RTO\_REQ bit is set to one, and the RTO field in the Extended INQUIRY Data VPD page (see x.x.x). is set to 000b,

L

then 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.

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.

••••

#### 1.1.1.1 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	PROTECTION 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	PROTECTION FIELDS USAGE					
1	FOV							Vendor specific	
2		Reserved							
3		Reserved							
4	(MSB)	(MSB)							
7	DEFECT LIST LENGTH (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). The PROTECTION FIELD USAGE field shall be ignored unless the FMTPINFO bit is set to one, RTO\_REQ bit is set to one, and the RTO field in the Extended INQUIRY Data VPD page (see x.x.x). is not set to 000b.

<u>Code</u>	Description
<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 1.0.3).
<u>001b</u>	The logical unit shall be formatted to allow 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 1.0.3). If the RTO field in the Extended INQUIRY Data VPD page (see SPC-4). is not set to 011b, then 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.
<u>010b - 111b</u>	Reserved

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.

## 1.2 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 serverField in protection information?		Extended INQUIRY Data VPD page bit value <sup>d</sup>	lf check fails <sup>bc</sup> , 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 <sup>а</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED					
	No		APP_CHK = $0$	No check performed					
		LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 <sup>g</sup>	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 f the device protection in If the device a) LOGICAL READ ( b) LOGICAL FEFFh, then the de g If the RTO_E server chec with the log	No         No protection information available to check           a         The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. The method for acquiring this knowledge is not defined by this standard.           b         If an error is reported, the sense key shall be set to ABORTED COMMAND.           c         If multiple errors occur, the selection of which error to report is not defined by this standard.           d         See the Extended INQUIRY Data VPD page (see SPC-3) for the definitions of the GRD_CHK bit, APP_CHK bit, and REF_CHK bit.           e         If the device server detects a LOCICAL BLOCK APPLICATION TAC 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 1.5.2); or           b)         LOGICAL BLOCK APPLICATION TAG field set to FFFFh, LOGICAL BLOCK REFERENCE TAG field set to FFFFF           e         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.								

Table 10 —	- Protection	information	checking for READ	(6)
------------	--------------	-------------	-------------------	-----

## 1.3 READ (10) command

I

I

I

I

#### T10/05-156 revision 4

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 <sup>i</sup>	Extended INQUIRY Data VPD page bit value <sup>g</sup>	lf check fails <sup>df</sup> , 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 <sup>с</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
<del>أ</del> _000b		No	TAG	APP_CHK = 0	No check performed		
			LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 <sup>k</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 GUARD	grd_chk = 1	LOGICAL BLOCK GUARD CHECK FAILED		
			GUARD	grd_chk = 0	No check performed		
	Yes		LOGICAL BLOCK APPLICATION TAG	арр_снк = 1 <sup>с</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
001b <u>101b</u> <sup>b-j</sup>			IAG	АРР_СНК = 0	No check performed		
			LOGICAL BLOCK REFERENCE	REF_CHK = 1 <sup>k</sup>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
			TAG	ref_chk = 0	No check performed		
	No <sup>a</sup>	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 <sup>i</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		
	Yes	Yes <sup>e</sup>	LOGICAL BLOCK APPLICATION TAG	арр_снк = 1 <sup>с</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
010b <sup>b-j</sup>			TAG	APP_CHK = $0$	No check performed		
			LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 <sup>k</sup>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
			TAG	$REF_CHK = 0$	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-j</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 information available to transmit to the data-in buffered checking					

I

I

Table 11 — RDPROTECT field (part 2 of 4)

I

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

## Table 11 — RDPROTECT field (part 3 of 4)

Table 11 — RDPROTECT field (part 4 of 4)

