Date: February 23, 2004

To: T10 Committee (SCSI)

From: Jim Coomes (Seagate)

Subject:SBC 32 Byte Commands for End-to-End Data Protection

## Revision history

rev 6 -

a) The function of non-LBA locked reference tag is labeled application client reference tag ownership.
b) A bit is added to standard inquiry data to report support of application client ownership of the reference tag.
c) A bit is added to Format to enable application client ownership of the reference tag.
d) A bit is added to Long read capacity data to indicate when application client ownership reference tag is enabled.
e) The new 32 byte commands with reference tag seed are failed if the medium is not formatted with protection information and application client ownership of the reference tag enabled.
f) When the application client owns the reference tag, WRITE (6), VERIFY (10), (12), (16), WRITE AND VERIFY (10), (12), (16), WRITE SAME (10), (16) commands are failed.
g) When the application client owns the reference tag, READ commands with RDPROTECT equal 000b are allowed. READ commands with RDPROTECT 001b, 010b or 011d are failed.
h) When the application client owns the reference tag, Write commands with WRPROTECT equal 000b are allowed. WRITE commands with WRPROTECT 001b, 010b or 011d are failed.
i) Protection information field names DATA BLOCK XXX TAG were changed to LOGICAL BLOCK XXX TAG per SBC-2.

rev 5 - Updated per input at Nov 03 CAP meeting and implementation in SBC-2 r11.

a) Changed the modification of footnotes “c” in SBC-2 r11 to take the “may” out of 32 byte commands providing the application information knowledge.
b) Changed the xxPROTECT fields in all 32 byte commands to 3 bit fields.
c) The LOGICAL BLOCK APPLICATION VALUE field is renamed EXPECTED LOGICAL BLOCK APPLICATION TAG.
d) The descriptions of the new fields in the 32 byte commands are revised.

rev 4 - Added Write Same 32 byte command. Rewrite of the command field descriptions to direct the checking requirements back to the 10 byte version of the commands.

rev 3- Added 32 byte Verify and Write And Verify command.

rev 2 - This revision changed text in proposal 03-176r6 for the RDPROTECT field, Table Footnote c and the WRPROTECT field, Table Footnote c. Editorial convention was corrected for “a bit set to one”

rev 1 -This revision changes the RELADR field in the proposed commands to reserved, RDPROTECT and WRPROTECT field descriptions to reference 03-176r5 and qualifies LOGICAL BLOCK APPLICATION TAG checking with the RDPROTECT and WRPROTECT fields.

## Overview

There is a need to provide the initial value of the LOGICAL BLOCK REFERENCE TAG ~~proposed in 03-365rX~~ defined in SBC-2 on a command by command basis. One use case is in a configuration where a controller (e.g., a RAID) remaps the LBA to a different LBA space on a physical LUN. By passing the initial value of the LOGICAL BLOCK REFERENCE TAG in the command to the LUN, the original data protection block may be passed through the controller to the LUN and checked. This function provides end to end protection in the remapping case.

To provide the space for the initial LOGICAL BLOCK REFERENCE TAG and maintain 8 byte LBA space, 32 byte formats are proposed for read and write operations.

This proposal additionally provides a mechanism to enable device server checking to the LOGICAL BLOCK APPLICATION TAG in the protection information.

## Changes to document SBC-2 r~~11~~12

### 4.15.2 Protection information format

The LOGICAL BLOCK REFERENCE TAG field is ~~set to the least significant four bytes of the LBA to which~~ an incrementing value associated with the logical block ~~is associated~~. For commands that do not include an INITIAL LOGICAL BLOCK REFERENCE TAG field, the~~The~~ first logical block in application client data buffer shall contain the least significant four bytes of the LBA contained in the LOGICAL BLOCK ADDRESS field of the command associated with the logical block. For commands that include an INITIAL LOGICAL BLOCK REFERENCE TAG field, the first logical block transferred shall contain the LOGICAL BLOCK REFERENCE TAG equal to the value in the command. Each logical block in the application client data buffer contains a LOGICAL BLOCK REFERENCE TAG field with the logical block reference tag of the previous logical block plus one. 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.

