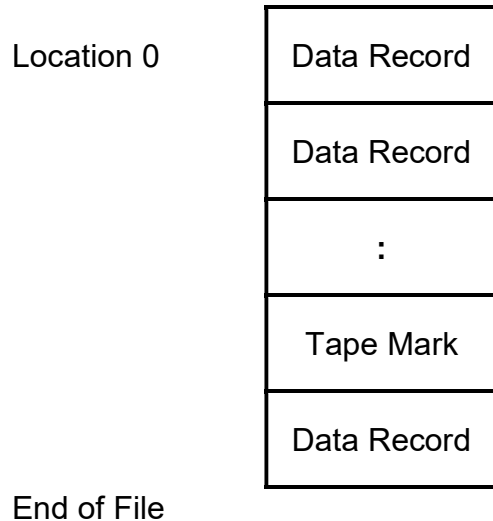


SIMH Magtape Representation and Handling

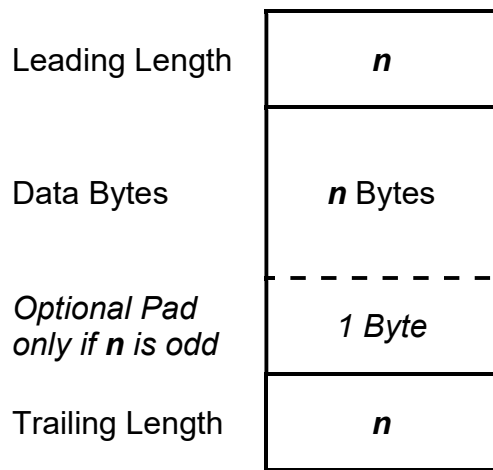
Bob Supnik, 17 Jan 2022

Magtape Representation

SIMH represents magnetic tapes as disk files. Each disk file contains a series of objects. Objects are either metadata markers, like tape mark or end of medium, or they are data records. Location 0 of the file is interpreted as beginning of tape; end of file is interpreted as end of medium. Pictorially:



Metadata markers occupy 4 bytes and are stored in little-endian order. Data records consist of a leading 4-byte record length, an even number of bytes of data, and a trailing 4-byte record length that must be the same as the initial record length. The leading and trailing record lengths allow a record to be accessed either forward or backward. The record length words are stored in little-endian order. A record of length n appears as follows:



If the record length is odd, a pad byte is appended to the data bytes to produce an even number. The extra byte is undefined but should be 0.

Metadata markers and record length words are unsigned integers that are divided into two fields, as follows:

31	30	29	28	27	26	25	24	23	...	4	3	2	1	0
Class Number				Marker Value										
Class Number				Record Length										

Interpretation of the fields depends on the specific SIMH format supported.

SIMH Extended Format

The extended format assigns the sixteen possible record and marker classes as follows:

<i>Class (hex)</i>	<i>Value or Length</i>	<i>Interpretation</i>
0	0	Tape Mark
0	>0	Good Data Record
1-6	any	Private Data Record
7	any	Private Marker
8	0	Bad Data Record, no data recovered
8	>0	Bad Data Record
9-D	any	Reserved Data Record
E	any	Tape Description Data Record
F	any	Reserved Marker

Typically, data records copied from a physical magnetic tape are written as Class 0 (good) data records. If the physical record reported a data error (e.g., a parity or cyclic redundancy check error), it is written as a Class 8 (bad) data record to indicate that the data integrity is in question.

An application may use the private data record and marker classes for any purposes it desires. These classes are permanently assigned to private use; SIMH will not interpret objects of these classes. Unless a simulator explicitly asks to receive one or more private classes, the SIMH tape library will ignore them by skipping over them until a standard data record or marker is encountered.

For example, an application might use a private record to record the recovered parity bits corresponding to a bad data record. Such an application would write one private record after each bad record that would indicate the location of the

parity errors within the immediately preceding data record. Another use might be to write recovered CRC and LRC bytes after each good or bad data record. Still another use might be to pair standard data records with private records containing the corresponding raw NRZI or PE signal streams. Private markers might be used to indicate the original tape density (BPI) or to indicate "erased and not yet written" records within a block-structured tape image, such as those produced by some cartridge tape drives.

The Class E (tape description) record is assigned to descriptive information regarding the tape image. The contents of Class E records are user-defined, and SIMH does not interpret them. In this respect, they are treated as private records, although with a specified purpose. An application might choose to include in such a record one or more text lines describing the tape and an explanation of how any private classes are used within the image. As another example, a Class E record might hold a JPEG photo of the tape label. The internal layouts and interpretations are entirely up to the application.

Classes 9 through D and F are reserved for future assignment. New extensions to the SIMH format will come from allocations within these classes.