				<b>a</b> (part 4 of 4)	
Code	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information <sup>i</sup>	Extended INQUIRY Data VPD page bit value <sup>g</sup>	If check fails <sup>df</sup> , additional sense code
b If the sense If the termin additi C The C LOGIC set to LOGIC Other If an C e Trans f If mul g See t h APP_C h If the i If the i If the i If the g L b) L f then f f f the f f f f f f f f f f f f f f f f f f f	atted with protect e key set to ILLE logical unit doe nated with CHE ional sense cod device server sh CAL BLOCK APPLIC one in the Con CAL BLOCK APPLIC rwise, this know error is reported smit protection in litiple errors occu he Extended IN CHK bit, and the application clien oGICAL BLOCK APPLIC the Checking of all pr application clien oGICAL BLOCK APPLIC FFF FFFFh, an the checking of RTO_EN bit is se process the com nands, and REA of (12) command CK CONDITION set to INVALID RTO_EN bit field er checks the log cliated with the log	tion information s GAL 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 acq l, the sense key sl normation to the o QUIRY Data VPD REF_CHK bit. To device serve otection information to device serve otection information to device serve opelication TAG field all protection infor to zero in the RE opelication TAG field all protection infor to zero in the RE opelication to the set to zero in the RE opelication the set to zero in the RE opelication to the set to zero in the set to zero i	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 k and the LOGICAL BL juired by a method hall be set to ABO data-in buffer. f which error to re- page (see SPC- on in the associated r detects a: end set to FFFFh a ever data (see 1.5 end set to FFFFh, d is set to 010b, rmation in the associated r detects a: end set to FFFFh, d is set to 010b, rmation in the associated swith the RDPROT o one, the device o one, the device o commands with ense key set to HE RATION CODE. The READ CAPACITY is commands with ense key set to HE RATION CODE. The READ CAPACITY is contents of the in expected in the associated of the respected in the associated of the respected in the associated of the respected of the	d with CHECK CO I sense code set to n the requested co nse key set to ILLE on tag if it has know command (see 1.4) nowledge is acquir OCK APPLICATION T d not defined by th DRTED COMMANI eport is not defined 3) for the definition AL BLOCK APPLICATION and the RTO_EN fiel and the RTO_EN fiel and the RTO_EN fiel acciated logical block sh and the RTO_EN fiel acciated logical block sh acciated logical block sh a	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 1.5), 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. This REFERENCE TAG field in a

## 1.4 READ (32) command

I

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 1.5.2), the device servershall terminate the 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 bit field is not set to zero one, the device server may process the command (see 1.0.3).

Byte\Bit	7	6	5	4	3	2	1	0	
0		OPERATION CODE (7Fh)							
1				CO	NTROL				
2				Por	erved				
5		_		Nea	erveu				
6		Reserved			(	GROUP NUMB	ER		
7			A	DDITIONAL C	DB LENGTH (1	l8h)			
8	(MSB)				TION (0009h	.)			
9				SERVICE AC		')		(LSB)	
10		RDPROTEC	Г	DPO	FUA	Reserved	FUA_NV	Reserved	
11				Reserved				XLB INVALID	
12	(MSB)				OCK ADDRES	9			
19				LOOIOAL DE		0		(LSB)	
20	(MSB)		EXPECTED			FERENCE TAG	3		
23								(LSB)	
24	(MSB)		EXPECT	ED LOGICAL E					
25			EXILON					(LSB)	
26	(MSB)					AG MASK			
27			– LOGICAL BLOCK APPLICATION TAG MASK (LSB)						
28	(MSB)			TRANSFI	ER LENGTH				
31								(LSB)	

#### Table 12 — READ (32) command

See the READ (10) command (see 1.3) 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 1.3). 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 1.3) 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 1.3), 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 1.3).

### 1.5 READ CAPACITY (16) command

#### 1.5.1 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)		LOGICAL BLOCK ADDRESS (LSB)						
9		-							
10	(MSB)				N LENGTH				
13		-		ALLOCATIC				(LSB)	
14			Reserved PMI					PMI	
15				CO	NTROL				

#### 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.

#### 1.5.2 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)		RETURNED LOGICAL BLOCK ADDRESS (LSB)							
7										
8	(MSB)		BLOCK LENGTH IN BYTES (LSB)							
11										
12		Res	Reserved RTO_EN PROT_E							
13			Reserved							
31				11050						

#### 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\_FFFFF.

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 units 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 1.0.3).
<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.

