

Date: Aug. 17, 1995  
To: X3T10 Committee  
From: Gerry Houlder, Seagate Technology  
Subj: Set Capacity Proposal

This proposal came about because some customers want to define the capacity reported by a disk drive as something less than its full capacity. The main reason for this is customers wanting drives from different vendors to have exactly the same number of logical blocks. This is desired for ease of replacement, especially in disk array applications.

Another reason is that drives are getting too large for some operating systems to handle. For example, many operating systems will only handle 8 GB of storage on a single device and several vendors are making drives with a larger capacity than this. In this case, the customer may want to map part of the capacity as LUN 0 and another part as LUN 1. This proposal addresses that need also.

In discussions at the July 11 working group meeting, it was pointed out that several vendors (including Seagate) use the Number of Blocks field in the MODE SELECT block descriptor to achieve this purpose. This is only a 3 byte field, however, so an extension of some sort would have to be made so that 4 byte values can be used. About 60% of that group felt that the backwards compatibility issues caused by such an extension made this option unacceptable.

The group did favor adding a set capacity capability to the READ CAPACITY command. The changes to SBC to add this capability are detailed at the end of this proposal.

The set capacity function is activated by a new bit (SC, for Set Capacity) and uses the Logical block address field as the last LBA desired for the drive. The target returns 8 bytes of data, with the Returned logical block address field set to the actual last logical block that is set into the drive. This value will be less than or equal to the requested capacity, depending on the capacity available in the target.

This feature includes an option to split the capacity of a drive into several LUNs. For example, a 10 GB drive can be split into 3 LUNs as follows:

1. Send a READ CAPACITY w/SC option to LUN 0, setting the capacity to 4 GB. This results in a 4 GB LUN 0 and 6 GB on the drive is unassigned.
2. Send a READ CAPACITY w/SC option to LUN 1, setting the capacity to 4 GB. This results in a 4 GB LUN 0, a 4 GB LUN 1, and 2 GB unassigned.
3. Send a READ CAPACITY w/SC option to LUN 2, requesting a capacity of 4 GB. The drive will return a capacity of 2 GB (all of its remaining capacity). This results in a 4 GB LUN 0, 4 GB LUN 1, and 2 GB LUN 2.
4. If a READ CAPACITY w/SC option is sent to LUN 3, the command should end with CHECK status, sense bytes set to LUN not supported (no capacity remains). The capacity of the other LUNs is unchanged.

The SBC description of the READ CAPACITY command should read as described on the next page. Additions for the set capacity feature are in *italics*. Deletions are marked with ~~strike through~~.

### 6.1.7 READ CAPACITY command

The READ CAPACITY command (see table 17) provides a means for the application client to request information regarding the capacity of the logical unit *or to set the capacity of a logical unit*.

**Table 17 - READ CAPACITY command**

Bit Byte	7	6	5	4	3	2	1	0	
0				Operation code (25h)					
1				Reserved				RelAdr	
2	(MSB)								
3				Logical block address					
4									
5								(LSB)	
6				Reserved					
7				Reserved					
8				Reserved				/ SC	PMI
9				Control					

See 6.1.2 for a definition of the RelAdr bit and the logical block address field.

The logical block address shall be zero if the PMI bit *and SC bit are* zero. If the PMI *and SC bits are* zero and the logical block address is not zero, the device server shall return a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code set to ILLEGAL FIELD IN CDB.

*When the A partial medium indicator (PMI) bit and set capacity (SC) bit are* zero, ~~indicates that~~ the returned logical block address and the block length in bytes are those of the last logical block on the logical unit.

A PMI bit of one *and SC bit of zero* indicates that the returned logical block address and block length in bytes are those of the logical block address after which a substantial delay in data transfer will be encountered. This returned logical block address shall be greater than or equal to the logical block address specified by the RelAdr and logical block address fields in the command descriptor block.

**NOTE 6** This function is intended to assist storage management software in determining whether there is sufficient space on the current track, cylinder, etc., to contain a frequently accessed data structure, such as a file directory or file index, without incurring an access delay.

*A PMI bit of zero and SC bit of one (this combination is the set capacity feature) indicate that the logical block address field contains the requested last logical block address for the logical unit. If the LUN is zero, a value of zero in the logical block address field indicates that all other LUNs (if other LUNs are supported) shall become unsupported and LUN 0 shall be assigned all of the available capacity. If the LUN field is non-zero and other LUNs are supported, a value of zero in the logical block address field indicates that the logical unit shall have a capacity of zero and become unsupported. The returned logical block address and block length in bytes are the values for that logical unit after the set capacity function as been performed. This returned logical block address shall be less than or equal to the logical block address specified by the logical block*

address field in the command descriptor block (except when the LUN is zero and the logical block address field is zero). Once GOOD status is returned, the new last logical block address value shall be preserved across power cycles or reset events until superseded by another READ CAPACITY command with SC bit of one.

**NOTE X** If a set capacity operation requests less than the available capacity, the excess capacity is left unassigned to any LUN. This excess capacity is available for assignment in another set capacity operation, however.

**NOTE Y** The set capacity function is intended to assist storage management software by allowing an installation procedure that makes logical units have exactly the same capacity. This characteristic is especially desired for arrays, where a failed unit must be replaced by another unit with identical capacity. It is also useful for splitting very large capacity drives into several logical units. This allows a large drive to stay within the capacity limit for a single drive that may exist in storage management software.

If the PMI and SC bits are both one, the device server shall return a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code set to ILLEGAL FIELD IN CDB.

A reservation conflict shall occur when a READ CAPACITY command is received from an initiator other than the one holding a logical unit reservation. A reservation conflict shall occur when a READ CAPACITY command with set capacity feature is received from an initiator other than one holding any extent reservation. A READ CAPACITY command without set capacity feature shall not be affected by extent reservations.

The READ CAPACITY data (see table 18) shall be sent during the DATA IN phase of the command.

**Table 18 - READ CAPACITY data**

Bit	7	6	5	4	3	2	1	0
Byte								
0	(MSB)			Returned logical block address				
3								(LSB)
4	(MSB)			Block length In bytes				
7								(LSB)