To: INCITS T10 Committee

From: Kevin Butt, IBM

Date: November 22, 2002

Document: T10/03-017

Subject: SSC2: Locate to Filemark

At the November 2002 SSC-2 working group meeting when responding to Letter Ballot comments, it was noted that the Logical File Address and Logical Set Address portions of the Locate(16) command was not clearly specified. Since IBM was the company that requested this capability, I was asked to see if it could be removed. If it can not be removed I was given an action to clearly define Logical File Address and Logical Set Address. I was also given the action to cleary specify this in the Locate (16) command. This document is the response to this action item.

1. Added by 00-318

The Logical File Address and Logical Set Address were added by 00-318 as part of the Explicit address model. The stated goal of 00-318 is to allow a device driver to safely redrive commands, perform alternate pathing when a path has a hard failure, perform command queueing where in-order delivery is not guaranteed, etc. without the need for complicated logic to regain positioning agreements between the Initiator and the Target.

2. Need to Locate to an absolute mark

The implication of the explicit address model is that the device driver is aware of the current logical block and file number. As such, the driver would not want to use a relative positioning command. It is prone to data integrity problems and has caused protocols such as Fibre Channel to add special error recovery procedures to help maintain relative addressing (that is, the current state of the media in the drive).

If the driver is required to continue to do file positioning using relative commands, then one of the fundamental points of the explicit addressing model will have been missed. The explicit addressing model has a goal of no relative positioning requests and requires the following three fundamental points to acheive that goal:

1. Explicit block addressing
2. Explicit file addressing
3. Explicit set addressing

Tape is viewed as being difficult to deal with and has created some mistrust by the outside world because of the necessity to impose additional requirements on a protocol layer (Fibre Channel for instance) beyond that required by disk. As long as there are relative positioning commands, this view by the outside world and the additional requirements being imposed on the transport protocol layer will persist. This is not good for the tape community. It tempts users to skip tape solutions and adopt disk-only archiving techniques. The more mainstream tape is (that is, the fewer additional requirements or special cases), the more likely it can maintain its role as the predominant choice for archiving data.

The stated goal of allowing device drivers to safely redrive commands, etc. is part of the answer to these concerns.

Questions that are frequently asked about tape are:

Is it safe to run tape on iSCSI?

Is it safe to run tape on Infiniband?

It is IBMs position that the tape community needs to work towards a solution where it can be said that tape is as safe as disk becasue there are no additional requirements that need levied on the transport protocol layers.

3. Changes to SSC-2

3.1.23 filemark: A special recorded logical object within a partition, not containing user data, that provides a segmentation scheme for the contents of a partition.

3.1.34 logical block: A logical object that is a unit of data supplied or requested by an application client.

3.1.xx logical file: Zero or more logical blocks and setmarks ending in a filemark.

3.1.35 logical object: A unit of data, either a logical block or a mark.

3.1.36 logical object identifier: a unique identifier, within a partition, for a logical object (see 4.2.9).

3.1.xx logical set: Zero or more logical blocks and filemarks ending in a setmark.

3.1.37 logical file address: A unique identifier for a logical file. The first logical file address in a partition has an address of zero and begins at BOP. If a logical file has an address of n and is not the first or last logical file on media, the logical file immediately before that has an address of n-1 and the logical file immediately after has an address of n+1.

3.1.38 logical set address: A unique identifier for a logical set. The first logical set address in a partition has an address of zero and begins at BOP. If a logical set has an address of n and is not the first or last logical set on media, the logical set immediately before that has an address of n-1 and the logical set immediately after has an address of n+1.

3.1.53 setmark: A special recorded logical object within a partition, not containing user data, that provides a segmentation scheme similar to filemarks.

4.2.x Logical files within a partition

Application clients may use filemarks to separate groups of user data into logical files. Logical files shall contain zero or more logical blocks and setmarks and shall end with a filemark. Logical files shall have a logical file address.

4.2.x Logical file address

The logical file address value shall be a sequentially increasing number assigned to each logical file recorded in the partition starting with zero for the recorded logical file beginning at BOP.

The READ POSITION command may be used to determine a logical file address and the application client may use this value with a LOCATE(16) command to position to the same location at some future time, provided the volume has not been rewritten in the interim.

4.2.x Logical sets within a partition

Application clients may use setmarks to separate groups of user data into logical sets. Logical sets shall contain zero or more logical blocks and filemarks and shall end with a setmark. Logical sets shall have a logical set address.

4.2.x Logical set address

The logical set address value shall be a sequentially increasing number assigned to each logical set recorded in the partition starting with zero for the recorded logical set beginning at BOP.

The READ POSITION command may be used to determine a logical set address and the application client may use this value with a LOCATE(16) command to position to the same location at some future time, provided the volume has not been rewritten in the interim.

In Section 7.3

The DEST_TYPE field shall be used in conjunction with the LOGICAL OBJECT IDENTIFIER field to locate to the appropriate position of the medium. The DEST_TYPE field specifies whether the location specified is a logical object identifier, logical file address, or a logical set address. Upon completion of a LOCATE(16) command the logical position shall be as specified in the Logical position upon completion column in Table 32 regardless of direction. The DEST_TYPE field is defined in Table 32..

TABLE 1. DEST_TYPE field definitions

Code	Description	Logical position upon completion	Support
00b	Logical object	BOP side	M
01b	Logical file address	BOP side of the logical file	M (for explicit address mode) O (for implicit address mode)
10b	Logical set address	BOP side of the logical set	O
11b	Reserved		

A block address mode (BAM) bit of zero specifies the logical unit shall process this command as an implicit address command. A BAM bit of one specifies the logical unit shall process this command as an explicit address command.

The Logical Object Identifier field specifies the logical object identifier to which the logical unit shall position the medium based on the current setting of the Dest_Type field. An otherwise valid LOCATE(16) command to any position between beginning-of-data and the position immediately after the last object in the partition (position at end-of-data) shall not return a sense key of ILLEGAL REQUEST. A LOCATE(16) to a position past end-of-data shall return CHECK CONDITION status and the sense key shall be set to BLANK CHECK. Additionally, the sense data eom bit shall be set to one if end-of-data is located at or after early-warning.

In section 7.5

The service actions defined for the READ POSITION command are shown in table 36.

TABLE 2. Table 36 — READ POSITION service action codes

Code	Name	Description	Implementation Requirements	Reference
00h	SHORT FORM -- BLOCK ID	Device server shall return 20 bytes of data with the First Logical Object Location and Last Logical Object Location fields as logical object identifier values (see 4.2.9), relative to a partition.	M	table 37
01h	SHORT FORM -- VENDOR-SPECIFIC	Device server shall return 20 bytes of data with the First Logical Object Location and Last Logical Object Location fields as vendor specific values.	O	table 37
02h - 05h	Reserved	Illegal request		
06h	LONG FORM	Device server shall return 32 bytes of data.	M (for explicit address mode) O (for implicit address mode)	table 38
07h	Reserved	Illegal request		
08h	EXTENDED FORM	Device server shall return 28 bytes of data up to the maximum length specified by the ALLOCATION LENGTH field.	O	table 39
09h - 1Fh	Reserved	Illegal request		

Below Table 38 — READ POSITION data format, long form.

The File Number field specifies the number of filemarks between beginning-of-partition and the current logical position. This number is the logical file address of the logical file that contains the current logical position.

The Set Number field specifies the number of setmarks between beginning-of-partition and the current logical position. This number is the logical set address of the logical set that contains the current logical position.