## 1.6 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 <sup>h</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	ref_chk = 1 <sup>i</sup>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED		
		REFERENCE TAG	$REF_CHK = 0$	No check performed		
	No	No protection information on the 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 APPLICATION TAG LOGICAL BLOCK	арр_снк = 1 <sup>с</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
001b <u>101b</u> <sup>b</sup>	Yes		APP_CHK = 0	No check performed		
			ref_chk = 1 <sup>i</sup>	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 <sup>с</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
010b <sup>b</sup>	Yes	APPLICATION TAG	APP_CHK = 0	No check performed		
		LOGICAL BLOCK	ref_chk = 1 <sup>i</sup>	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 <sup>h</sup>	Extended INQUIRY Data VPD page bit value <sup>f</sup>	If check fails <sup>de</sup> , additional sense code			
	Yes	LOGICAL BLOCK GUARD	No check perform	ned			
011b <sup>b</sup>		LOGICAL BLOCK APPLICATION TAG	No check perforn	ned			
		LOGICAL BLOCK REFERENCE TAG	No check perforn	ned			
	No	Error condition	<u>ו</u> ו <sup>a</sup>				

 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 <sup>h</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		
	Yes	BLOCK GUARD	grd_chk = 0	No check performed		
100b <sup>b</sup>		LOGICAL BLOCK APPLICATION TAG	No check performed			
		LOGICAL BLOCK REFERENCE TAG	No check performed			
	No	Error condition a				
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)

			incolum (part 4	,			
Code	Logical unit formatted       Field in protection information       Extended INQUIRY Data VPD page bit value f       If check fails definition						
form sens b If the with sens c The LOGI Set to LOGI d If an e If mu f See g If the chece h If the b) I g the i If the comm device the L Gala	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 come the checking erro_EN bit fiel mand is a VER command), the the CHC BLOCK F	ection information LEGAL REQUES es not support pro- DITION status with NVALID FIELD II shall check the lo LICATION TAG field whedge may be of entrol mode page LICATION TAG field whedge may be of ed, the sense key cur, the selection NQUIRY Data VI e REF_CHK bits. Ent or device server ection information ent or device server ection information ent or device server APPLICATION TAG TY (16) paramet APPLICATION TAG TY (16) paramet APPLICATION TAG e RTO_EN field is of all protection id is set to zero- HFY (10) comma check checks the with the logical B device server server server checks the efference TAG	n shall be terminate ST and the addition rotection informatio th the sense key set N CDB. gical block application d. If the VERIFY (32 (see SPC-3), this d and the LOGICAL E obtained by a methor y shall be set to AB n of which error to r PD page (see SPC wer detects a LOGIC 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 logic of field in the CDB- he logical block refer ield. This knowledg	ion information (see 4.16) and has not been ed with CHECK CONDITION status with the al sense code set to INVALID FIELD IN CDB. In the requested command should be terminated at to ILLEGAL REQUEST and the additional tion tag if it has knowledge of the contents of the 2) command (see 1.7) is used and the ATO bit is knowledge is acquired from the EXPECTED BLOCK 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 LOG LOGICAL BLOCK REFERENCE TAG field set to FFFF associated logical block shall be disabled. CITY (16) parameter data (see 1.5) (i.e., the- command, or a VERIFY (16) command), the rence tag by comparing it to the lower 4 bytes of Lbit is set to one (i.e., the command is a VERIFY- al block reference tag based on the EXPECTED (see 4.16.2). If the RTO_EN bit field is not set to erence tag if it has knowledge of the contents of the may be acquired through the EXPECTED INITIAL command (see 1.7) or by a method not defined by			