### 5.9 READ (10) command

Table 32 RDPROTECT field, Footnote c

[c] The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. A READ (32) command provides this knowledge. Otherwise, the ~~The~~ method for acquiring ~~this~~ knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field is not defined by this standard

Table 32. RDPROTECT field, footnotes, add a new foot note in the Value column for values 001b, 010b, and 011b:

[h] If the RTO_EN bit (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one and the command type does not include the INITIAL LOGICAL BLOCK REFERENCE TAG field, the device server shall fail the command with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

### 5.22 VERIFY (10) command

Table 54, VRPROTECT field with BYTCHK set to zero, Footnote c

[c] The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. A VERIFY (32) command provides this knowledge. Otherwise, the ~~The~~ method for acquiring ~~this~~ knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field is not defined by this standard.

Table 54. VRPROTECT field, footnotes, add a new foot note in the Value column by values 000b, 001b, 010b, and 011b:

[h] If the RTO_EN bit (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one and the command type does not include the INITIAL LOGICAL BLOCK REFERENCE TAG field, the device server shall fail the command with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

Table 55 VRPROTECT field with BYTCHK set to one - checking data from application client Footnote c

[c] The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. A VERIFY (32) command provides this knowledge. Otherwise, the ~~The~~ method for acquiring ~~this~~ knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field is not defined by this standard.

Table 55. VRPROTECT field, footnotes, add a new foot note in the Value column by values 000b, 001b, 010b, and 011b:

[h] If the RTO_EN bit (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one and the command type does not include the INITIAL LOGICAL BLOCK REFERENCE TAG field, the device server shall fail the command with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

Table 56 VRPROTECT field with BYTCHK set to one - checking data on medium, Footnote c

[c] The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. A VERIFY (32) command provides this knowledge. Otherwise, the ~~The~~ method for acquiring ~~this~~ knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field is not defined by this standard.

Table 56. VRPROTECT field, footnotes, add a new foot note in the Value column by values 000b, 001b, 010b, and 011b:

[h] If the RTO_EN bit (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one and the command type does not include the INITIAL LOGICAL BLOCK REFERENCE TAG field, the device server shall fail the command with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

### 5.29 WRITE (10) command

Table 62. WRPROTECT field, footnotes

[c] The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. A WRITE (32) command provides this knowledge. Otherwise, the ~~The~~ method for acquiring ~~this~~ knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field is not defined by this standard.

[f] The device server shall write a properly generated CRC (see 4.15.3.2) into the LOGICAL BLOCK GUARD field~~, a properly calculated logical block reference tag into the LOGICAL BLOCK REFERENCE TAG field (see 4.15.2),~~ and if the APP_TAG_OWN bit in the Control mode page (see SPC-3) is set to one, FFFFh into the LOGICAL BLOCK APPLICATION TAG field as it writes the logical block to the medium. If the APP_TAG_OWN bit is set to zero, the device server may set the LOGICAL BLOCK APPLICATION TAG field to any value. If the RTO_EN bit (see 5.13) is set to zero, the device server shall write a properly calculated logical block reference tag based on the lower 4 bytes of the LBA into the LOGICAL BLOCK REFERENCE TAG field (see 4.15.2). If the RTO_EN bit is set to a one, the device server shall write a value of FFFFFFFFh into the LOGICAL BLOCK REFERENCE TAG field.

Table 62. WRPROTECT field, footnotes, add a new foot note for the Value column by values 000b, 001b, 010b, and 011b:

[h] If the RTO_EN bit (see 5.13) is set to zero, the device server may process the command. If the RTO_EN bit is set to one, WRITE (10), WRITE (12), 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 fail the following commands with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE:

a) WRITE (6) command,
b) WRITE (10), WRITE (12), or WRITE (16) commands with the WRPROTECT field set to 001b, 010b, 011b, or
c) a command type does not include the INITIAL LOGICAL BLOCK REFERENCE TAG field.

