To: T10 Committee
From: Gerry Houlder, Seagate Technology, gerry_houlder@seagate.com
Subj: More clarification of Application Tag behavior
Date: June 15, 2005

A previous proposal (05-101r1) http://www.t10.org/ftp/t10/document.05/05-101r1.pdf tries to restrict specific behavior of an Application Tag value of FFFFh to reads only. For reads, a value of FFFFh results in no checking of any protection bytes. Seagate doesn't agree with this – this behavior should apply to write commands also.

A common implementation of a target device using end-to-end protection feature is to use the same checking hardware for both read and write operations. It adds extra complication and opportunity for problems if the rules for reads and writes are different. Therefore the rules should be the same. The basic inputs are the CDB operation code (6 byte and 32 byte versions are special cases), 3 bit protection field (from CDB of reads and writes), and whether the Application Tag field in the data is FFFFh or not. T10 has crafted this feature to be the same for reads and writes in every way except the handling of Application Tag set to FFFFh. During discussion of this feature over the last year Seagate always thought the intent was to keep the handling exactly the same for reads and writes and we are distressed to learn that the wording doesn't require that today.

Handling the Application Tag value differently for writes than read makes no sense. If an initiator sends an Application Tag value of FFFFh to a target, why should the target be required to check the end-to-end protection fields before writing to the disk and then be required to ignore the protection fields during readback? This kind of behavior doesn't add to the reliability of the overall operation. Furthermore, the current wording requires the initiator to skip protection checking when the data reaches the initiator. This half-hearted rule doesn't fit the intent of increasing reliability.

This proposal describes changes needed to define the Application Tag value set to FFFFh as applying to writes as well as reads.

