

To: Membership of X3T9.2

From: Edward Lappin  
Exabyte Corporation  
tedl@exabyte.com

Date: November 15, 1993

Subject: Read Position Command for SCSI-3 Sequential Command Sets (SSC)

---

I am proposing a change to the READ POSITION command to return the complete current logical position. This new feature is proposed to be included in the SCSI-3 Sequential Commands (SSC) project. Support of this feature is optional.

## 1. Background

Currently, there is no mechanism for the initiator to determine the current file or set number from a target unless the initiator counts each mark encountered. If unrecoverable read errors occur, the target will make assumptions regarding the current file and set. These assumptions are undefined in SCSI-2 and may affect data recovery procedures.

Normally, the initiator can keep track of the current file and set number. However, if there is a media error that prevents the reading of one or more physical blocks, the file and set numbers may become non-deterministic from the initiator's point of view. It should be noted that Request Sense can return MEDIUM ERROR with FMK set to one but this does not differentiate between setmarks and filemarks. The current logical position also depends on the target's positioning after an unrecoverable error is encountered.

The determination of the current position under non-error conditions can be done either by the target or the initiator. Since (we assume) the initiator knows the position it is at, it knows the current set and file number. However, it may not always keep track of both, and the set number may not be correctly retained because of the setting or clearing of the RMSK bit (Mode page 0x10, byte 8). Normally, the initiator needs to know boundaries, not absolute Set or File numbers. Therefore, this may not be a problem.

The determination of the position when there is a non-recoverable media error is more complicated. If the area containing the error only consists of data blocks, there is no problem. However, if setmarks and filemarks are present, it is not clear when each should be reported and how the initiator knows that they have been found. If either setmarks and filemarks but not both are present, the FMK bit in Request Sense can be used. This only works if the initiator can

accurately determine which type of mark can be present in the affected area and if the initiator looks at the FMK bit when media errors occur. However, if both types of marks exist, the drive will not know the order and the initiator will not know the assumptions made.

## 2. Proposal

If the target assumed that the current logical position, including file and set numbers, could be returned to the initiator, then the initiator could determine what the drive assumed, not necessarily the same as what is on the tape. It should be noted, assuming the error occurred on a sequence of reads, that the first media error could have affected the set or file number. To accurately determine which, it is necessary to Space back one block, Read Position, Read one block, and Read Position again. After this operation, the initiator can issue a series of Reads until a good status is returned. Then, the initiator can issue a Read Position to determine what was skipped.

Another recovery uses the next good position. If the initiator can command the target to find the next good block and then read the complete current logical position, the initiator may be able to continue with minimal data loss knowing exactly where the drive has located to.

### 3. Changes

The following CDB structure replaces the READ POSITION command in the Sequential Command Sets (SSC) document under SCSI-3. The text and alternate return data format following the new CDB structure will be added to the description associated with the READ POSITION command in the SSC document.

| Bit<br>Byte | 7                    | 6 | 5 | 4        | 3 | 2 | 1    | 0  |
|-------------|----------------------|---|---|----------|---|---|------|----|
| 0           | Operation Code (34h) |   |   |          |   |   |      |    |
| 1           | Logical Unit Number  |   |   | Reserved |   |   | TCLP | BT |
| 2           | Reserved             |   |   |          |   |   |      |    |
| 3           | Reserved             |   |   |          |   |   |      |    |
| 4           | Reserved             |   |   |          |   |   |      |    |
| 5           | Reserved             |   |   |          |   |   |      |    |
| 6           | Reserved             |   |   |          |   |   |      |    |
| 7           | Reserved             |   |   |          |   |   |      |    |
| 8           | Reserved             |   |   |          |   |   |      |    |
| 9           | Control              |   |   |          |   |   |      |    |

A Total Current Logical Position (TCLP) bit of one indicates the target shall return data specifying the partition, file, and set number with the current logical position. A TCLP bit of zero indicates the target shall return data specifying the first and last block location with the number of bytes and blocks in the buffer. Support of TCLP is optional. If the target does not implement Total Current Logical Position and the TCLP bit is set to one, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

A block address type (BT) bit of one requests the target to return the first block location, the last block location, and block number fields as device-specific values. A BT bit of zero requests the target to return the first block location, the last block location, and block number fields as an SCSI logical block address (relative to a partition).

The following READ POSITION data shall be returned if the TCLP is set to 1.

| Bit<br>Byte | 7                | 6            | 5        | 4 | 3   | 2   | 1        | 0     |
|-------------|------------------|--------------|----------|---|-----|-----|----------|-------|
| 0           | BOP              | EOP          | Reserved |   | MPU | BPU | Reserved |       |
| 1           | Partition Number |              |          |   |     |     |          |       |
| 2           | Reserved         |              |          |   |     |     |          |       |
| 3           | Reserved         |              |          |   |     |     |          |       |
| 4           | (MSB)            | Block Number |          |   |     |     |          |       |
| 7           |                  |              |          |   |     |     |          | (LSB) |
| 8           | (MSB)            | File Number  |          |   |     |     |          |       |
| 11          |                  |              |          |   |     |     |          | (LSB) |
| 12          | (MSB)            | Set Number   |          |   |     |     |          |       |
| 15          |                  |              |          |   |     |     |          | (LSB) |
| 16          | Reserved         |              |          |   |     |     |          |       |
| 19          |                  |              |          |   |     |     |          |       |

The BOP, EOP, BPU, and Partition Number fields are as defined in the READ POSITION data returned when TLCP is set to 0.

A Mark Position Unknown (MPU) bit of one indicates the File Number and Set Number are unknown. A MPU bit of zero indicates the File Number and Set Number fields contain valid position information.

**NOTE:** If the target is assuming a position due to unrecoverable read errors, the MPU and BPU fields should be set to zero where possible. The MPU and BPU bits shall only be set if the target cannot assume or does not know the current mark or block, respectively.

The **Block Number** shall report the number of logical blocks between beginning-of-partition and the current logical position. Setmarks and filemarks count as one logical block each.

The **File Number** shall report the number of filemarks between beginning-of-partition and the current logical position.

The **Set Number** shall report the number of setmarks between beginning-of-partition and the current logical position.