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 <sup>h</sup>	Extended INQUIRY Data VPD page bit value <sup>f</sup>	If check fails <sup>d e</sup> , additional sense code			
		LOGICAL BLOCK GUARD	grd_chk = 1	LOGICAL BLOCK GUARD CHECK FAILED			
		GUARD	GRD_CHK = 0	No check performed			
0.001	Yes	LOGICAL BLOCK	APP_CHK = 1 <sup>c</sup> <sup>g</sup>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED			
000b		AFFEICATION TAG	APP_CHK = 0	No check performed			
		LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 <sup>i</sup>	LOGICAL BLOCK REFERENCE TAG CHECK FAILED			
			REF_CHK = 0	No check performed			
	No	No protection inform	nation on the medi	um available to check			
001b		LOGICAL BLOCK GUARD	No check perforr	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 <sup>a</sup>					

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

	Logical unit		Extended	
Code	formatted with protection information	Field in protection information <sup>h</sup>	INQUIRY Data VPD page bit value <sup>f</sup>	If check fails <sup>d e</sup> , additional sense code
1 <u>10</u> b - 111b	Reserved			
i If ti sei vit sei c Th LOO ofti d If a e If n f Se e If n f Se <u>e If n</u> <u>f Se</u> <u>i ffti</u> <u>a)</u> <u>b)</u> <u>i ffti</u> con de the de the con de the con de the con de the con de the con de the con de the con de de the con de con de the con de the con de the con de the con de the con de the con de t con de con de con de t t t t t t t f t t t t t t t t t t t	matted with prot hase key set to IL he logical unit do h CHECK CONI hase code set to I e device server a GICAL BLOCK APP to one in the Co GICAL BLOCK APP herwise, this known in error is report nultiple errors or e the Extended P_CHK bit, and the he application cli color call BLOCK <u>READ CAPAC</u> LOGICAL BLOCK <u>READ CAPAC</u> <u>LOGICAL BLOCK</u> <u>FFFFh, and the</u> he RTO_EN <del>bit</del> fic- mmand is a VEF vice server shall a LBA associated CO one the device a LOGICAL BLOCK	ection information sh LEGAL REQUEST a bes not support prote- DITION status with th INVALID FIELD IN Cl shall check the logical plication TAG field. If ontrol mode page (se plication TAG field an owledge may be obta ed, the sense key sh ccur, the selection of INQUIRY Data VPD the REF_CHK bit. ient or device server APPLICATION TAG field ITY (16) parameter d APPLICATION TAG field is set to zero in th RIFY (10) command, check checks the log d with the logical bloc e device server shall GK REFERENCE TAG field.	all be terminated v ind the additional s ction information the sense key set to DB. al block application the VERIFY (32) c es SPC-3), this kno d the LOGICAL BLOC ined by a method r all be set to ABOR which error to report page (see SPC-3) detects a LOGICAL E nall be disabled for detects a: d set to FFFFh and ata (see 1.5.2); or d set to FFFFh, LOC to 010b, irmation in the asso e READ CAPACIT a VERIFY (12) cor gical block reference k. If the RTO_EN bit check the logical to eld in the CDB (see ogical block reference). This knowledge m	information (see 4.16) and has not been with CHECK CONDITION status with the ense 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 ommand (see 1.7) is used and the ATO bit is wledge is acquired from the EXPECTED CK APPLICATION TAG MASK field in the CDB. hot defined by this standard. TED COMMAND. ort is not defined by this standard. for the definitions of the GRD_CHK bit, the BLOCK APPLICATION TAC field set to FFFFh, the the associated logical block. the RTO_EN field is set to 000b or 001b in the BICAL BLOCK REFERENCE TAG field set to FFFF Deciated logical block shall be disabled. Y (16) parameter data (see 1.5) (i.e., the nmand, or a VERIFY (16) command), the ce tag by comparing it to the lower 4 bytes of is set to one (i.e., the command is a VERIFY block reference tag based on the EXPECTED e 4.16.2). If the RTO_EN bit field is not set to nce tag if it has knowledge of the contents of hay be acquired through the EXPECTED INITIAL mand (see 1.7) or by a method not defined by