Changes to SBC-2

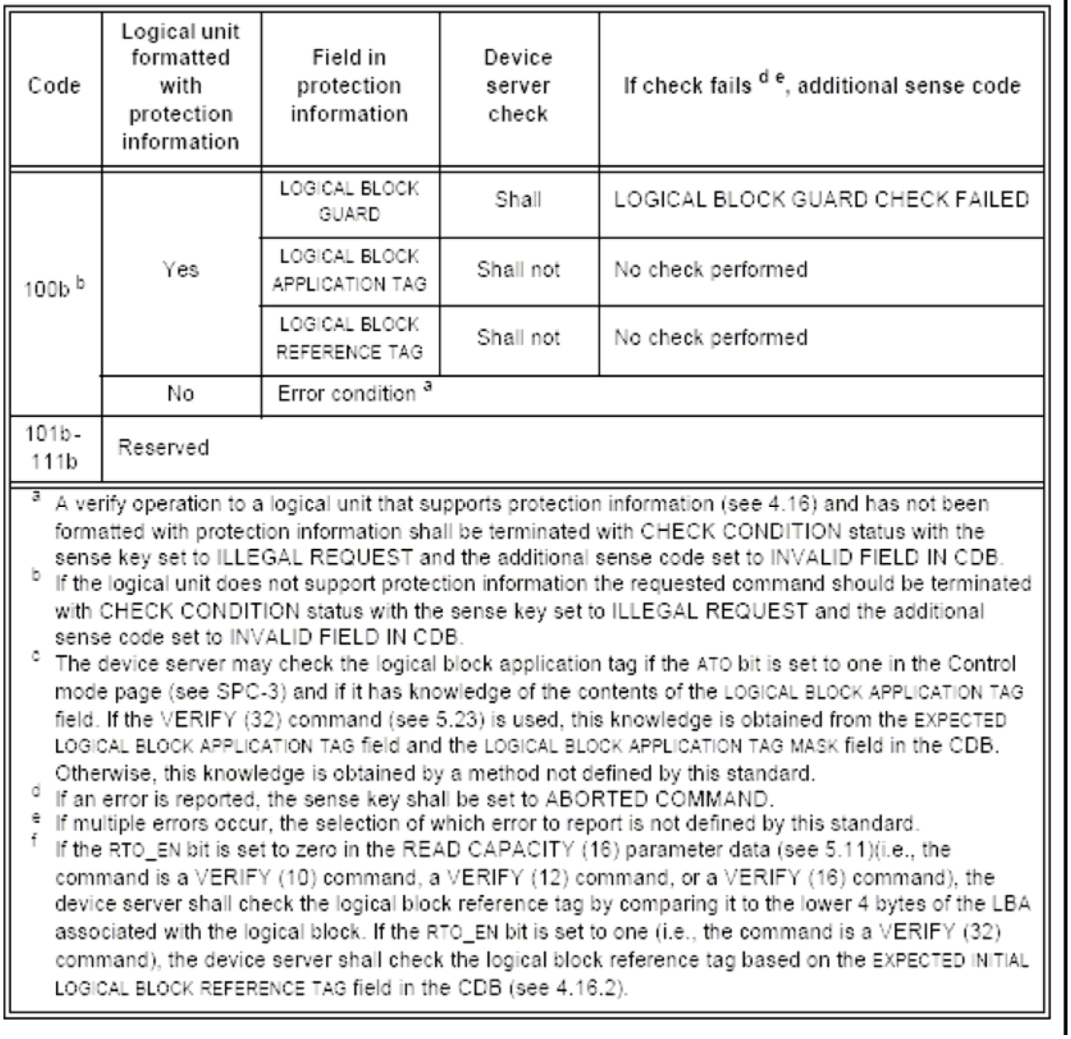
4.16.2 Protection information format

[This paragraph follows table 7. Text includes change proposed in 05-101r1.]

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 when reading ~~from the medium~~ or writing user data.

5.20 Verify(10) command

Table 56 — VRPROTECT field with BYTCHK set to one - checking protection information from the data-out buffer (part 3 of 3)



Code	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails [d] [e], additional sense code
100b [b]	Yes	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	Error condition [a]		
101b - 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. If the VERIFY (32) command (see 5.23) is used, this knowledge is obtained from the EXPECTED LOGICAL BLOCK APPLICATION TAG field and the LOGICAL BLOCK APPLICATION TAG MASK field in the CDB. Otherwise, this knowledge is obtained by a method not defined by this standard.

[d] If an error is reported, the sense key shall be set to ABORTED COMMAND.

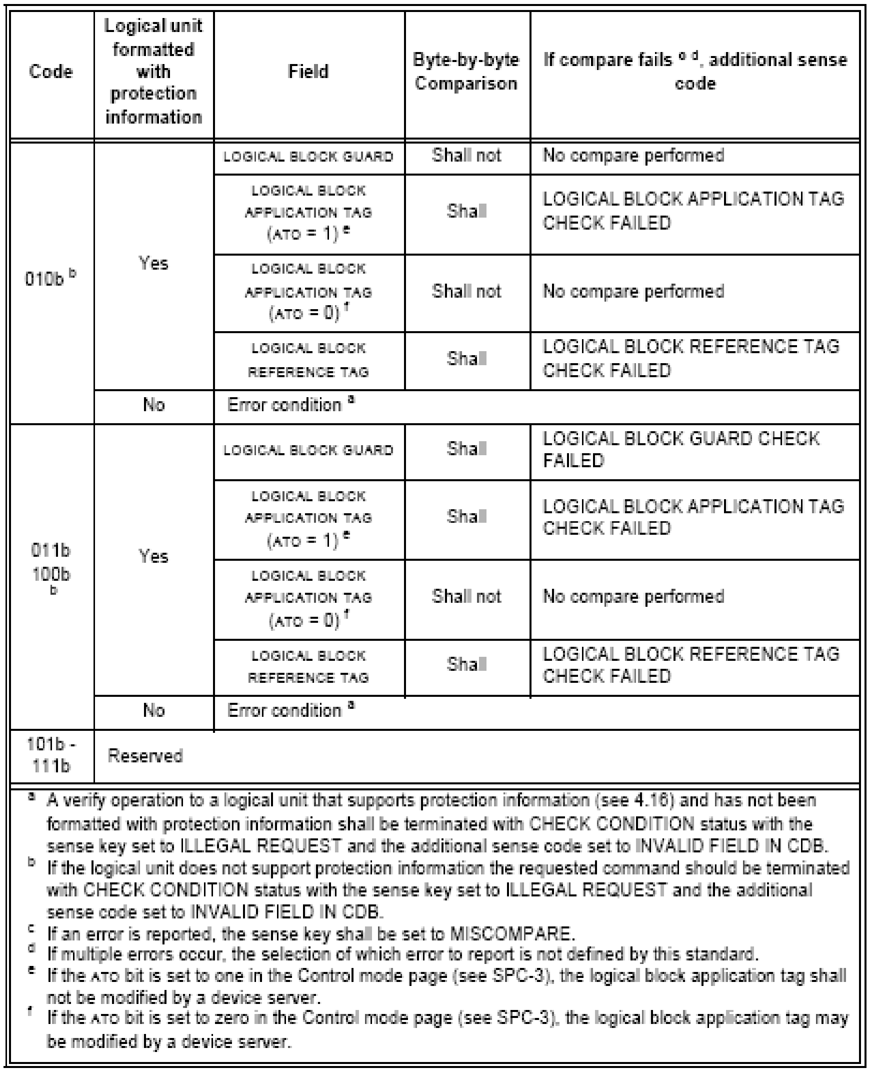
[e] If multiple errors occur, the selection of which error to report is not defined by this standard.