## 5.3 FORMAT UNIT command

**Table 12 — FORMAT UNIT command**

| Byte\Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | OPERATION CODE (04h) | | | | | | | |
| 1 | FMTPINFO | RTO_REQ | LONGLIST | FMTDATA | CMPLIST | DEFECT LIST FORMAT | | |
| 2 | Vendor specific | | | | | | | |
| 3 | (MSB) | INTERLEAVE | | | | | | |
| 4 | | | | | | | | (LSB) |
| 5 | CONTROL | | | | | | | |

If the FMTPINFO bit set to zero, the reference tag own request (RTO_REQ) bit is ignored. If the FMTPINFO bit 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 in protection information (see 4.15). If the FMTPINFO bit set to one and the RTO_REQ bit is set to zero, the device server shall disable application client ownership of the LOGICAL BLOCK REFERENCE TAG field. A successful format that changes whether protection information is included shall cause the RTO_EN bit in the READ CAPACITY (16) data to be changed.

## 5.13 READ CAPACITY (16) command

**Table 38 — Long read capacity data**

| Byte\Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | (MSB) | RETURNED LOGICAL BLOCK ADDRESS | | | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | (MSB) | BLOCK LENGTH IN BYTES | | | | | | |
| 11 | | | | | | | | (LSB) |
| 12 | Reserved | | | | | | RTO_EN | PROT_EN |
| 13 | Reserved | | | | | | | |
| 31 | | | | | | | | |

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 with protection information (see 4.15) enabled and the RTO_REQ bit set to one for the format). A RTO_EN bit set to zero indicates that application client ownership of the LOGICAL BLOCK REFERENCE TAG field in protection information is disabled.

## Additions to document SBC-2 r12

## 5.xx READ (32) Command

The READ (32) command (see table Table xx —) requests that the device server transfer data to the application client. The most recent data value written in the addressed logical block shall be returned.

**Table xx — READ (32) command**

| Byte\Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | OPERATION CODE (7Fh) | | | | | | | |
| 1 | CONTROL | | | | | | | |
| 2 | Reserved | | | | | | | |
| 6 | | | | | | | | |
| 7 | ADDITIONAL CDB LENGTH (18h) | | | | | | | |
| 8 | (MSB) | SERVICE ACTION (TBDh) | | | | | | |
| 9 | | | | | | | | (LSB) |
| 10 | RDPROTECT | | | DPO | FUA | Reserved | | |
| 11 | Reserved | | | | | | | |
| 12 | (MSB) | LOGICAL BLOCK ADDRESS | | | | | | |
| 19 | | | | | | | | (LSB) |
| 20 | (MSB) | 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) | TRANSFER LENGTH | | | | | | |
| 31 | | | | | | | | (LSB) |

If the RTO_EN bit (see 5.13) is set to zero, the device server shall fail the READ (32) command with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE. If the RTO_EN bit is set to one may process the command.

The INITIAL LOGICAL BLOCK REFERENCE TAG field contains the value of the LOGICAL BLOCK REFERENCE TAG expected on the first logical block of the range of logical blocks for this command. See the READ (10) command (see 5.2.8) for a description of the checking enables and requirements.

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled by the RDPROTECT field (see 5.2.8) and the APP_CHK bit in the Protection Information VPD page (see SPC-3), the EXPECTED LOGICAL BLOCK APPLICATION TAG field contains a value that is expected in the LOGICAL BLOCK APPLICATION TAG with the LOGICAL BLOCK APPLICATION TAG MASK applied in the protection information of logical blocks for this command.

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled by the RDPROTECT field (see 5.2.8) and the APP_CHK bit in the Protection Information VPD page (see SPC-3), the LOGICAL BLOCK APPLICATION MASK field contains a value that is a bit mask for enabling the checking of the LOGICAL BLOCK APPLICATION TAG in the protection information for each logical block of the range of logical blocks for this command. A LOGICAL BLOCK APPLICATION TAG MASK bit set to one enables the checking of the corresponding bit in the EXPECTED LOGICAL BLOCK APPLICATION TAG field with the LOGICAL BLOCK APPLICATION TAG.

See 4.2.1.8 for reservation requirements for this command. See the READ (10) command (see 5.2.8) for a description of the other fields in this command.

