T10/02-095r0

2002mar01

Joe Breher

Exabyte Corp

joebre@exabyte.com

Background

SSC-2 recently completed a successful letter ballot 01-350r0. The results, including all comments are available as 01-351r0.

In letter ballot comment resolution of 2002jan14, Exabyte comment 15 was accepted in principle. In this discussion, it was noted that SPC-3 should probably incorporate the same change. Joe Breher was given an action item at that meeting to request the relevant change at the CAP WG.

Joe Breher subsequently posted 02-044r0, “Resolving ambiguity in sense data information field for sequential buffered mode”. An agenda item was granted to discuss this proposal at the 2002jan16 CAP WG meeting (item 5.15).

At the CAP WG, it was pointed out that agenda item 5.5 would affect the proposed changes of 02-044r0. This agenda item discussed 01-199, “SPC-3 SBC-2 SSC-2 sense data changes”, brought to the meeting in r2 form. The owners of 01-199 and 02-044 agreed to incorporate the principles of 02-044 into 01-199. This inclusion of 02-044 yielded part of the changes from 01-199r2 to 01-199r3. The CAP WG then voted to include 01-199r3 into SPC-3, SBC-2, and SSC-2.

Upon further examination, it appears that several parts of 01-199r3 have inadvertently not incorporated the principles within 02-044. This proposal attempts to rectify this omission.

Additionally, ENDL Comment 176 questioned the accuracy of section 5.7.1. I assert herein that 5.7.1 is indeed in error, and attempt to repair it. The specific issue is that There is no FIXED bit in the WRITE FILEMARKS(16) CDB. It is therefore nonsensical to return differing information according to the state of this bit. Accordingly, I suggest using WRITE FILEMARKS(6) as a prototype for WRITE FILEMARKS(16).

The changes proposed herein affect only SSC-2. They will be discussed as deltas from SSC-2, as if the approved changes in 01-199r3 have already been performed to the SSC-2 document.

Changes to SSC-2

5.6.1 WRITE(16) command introduction

…
D) if the device is operating in buffered mode and the FIXED bit is set to zero, the INFORMATION field shall be set to the total number of bytes, filemarks, and setmarks not written (the number of bytes not transferred from the application client plus the number of bytes, filemarks, and setmarks remaining in the logical unit's buffer).
Change to
…(the number of data bytes not transferred from the application client plus the number of data bytes, plus the number of filemarks, plus the number of setmarks remaining in the logical unit's buffer).

5.7.1 WRITE FILEMARKS(16) command introduction

…
The INFORMATION field shall be defined as follows:
A) If the device is operating in unbuffered mode (see 3.1.58) and the FIXED bit is set to one, the

INFORMATION field shall be set to the requested transfer length minus the actual number of blocks written;
B) if the device is operating in unbuffered mode and the FIXED bit is set to zero, the INFORMATION field shall be set to the requested transfer length;
C) if the device is operating in buffered mode (see 3.1.8) and the FIXED bit is set to one, the INFORMATION field shall be set to the total number of blocks, filemarks, and setmarks not written (the number of blocks not transferred from the application client plus the number of blocks, filemarks, and setmarks remaining in the logical unit's buffer. The value in the INFORMATION field may exceed the transfer length; and
D) if the device is operating in buffered mode and the FIXED bit is set to zero, the INFORMATION field shall be set to the total number of bytes, filemarks, and setmarks not written (the number of bytes not transferred from buffer).
Change to
The INFORMATION field shall be defined as follows:
A) If the device is operating in unbuffered mode (see 3.1.58) the INFORMATION field shall be set to the requested transfer length minus the actual number of filemarks or setmarks written;
B) if the device is operating in buffered mode and the buffered data was written in variable block mode (see 6.8 or 5.6), the INFORMATION field shall be set to the total number of data bytes, filemarks, and setmarks not written (the number of filemarks or setmarks not transferred from the application client plus the number data bytes, plus the number of filemarks, plus the number of setmarks remaining in the logical unit's buffer).
C) if the device is operating in buffered mode (see 3.1.8) and the buffered data was written in fixed block mode (see 6.8 or 5.6), the INFORMATION field shall be set to the total number of blocks, filemarks, and setmarks not written (the number filemarks or setmarks not transferred from the application client plus the number of logical blocks, plus the number of filemarks, plus the number of setmarks remaining in the logical unit's buffer).

6.8.1 WRITE(6) command introduction

…
and D) if the device is operating in buffered mode and the FIXED bit is set to zero, the INFORMATION field shall be set to the total number of bytes, filemarks, and setmarks not written (the number of bytes not transferred from the application client plus the number of bytes, filemarks, and setmarks remaining in the logical unit's buffer).
Change to
D) if the device is operating in buffered mode and the FIXED bit is set to zero, the INFORMATION field shall be set to the total number of data bytes, filemarks, and setmarks not written (the number of data bytes not transferred from the application client plus the number data bytes, plus the number of filemarks, plus the number of setmarks remaining in the logical unit's buffer).

6.9.1 WRITE FILEMARKS(6) command introduction

…
B) if the device is operating in buffered mode (see 3.1.8) and the buffered data was written in variable block mode (see 6.8 or 5.6), the INFORMATION field shall be set to the total number of bytes, filemarks, and setmarks not written (the number of filemarks or setmarks not transferred from the application client plus the number of bytes, filemarks and setmarks remaining in the logical unit's buffer). It is possible for the value in the INFORMATION field to exceed the transfer length; or
Change to
B) if the device is operating in buffered mode and the buffered data was written in variable block mode (see 6.8 or 5.6), the INFORMATION field shall be set to the total number of data bytes, filemarks, and setmarks not written (the number of filemarks or setmarks not transferred from the application client plus the number data bytes, plus the number of filemarks, plus the number of setmarks remaining in the logical unit's buffer).

Storm clouds on the horizon

With or without these proposed changes, residual count reporting for buffered data is, and always was, broken. Herein is a discussion of the problems.

While sense data for WRITE and WRITE FILEMARKS report residual data count in the INFORMATION field, several commands (e.g. TEST UNIT READY) do not. Deferred error reporting for these other commands really should be expanded in order to report the residual count. I suppose it could be argued that, upon receiving a deferred error due to a failed buffered write, the initiator could issue another arbitrary WRITE or WRITE FILEMARKS, expecting it to fail. The returned sense data for this subsequent command would specify the residual count in the buffer. While this appears to be a workable scheme for determining the residual data count, it is philosophically troublesome.

More egregious is the residual reporting for WRITE FILEMARKS. The specific issue is that the INFORMATION field is reported based upon “the buffered data was written in variable block mode” or “the buffered data was written in fixed block mode”. However, the initiator can freely alternate sending WRITE commands with FIXED=0 with WRITE commands with FIXED=1. As such, at the time the error is encountered, the buffer may contain a mix of data written in fixed block mode and data written in variable block mode. What is the device to report under these conditions? I cannot find anywhere in the specification that this condition is addressed. One may argue that an initiator shouldn’t do that. However, that seems a rather weak argument, from a specification standpoint.