[f] If the RTO_EN bit is set to zero in the READ CAPACITY (16) parameter data (see 5.11)(i.e., the command is a VERIFY (10) command, a VERIFY (12) command, or a VERIFY (16) command), the device server shall check the logical block reference tag by comparing it to the lower 4 bytes of the LBA associated with the logical block. If the RTO_EN bit is set to one (i.e., the command is a VERIFY (32) command), the device server shall check the logical block reference tag based on the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field in the CDB (see 4.16.2).

The "Field in protection information" column header needs to add note "g".

g If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field set to FFFFh, the checking of all protection information shall be disabled for the associated logical block.

Table 57 — 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
010b [b]	Yes	LOGICAL BLOCK GUARD	Shall not	No compare performed
		LOGICAL BLOCK APPLICATION TAG (ATO = 1) [e]	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (ATO = 0) [f]	Shall not	No compare performed
		LOGICAL BLOCK REFERENCE TAG	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	No	Error condition [a]		
011b 100b [b]	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (ATO = 1) [e]	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (ATO = 0) [f]	Shall not	No compare performed
		LOGICAL BLOCK REFERENCE TAG	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	No	Error condition [a]		
101b - 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.

[e] 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.

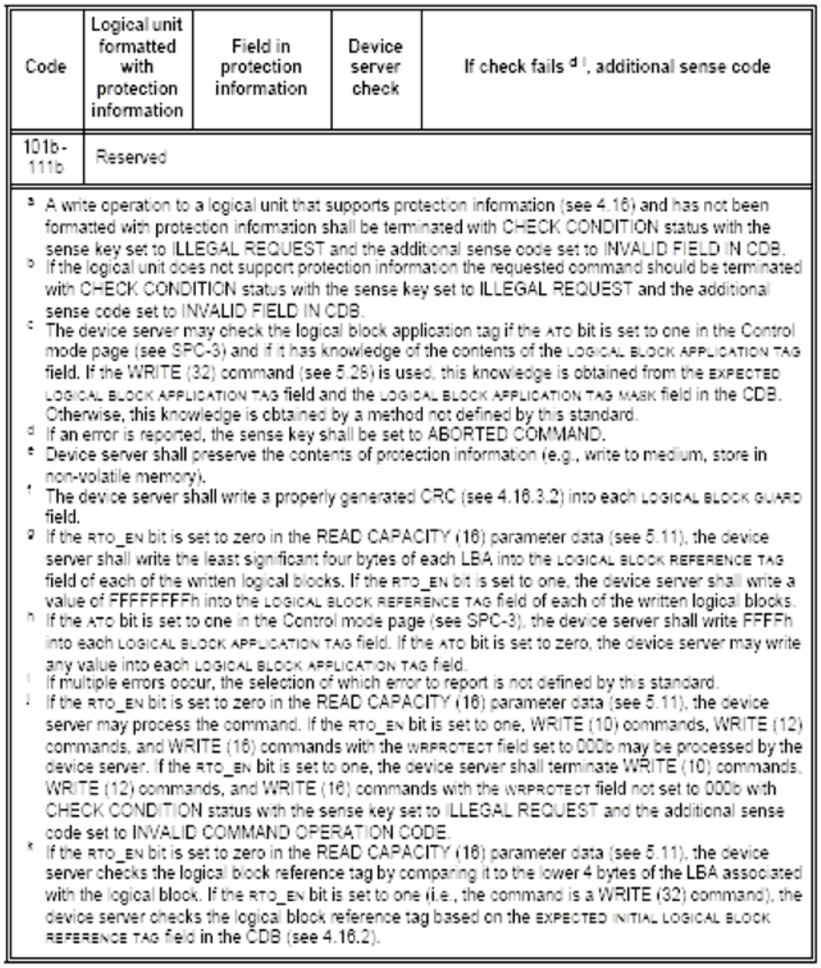
[f] If the ATO bit is set to zero in the Control mode page (see SPC-3), the logical block application tag may be modified by a device server.