Three standard markers are currently defined:

<i>Marker (hex)</i>	<i>Interpretation</i>
00000000	Tape Mark
FFFFFFFFE	Erase Gap
FFFFFFFF	End of Medium

A series of four-byte erase gap markers is used to represent an erased portion of a tape image. The count of markers reflects the physical length of the gap at the assumed density. For example, a three-inch gap at 800 BPI would occupy 2400 bytes on a physical tape. In a tape image representing an 800 BPI tape, 600 four-byte erase markers would be written.

Part of the Class F marker range must be reserved to recognize "half-erase-gap" markers. These arise because data records occupy a multiple of two bytes, while markers occupy four bytes. If a data record that overwrites a longer erase gap occupies a multiple of four bytes, then it would overlay an integral number of erase gap markers. Interpretation in this case is straightforward, as the first object following the trailing data record length word would be the defined erase gap marker.

However, if the overwriting data record occupies only a multiple of two bytes, then it will overlay the first two bytes of the four-byte erase gap marker that follows the trailing record length word. A forward read of the image after the data record retrieves the four-byte sequence FF FF FE FF because it reads half of the overwritten marker and half of the following full marker. This special value,

FFFEFFFF (hex), is recognized as a "half-gap" marker, and reading is realigned by backing up the resulting file position by two bytes. A forward read then continues with the first full erase gap marker of the remaining gap.

When reading in reverse, the problem is more difficult. The four-byte value preceding the full marker that begins the gap consists of half of the overwritten gap marker (FF FF) and half of the four-byte length word of the preceding data record. The difficulty is that "the half-gap marker" is actually a range of Class F marker values. They all start with FF FF—the truncated half of the overlaid erase gap marker—but the following two bytes from the upper part of the length word may assume any value from 00 00 through FF FD (the values FF FE and FF FF are disallowed, as otherwise the marker would have the same value as a full erase gap or EOM marker). A value in this range is recognized as a half-gap marker, and realignment is done by backing up the file position by two bytes to point at the first byte of the erase gap. A reverse read then continues with the data record by retrieving the complete four-byte trailing data record length word.

An end-of-medium marker is used to indicate the logical end of a tape. The physical end of a tape image file serves the same purpose, so an EOM marker is redundant if it is placed as the last object in a file. It is typically used to shorten an image logically without physically truncating the image file and is equivalent to writing erase gaps from the EOM point through the physical end of the file. The SIMH tape library will not read or position past an EOM marker, although an image may be extended by overwriting the marker.

The complete Class F range assignments are as follows:

<i>Marker (hex)</i>	<i>Interpretation</i>
F0000000 – FFFDFFFF	Available for future marker assignments
FFFE0000 – FFFEFFFF	Illegal (would be seen as full gap in reverse reads)
FFFEFFFF	Interpret as half-gap in forward reads
FFFF0000 – FFFFFFFD	Interpret as half-gap in reverse reads
FFFFFFFE	Erase Gap
FFFFFFFF	End of Medium

A conforming writer will never write the illegal marker values, and a conforming reader will recognize the half-gap marker values and resynchronize as described above.

SIMH Standard Format

The standard format is a subset of SIMH Extended format. Data records are restricted to Class 0 (good) and Class 8 (bad), and the record length is restricted

to 24 bits (16 MB). Metadata markers are restricted to tape mark, erase gap, and end of medium.

A SIMH-format file may contain any of the extended-format objects, but a reader conforming to the standard format will recognize only those listed here. All other objects will be ignored.

Magtape Operations

Magnetic tape drives can perform the following operations:

- Read forward
- Read backward
- Write forward
- Space forward record(s)
- Space backward record(s)
- Write tape mark
- Security erase
- Write erase gap
- Write private marker

On a real magtape, all operations are implicitly sequential, that is, they start from the current position of the tape medium. SIMH implements this with the concept of the *current tape position*, kept in the pos field of the tape drive's UNIT structure. SIMH starts all magtape operations at the current position and updates the current position to reflect the results of the operation:

- Read forward. Starting at the current position, read the next 4 bytes from the file, skipping any intervening gap and unrecognized marker or record classes. If those bytes are a valid record length, read the data record and position the tape past the trailing record length. If they are a tape mark, signal tape mark and position the tape past the tape mark. If they are end of medium, or an end of file occurs, signal no more data ('long gap' or 'bad tape') and do not change the tape position.
- Read reverse. If the current position is beginning of tape, signal BOT. Otherwise, starting at the current position, read the preceding 4 bytes from the file, skipping any intervening gap and unrecognized marker or record classes. If those bytes are a valid record length, read the data record and position the tape before the initial record length. If they are a tape mark, signal tape mark and position the tape before the tape mark. If they are end of medium, or an end of file occurs, signal no more data ('long gap' or 'bad tape') and position the tape before the end of medium marker.
- Write. Starting at the current position, write the initial record length, followed by the data record, followed by the trailing record length. Position the tape after the trailing record length.