## 5.xx WRITE (32) command

The WRITE (32) command (see table Table xx —) requests that the device server write the data transferred from the application client to the medium

**Table xx — WRITE (32) command**

| Byte\Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | OPERATION CODE (7Fh) | | | | | | | |
| 1 | CONTROL | | | | | | | |
| 2 | Reserved | | | | | | | |
| 6 | | | | | | | | |
| 7 | ADDITIONAL CDB LENGTH (18h) | | | | | | | |
| 8 | (MSB) | SERVICE ACTION (TBDh) | | | | | | |
| 9 | | | | | | | | (LSB) |
| 10 | WRPROTECT | | | DPO | FUA | Reserved | | |
| 11 | Reserved | | | | | | | |
| 12 | (MSB) | LOGICAL BLOCK ADDRESS | | | | | | |
| 19 | | | | | | | | (LSB) |
| 20 | (MSB) | 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) | TRANSFER LENGTH | | | | | | |
| 31 | | | | | | | | (LSB) |

If the RTO_EN bit (see 5.13) is set to zero, the device server shall fail the WRITE (32) command with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE. If the RTO_EN bit is set to one may process the command.

The INITIAL LOGICAL BLOCK REFERENCE TAG field contains the value of the LOGICAL BLOCK REFERENCE TAG expected on the first logical block of the range of logical blocks for this command. See the WRITE (10) command (see 5.2.31) for a description of the checking enables and requirements.

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled by the WRPROTECT field (see 5.2.31) and the APP_CHK bit in the Protection Information VPD page (see SPC-3), the EXPECTED LOGICAL BLOCK APPLICATION TAG field contains a value that is expected in the LOGICAL BLOCK APPLICATION TAG with the LOGICAL BLOCK APPLICATION TAG MASK applied in the protection information of logical blocks for this command.

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled by the WRPROTECT field (see 5.2.31) and the APP_CHK bit in the Protection Information VPD page (see SPC-3), the LOGICAL BLOCK APPLICATION MASK field contains a value that is a bit mask for enabling the checking of the LOGICAL BLOCK APPLICATION TAG in the protection information for each logical block of the range of logical blocks for this command. A LOGICAL BLOCK

APPLICATION TAG MASK bit set to one enables the checking of the corresponding bit in the EXPECTED LOGICAL BLOCK APPLICATION TAG field with the LOGICAL BLOCK APPLICATION TAG.

See 4.2.1.8 for reservation requirements for this command. See the WRITE (10) command (see 5.2.31) for a description of the other fields in this command.

## 5.xx WRITE AND VERIFY (32) Command

The WRITE AND VERIFY (32) command (see table Table xx —) requests that the device server write the data transferred from the application client to the medium and then verify that the data and protection information, if any, is correctly written. The data is only transferred once from the application client to the device server.

**Table xx — WRITE AND VERIFY (32) command**

| Byte\Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | OPERATION CODE (7Fh) | | | | | | | |
| 1 | CONTROL | | | | | | | |
| 2 | Reserved | | | | | | | |
| 6 | | | | | | | | |
| 7 | ADDITIONAL CDB LENGTH (18h) | | | | | | | |
| 8 | (MSB) | SERVICE ACTION (TBDh) | | | | | | |
| 9 | | | | | | | | (LSB) |
| 10 | WRPROTECT | | | DPO | Reserved | EBP | BYTCHK | Reserved |
| 11 | Reserved | | | | | | | |
| 12 | (MSB) | LOGICAL BLOCK ADDRESS | | | | | | |
| 19 | | | | | | | | (LSB) |
| 20 | (MSB) | 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) | TRANSFER LENGTH | | | | | | |
| 31 | | | | | | | | (LSB) |

If the RTO_EN bit (see 5.13) is set to zero, the device server shall fail the WRITE and VERIFY (32) command with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE. If the RTO_EN bit is set to one may process the command.

The INITIAL LOGICAL BLOCK REFERENCE TAG field contains the value of the LOGICAL BLOCK REFERENCE TAG expected on the first logical block of the range of logical blocks for this command. See the WRITE AND VERIFY(10) command (see 5.2.34) for a description of the checking enables and requirements.

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled by the WRPROTECT field (see 5.2.31) and the APP_CHK bit in the Protection Information VPD page (see SPC-3), the EXPECTED LOGICAL BLOCK APPLICATION TAG field contains a value that is expected in the LOGICAL BLOCK APPLICATION TAG with the LOGICAL BLOCK APPLICATION TAG MASK applied in the protection information of logical blocks for this command.