Change the “Field” column to “Field in protection information” and add note “g”.

g If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field set to FFFFh, the checking of all protection information shall be disabled for the associated logical block.

5.25 Write(10) command

Table 63 — 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
101b - 111b	Reserved			

[a] A write operation to a logical unit that supports protection information (see 4.16) and has not been formatted with protection information shall be terminated with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

[b] If the logical unit does not support protection information the requested command should be terminated with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

[c] The device server may check the logical block application tag if the ATO bit is set to one in the Control mode page (see SPC-3) and if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. If the WRITE (32) command (see 5.28) is used, this knowledge is obtained from the EXPECTED LOGICAL BLOCK APPLICATION TAG field and the LOGICAL BLOCK APPLICATION TAG MASK field in the CDB. Otherwise, this knowledge is obtained by a method not defined by this standard.

[d] If an error is reported, the sense key shall be set to ABORTED COMMAND.

[e] Device server shall preserve the contents of protection information (e.g., write to medium, store in non-volatile memory).

[f] The device server shall write a properly generated CRC (see 4.16.3.2) into each LOGICAL BLOCK GUARD field.

[g] If the RTO_EN bit is set to zero in the READ CAPACITY (16) parameter data (see 5.11), the device server shall write the least significant four bytes of each LBA into the LOGICAL BLOCK REFERENCE TAG field of each of the written logical blocks. If the RTO_EN bit is set to one, the device server shall write a value of FFFFFFFFh into the LOGICAL BLOCK REFERENCE TAG field of each of the written logical blocks.

[h] If the ATO bit is set to one in the Control mode page (see SPC-3), the device server shall write FFFFh into each LOGICAL BLOCK APPLICATION TAG field. If the ATO bit is set to zero, the device server may write any value into each LOGICAL BLOCK APPLICATION TAG field.

[i] If multiple errors occur, the selection of which error to report is not defined by this standard.

[j] If the RTO_EN bit is set to zero in the READ CAPACITY (16) parameter data (see 5.11), the device server may process the command. If the RTO_EN bit is set to one, WRITE (10) commands, WRITE (12) commands, and WRITE (16) commands with the WRPROTECT field set to 000b may be processed by the device server. If the RTO_EN bit is set to one, the device server shall terminate WRITE (10) commands, WRITE (12) commands, and WRITE (16) commands with the WRPROTECT field not set to 000b with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID COMMAND OPERATION CODE.

[k] If the RTO_EN bit is set to zero in the READ CAPACITY (16) parameter data (see 5.11), the device server checks the logical block reference tag by comparing it to the lower 4 bytes of the LBA associated with the logical block. If the RTO_EN bit is set to one (i.e., the command is a WRITE (32) command), the device server checks the logical block reference tag based on the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field in the CDB (see 4.16.2).

The “Field in protection information” column header needs to add note “l” (letter l).

l. If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field set to FFFFh, the checking of all protection information shall be disabled for the associated logical block.