- Space forward record(s). Starting at the current position, read the next 4 bytes from the file, skipping any intervening gap and unrecognized marker or record classes. If those bytes are a valid record length, position the tape past the trailing record length and continue until operation count exhausted or metadata encountered. If those bytes are a tape mark, signal tape mark and position the tape after the tape mark. If they are end of medium, or an end of file occurs, signal no more data ('long gap' or 'bad tape') and do not change the tape position.
- Space reverse record(s). If the current position is beginning of tape, signal BOT. Otherwise, starting at the current position, read the preceding 4 bytes from the file, skipping any intervening gap and unrecognized marker or record classes. If those bytes are a valid record length, position the tape before the initial record length and continue until operation count exhausted, BOT, or metadata encountered. If they are a tape mark, signal tape mark and position the tape before the tape mark. If they are end of medium, or an end of file occurs, signal no more data ('long gap' or 'bad tape') and position the tape before the end of medium marker.
- Write tape mark. Starting at the current position, write a tape mark marker. Position the tape beyond the new tape mark.
- Security erase. Starting at the current position, write an end of medium marker. Do not update the tape position.
- Write erase gap. Starting at the current position, erase the amount of tape indicated by the specified length and bpi or by the specified number of bytes. If the end of the gap overwrites an existing record, shorten that record appropriately. Position the tape after the gap.
- Write private marker. Starting at the current position, write a private marker. Position the tape beyond the new marker.

Magtape Error Handling

The following matrix defines error responses versus events for simulated magtapes. PNU signifies position not updated; PU signifies position updated.

	Unit not attached	Tape mark	End of medium mark	Write locked	End of attached file	Data read or write error
Read forward	Error: unit not ready, PNU	Error: tape mark, PU	Error: bad tape or runaway tape, PNU	ok	Error: bad tape or runaway tape, PNU	Error: parity or data, PU if gap precedes error, else PNU
Read reverse	Error: unit not ready, PNU	Error: tape mark, PU	Error: bad or runaway tape, PU	ok	Error: bad or runaway tape, PU	Error: parity or data, PU if gap precedes error, else PNU
Write forward	Error: unit not ready, PNU	na	na	Error: unit write locked, PNU	na	Error: parity or data, PNU
Space records forward	Error: unit not ready, PNU	Error: tape mark, PU	Error: bad or runaway tape, PNU	ok	Error: bad or runaway tape, PNU	Error: parity or data, PU if gap precedes error, else PNU
Space records reverse	Error: unit not ready, PNU	ok	Error: bad or runaway tape, PU	ok	Error: bad or runaway tape, PU	Error: parity or data, PU if gap precedes error, else PNU
Write tape mark	Error: unit not ready, PNU	na	na	Error: unit write locked, PNU	na	Error: parity or data, PNU
Erase	Error: unit not ready, PNU	na	na	Error: unit write locked, PNU	na	Error: parity or data, PNU
Write gap	Error: unit not ready, PNU	na	na	Error: unit write locked, PNU	na	Error: parity or data, PNU
Write marker	Error: unit not ready, PNU	na	na	Error: unit write locked, PNU	na	Error: parity or data, PNU

The behavior of simulated tapes mirrors that of real tapes, except for errors that make determination of the record length impossible. On a real tape, a read or write error would update the position of the tape. On a simulated tape, this isn't possible; the length of the record is unknown. Real tape drivers would try to recover from the error by backspacing over the erroneous record and trying again. This won't work on a simulated tape.

For intelligent tapes, like the TK50 and the TS11, this problem is handled by reporting "position lost". This status tells the tape driver that tape position is no longer known, and normal error recovery isn't possible. Older tapes do not have this status. Accordingly, these tapes implement a limited form of state "memory" for error recovery. If an error occurs on a forward operation, and the position is not updated, the simulated tape unit "remembers" this fact. If the next operation is a backspace record, the first backspace is skipped, because the simulated tape is still positioned at the start of the erroneous record. If a read is then attempted, the tape will read the record that caused the original error.

A corresponding error recovery method is used for reverse reads immediately followed by forward spacing. The spacing operation is suppressed, so that a reposition-and-retry recovery operation accesses the erroneous record again.

Magtape Emulation Library

SIMH provides a support library, `sim_tape.c` (and its header file `sim_tape.h`), that implements the standard tape format and functions. The library is described in detail in the associated document, "Writing A Simulator For The SIMH System".