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled by the WRPROTECT field (see 5.2.31) and the APP_CHK bit in the Protection Information VPD page (see SPC-3), the LOGICAL BLOCK APPLICATION MASK field contains a value that is a bit mask for enabling the checking of the LOGICAL BLOCK APPLICATION TAG in the protection information for each logical block of the range of logical blocks for this command. A LOGICAL BLOCK APPLICATION TAG MASK bit set to one enables the checking of the corresponding bit in the EXPECTED LOGICAL BLOCK APPLICATION TAG field with the LOGICAL BLOCK APPLICATION TAG.

See 4.2.1.8 for reservation requirements for this command. See the WRITE AND VERIFY(10) command (see 5.2.34) for a description of the other fields in this command.

## 5.xx VERIFY (32) Command

The VERIFY (32) command (see table Table xx —) requests that the device server verify the data on the medium.

**Table xx — VERIFY (32) command**

| Byte\Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | OPERATION CODE (7Fh) | | | | | | | |
| 1 | CONTROL | | | | | | | |
| 2 | Reserved | | | | | | | |
| 6 | | | | | | | | |
| 7 | ADDITIONAL CDB LENGTH (18h) | | | | | | | |
| 8 | (MSB) | | | SERVICE ACTION (TBD) | | | | |
| 9 | | | | | | | | (LSB) |
| 10 | VRPROTECT | | | DPO | Reserved | EBP | BYTCHK | Reserved |
| 11 | Reserved | | | | | | | |
| 12 | (MSB) | | | LOGICAL BLOCK ADDRESS | | | | |
| 19 | | | | | | | | (LSB) |
| 20 | (MSB) | | | 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) | | | TRANSFER LENGTH | | | | |
| 31 | | | | | | | | (LSB) |

If the RTO_EN bit (see 5.13) is set to zero, the device server shall fail the VERIFY (32) command with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE. If the RTO_EN bit is set to one may process the command.

The INITIAL LOGICAL BLOCK REFERENCE TAG field contains the value of the LOGICAL BLOCK REFERENCE TAG expected on the first logical block of the range of logical blocks for this command. See the VERIFY (10) command (see 5.2.27) for a description of the checking enables and requirements.

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled by the VRPROTECT field (see 5.2.27) and the APP_CHK bit in the Protection Information VPD page (see SPC-3), the EXPECTED LOGICAL BLOCK APPLICATION

TAG field contains a value that is expected in the LOGICAL BLOCK APPLICATION TAG with the LOGICAL BLOCK APPLICATION TAG MASK applied in the protection information of logical blocks for this command.

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled by the VRPROTECT field (see 5.2.27) and the APP_CHK bit in the Protection Information VPD page (see SPC-3), the LOGICAL BLOCK APPLICATION MASK field contains a value that is a bit mask for enabling the checking of the LOGICAL BLOCK APPLICATION TAG in the protection information for each logical block of the range of logical blocks for this command. A LOGICAL BLOCK APPLICATION TAG MASK bit set to one enables the checking of the corresponding bit in the EXPECTED LOGICAL BLOCK APPLICATION TAG field with the LOGICAL BLOCK APPLICATION TAG.

See 4.2.1.8 for reservation requirements for this command. See the VERIFY (10) command (see 5.2.27) for a description of the other fields in this command.

## 5.xx WRITE SAME (32) Command

The WRITE SAME (32) command (see table Table xx —) requests that the device server write the single block of data transferred by the application client to the medium multiple times to consecutive multiple logical blocks.

**Table xx — WRITE SAME (32) command**

| Byte\Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | OPERATION CODE (7Fh) | | | | | | | |
| 1 | | CONTROL | | | | | | |
| 2 | Reserved | | | | | | | |
| 6 | | | | | | | | |
| 7 | | ADDITIONAL CDB LENGTH (18h) | | | | | | |
| 8 | (MSB) | SERVICE ACTION (TBD) | | | | | | |
| 9 | | | | | | | | (LSB) |
| 10 | WRPROTECT | | | Reserved | | PBDATA | LBDATA | Reserved |
| 11 | Reserved | | | | | | | |
| 12 | (MSB) | LOGICAL BLOCK ADDRESS | | | | | | |
| 19 | | | | | | | | (LSB) |
| 20 | (MSB) | INITIAL LOGICAL BLOCK REFERENCE TAG | | | | | | |
| 23 | | | | | | | | (LSB) |
| 24 | (MSB) | EXPECTED LOGICAL BLOCK APPLICATION TAG | | | | | | |
| 25 | | | | | | | | (LSB) |
| 26 | (MSB) | Reserved | | | | | | |
| 27 | | | | | | | | (LSB) |
| 28 | (MSB) | TRANSFER LENGTH | | | | | | |
| 31 | | | | | | | | (LSB) |

If the RTO_EN bit (see 5.13) is set to zero, the device server shall fail the WRITE SAME (32) command with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE. If the RTO_EN bit is set to one may process the command.