I

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 <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>			
	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>			
		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>			
	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED	
100b <sup>b</sup>		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 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 <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			
	No	Error condition <sup>a</sup>					
1 <u>10</u> b - 111b	Reserved						
<ul> <li>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>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>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 VERIFY (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.</li> <li>Otherwise, this knowledge is obtained by a method not defined by this standard.</li> <li>If an error is reported, the sense key shall be set to ABORTED COMMAND.</li> <li>If multiple errors occur, the selection of which error to report is not defined by this standard.</li> <li>If the RTO_EN bit field is set to zero in the READ CAPACITY (16) parameter data (see 1.5.2)(<del>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 one (i.e., the command is a VERIFY (32) command, a VERIFY (12) command, or a VERIFY (16) command, is a VERIFY (32) command, is a VERIFY (52) command, is a VE</del></li></ul>							

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 <sup>c d</sup> , additional sense code				
000b	Yes		No protection information received from application client to compare. Only user data is compared within each logical block.					
0000	No	No protection information or the medium or received from application client to compare. Only user data is compared within each logical block.						
	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED				
001b		LOGICAL BLOCK APPLICATION TAG (ATO = 1) <sup>e</sup>	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED				
011b 100b <sup>b</sup>		LOGICAL BLOCK APPLICATION TAG (ATO = 0) <sup>f</sup>		No compare performed				
		LOGICAL BLOCK REFERENCE TAG	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED				
	No	Error condition <sup>a</sup>						

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 <sup>c d</sup> , additional sense code		
		LOGICAL BLOCK GUARD	Shall not	No compare performed		
	Yes	LOGICAL BLOCK APPLICATION TAG (ATO = 1) <sup>e</sup>	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
010b <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>				
	<u>Yes</u>	LOGICAL BLOCK GUARD	<u>Shall</u>	LOGICAL BLOCK GUARD CHECK FAILED		
		LOGICAL BLOCK APPLICATION TAG (ATO = 1) <sup>©</sup>	<u>Shall</u>	LOGICAL BLOCK APPLICATION TAG CHECK FAILED		
<u>101b <sup>b</sup></u>		$\frac{\text{LOGICAL BLOCK}}{\text{APPLICATION TAG}}$ $\frac{\text{(ATO = 0)}^{\text{f}}}{\text{(ATO = 0)}^{\text{f}}}$	Shall not	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					
<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 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> </ul>						

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

## 1.7 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 1.5.2), 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. If the RTO\_EN bit field is not set to zero one, the device server may process the command (see 1.0.3).

Byte\Bit	7	6	5	4	3	2	1	0	
0			•	OPERATIO	N CODE (7Fh	)	-		
1		CONTROL							
2		Reserved							
5									
6		Reserved			C	GROUP NUMB	ER		
7			A	DDITIONAL C	ob length (1	8h)			
8	(MSB)				тіоn (000Ah	)			
9		_		OLIVIOL AO		)		(LSB)	
10		VRPROTEC	Т	DPO	Rese	erved	ВҮТСНК	Reserved	
11				Reserved				XLB INVALID	
12	(MSB)				OCK ADDRES	9			
19		_		LUGICAL BL	JOR ADDRES	5		(LSB)	
20	(MSB)		EXDECTED	INITIAL LOGIC			2		
23		_	EXFLUTED		AL BLOCK KE	FERENCE TA	5	(LSB)	
24	(MSB)		EVPECT	ED LOGICAL B					
25		_	EXFECT					(LSB)	
26	(MSB)					G MASK			
27		– LOGICAL BLOCK APPLICATION TAG MASK (LSB						(LSB)	
28	(MSB)				ION LENGTH				
31		_		VERIFICAT				(LSB)	

#### Table 20 — VERIFY (32) command

See the VERIFY (10) command (see 1.6) 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 1.6). 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 1.6) 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 16, table 17, table 18, and table 19 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

#### T10/05-156 revision 4

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 1.6).