The INITIAL LOGICAL BLOCK REFERENCE TAG field contains the value of the LOGICAL BLOCK REFERENCE TAG expected in the block of data transferred by the application client for this command and the LOGICAL BLOCK

REFERENCE TAG for the first logical block written to the medium. See the WRITE SAME (10) command (see 5.2.38) for a description of the checking enables and requirements.

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled by the WRPROTECT field (see 5.2.31) and the APP_CHK bit in the Protection Information VPD page (see SPC-3), the EXPECTED LOGICAL BLOCK APPLICATION TAG field contains a value that is expected in the LOGICAL BLOCK APPLICATION TAG with the LOGICAL BLOCK APPLICATION TAG MASK applied in the protection information of the block of data transferred by the application client for this command.

When checking of the LOGICAL BLOCK APPLICATION TAG is enabled by the WRPROTECT field (see 5.2.31) and the APP_CHK bit in the Protection Information VPD page (see SPC-3), the LOGICAL BLOCK APPLICATION MASK field contains a value that is a bit mask for enabling the checking of the LOGICAL BLOCK APPLICATION TAG in the protection information of the block of data transferred by the application client for this command. A LOGICAL BLOCK APPLICATION TAG MASK bit set to one enables the checking of the corresponding bit in the EXPECTED LOGICAL BLOCK APPLICATION TAG field with the LOGICAL BLOCK APPLICATION TAG.

See 4.2.1.8 for reservation requirements for this command. See the WRITE SAME (10) command (see 5.2.38) for a description of the other fields in this command.

## Changes to document SPC-3 r17

### 6.4.2 .Standard INQUIRY data

**Table 76 — Standard INQUIRY data format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | PERIPHERAL QUALIFIER | | | PERIPHERAL DEVICE TYPE | | | | |
| 1 | RMB | Reserved | | | | | | |
| 2 | VERSION | | | | | | | |
| 3 | Obsolete | Obsolete | NORMACA | HISUP | RESPONSE DATA FORMAT | | | |
| 4 | ADDITIONAL LENGTH (n-4) | | | | | | | |
| 5 | SCCS | ACC | ALUA | | 3PC | Reserved | RFTG_OWN | PROTECT |
| 6 | BQUE | ENCSERV | VS | MULTIP | MCHNGR | Obsolete | Obsolete | ADDR16[a] |
| 7 | Obsolete | Obsolete | WBUS16[a] | SYNC[a] | LINKED | Obsolete | CMDQUE | VS |
| 8 | (MSB) | VENDOR IDENTIFICATION | | | | | | |
| 15 | | | | | | | | (LSB) |
| 16 | (MSB) | PRODUCT IDENTIFICATION | | | | | | |
| 31 | | | | | | | | (LSB) |
| 32 | (MSB) | PRODUCT REVISION LEVEL | | | | | | |
| 35 | | | | | | | | (LSB) |
| 36 | Vendor specific | | | | | | | |
| 55 | | | | | | | | |
| 56 | Reserved | | | | CLOCKING[a] | | QAS[a] | IUS[a] |
| 57 | Reserved | | | | | | | |
| 58 | (MSB) | VERSION DESCRIPTOR 1 | | | | | | |
| 59 | | | | | | | | (LSB) |
| | ⋮ | | | | | | | |
| 72 | (MSB) | VERSION DESCRIPTOR 8 | | | | | | |
| 73 | | | | | | | | (LSB) |
| 74 | Reserved | | | | | | | |
| 95 | | | | | | | | |
| | Vendor specific parameters | | | | | | | |
| 96 | Vendor specific | | | | | | | |
| n | | | | | | | | |

[a] The meanings of these fields are specific to SPI-5 (see 6.4.3). For SCSI protocols other than the SCSI Parallel Interface, these fields are reserved.

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