## 1.8 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)						
4		TRANSFER LENGTH						
5	CONTROL							

Table 21 — W	RITE (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 1.5.2); or
  - B) FFFFFFFh, if the RTO\_EN bit field is not set to zero one;

and

c) the LOGICAL BLOCK APPLICATION TAG field set to:

I

- 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).

## 1.9 WRITE (10) command

...

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 <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	No protection information received 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						
	Yes <sup>e</sup>	LOGICAL BLOCK GUARD	Shall not	No check performed				
011b <sup>b_j</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	ailable to check				

Table 22 —	- WRPROTECT	field	(part 1	of 3)
------------	-------------	-------	---------	-------

I

I

Code	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails <sup>d i</sup> , additional sense code		
	Yes <sup>e</sup>	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED		
ز <sup>ل</sup> 100b		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed		
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed		
	No <sup>a</sup>	No protection information available to check				
	<u>Yes</u> ª	LOGICAL BLOCK GUARD	<u>Shall</u>	LOGICAL BLOCK GUARD CHECK FAILED		
<del>ز م <u>101b</u></del>		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 <sup>a</sup></u>	No protection information available to check				

## 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 <sup>d i</sup> , additional sense code
1 <u>10</u> b - 111b	Reserved			
form sens b If the with sens c The modu field. LOGIO Othe d If an e Devi field. g If the serve field shall logic h If the into o any v i If mu <del>j</del> field. k REFE chece REFE	atted with prote e key set to ILL logical unit doe CHECK COND e code set to IN device server me e page (see SP If the WRITE ( CAL BLOCK APPL rwise, this know error is reporte ce server shall volatile memory device server shall er shall write the of each of the v write a value o al blocks. ATO bit is set to each LOGICAL BI value into each litiple errors occ RTO_EN bit is set process the com mands, and WR ce server. If the TE (12) comma CK CONDITION set to INVALIE er checks the lo the logical block ce server check RENCE TAC field ks the logical block RENCE TAG field	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 a wledge is obtained d, the sense key sl preserve the conte location TAG field a wledge is obtained d, the sense key sl preserve the conte location the contro Location the Contro	hall be termi and the addi ection inform he sense ke CDB. al block app nowledge of 1.10) 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	to report is not defined by this standard. ITY (16) parameter data (see 1.5), 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-

## 1.10 WRITE (32) command

I

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

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 1.5.2), 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 1.0.3).

Byte\Bit	7	6	5	4	3	2	1	0	
0	OPERATION CODE (7Fh)								
1				CON	NTROL				
2				Res	erved				
5		_		No.					
6		Reserved			(	GROUP NUMB	ER		
7			A	DDITIONAL C	DB LENGTH (1	l8h)			
8	(MSB)				TION (000Bh	1)			
9				SERVICE AC		')		(LSB)	
10		WRPROTEC	Т	DPO	FUA	Reserved	FUA_NV	Reserved	
11				Reserved				XLB INVALID	
12	(MSB)		- LOGICAL BLOCK ADDRESS						
19									
20	(MSB)		- EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG						
23		- EXFECTED INITIAL LOGICAL BLOCK REFERENCE TAG						(LSB)	
24	(MSB)	- EXPECTED LOGICAL BLOCK APPLICATION TAG							
25								(LSB)	
26	(MSB)	- LOGICAL BLOCK APPLICATION TAG MASK							
27								(LSB)	
28	(MSB)			TRANSE					
31		– TRANSFER LENGTH –					(LSB)		

#### Table 23 — WRITE (32) command

See the WRITE (10) command (see 1.9) 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 1.9). 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 1.9) 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 22 in 1.9), 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 1.9).