

LIF Implementation

Definitions

This section is included to help you understand some of the terms used in LIF.

Cylinder: A cylinder number denotes the position of an actuator when more than one detector is positioned by the same actuator mechanism.

Cylinder Mode Addressing: This is used where there are several surfaces on a particular medium. It would normally require three parameters to find a particular piece of data in this case; namely: cylinder number, surface (side) number, and sector number. Cylinder mode addressing sequentializes the medium so that any location can be defined by a single parameter. This mode is defined as follows: the absolute sector following the last sector on cylinder 0, surface 0, is taken to be the first sector on cylinder 0, surface 1, rather than the first sector on cylinder 1, surface 0.

Directory: A file that is found in a known location and contains a list of user files on the medium. The directory also contains the following information about each file: starting location of the file, length of the file, and type of data in the file.

Double Word Integer: A 32-bit signed binary integer. For this application, a double word integer is considered to be packed into two 16-bit words so that the most significant bits are in the first word, and the least significant bits are in the second word.

File: A user defined collection of logically contiguous records written onto the data portion of each sector and containing a set of user defined data.

Integer: A 16-bit signed binary number.

Interchangeability: The ability to read information from a medium on a host computer other than the one that originally wrote the information.

Sector: Used interchangeably with PHYSICAL RECORD. A grouping of contiguously recorded bits. This includes a header made up of a series of bits that recognizes and synchronizes the start of the sector and the body of the sector. The body contains the user information and certain device applications such as checksum and error correcting code (ECC). All references to sector location are in the linear list mode.

Track: The area of the medium that can be read by a single detector (head) without movement of that detector.

Unit of Addressing: This is a blocking factor that establishes valid start addresses. The LIF addressing unit is always 256 bytes.

Volume Label: Also referred to as "Directory Header", this is a file created on initialization of the medium at a known location (usually the first sector) that contains information about the medium. This information includes things such as type of media, and start and length of the directory.

Volume Number: In certain instances, a file may be placed on several different media. This usually happens because the size of the file is greater than the capacity of the medium. In this case, the volume number is an indication of which portion of a file is on this medium. Note: the Model 64000 does not support multivolume files.

Logical Media Layout

Figure A-1 shows the linear layout of all disc and preformatted tape media.

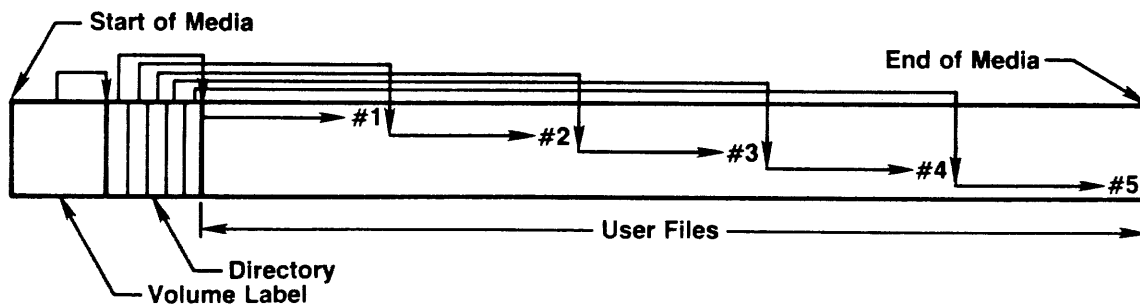


Figure A-1. Linear Layout For Disc and Preformatted Tape Media

Volume Label

The purpose of the volume label is to allow easy identification of the media type and to find file information (directory). The volume label is two sectors long and its format is as follows.

WORD (16 BITS)	CONTENTS
0	100000 OCTAL (LIF ID)
1	VOLUME LABEL (0-6 ASCII CHARACTERS)
2	
3	
4	DIRECTORY START ADDRESS
5	
6	OCTAL 10000 NEEDED FOR SYSTEM 3000
7	DUMMY (SET TO 0)
8	LENGTH OF DIRECTORY (FIXED AT INITIALIZATION)
9	
10	VERSION NUMBER (0 FOR MEDIA WITH NO EXTENSIONS)
11	SET TO 0
12-20	LEVEL 1 EXTENSIONS-SEE EXTENSIONS SECTION
21-126	RESERVED FOR EXTENSIONS AND FUTURE USE (SET TO 0)
127	RESERVED BY SYSTEM 250 FOR MEDIA MAINTENANCE WORD
128-255	SET TO 0

Location

The location of the volume label is defined to be the first unit of addressing available on the medium. All words in the second addressing block are to be initialized to 0 for compatibility with the System 3000.

LIF Id

This identifies media written in the logical interchange format specified by this standard.

Volume Label

Characters (0-6 ASCII) are packed with the first character of a pair in the high order byte. Trailing characters are spaces. To provide maximum interchange, characters are limited to upper case letters (A-Z), digits (0-9) and the underscore (_) character. The first character (if any) must be a letter. The purpose of this field is to identify the particular volume in a drive. Default volume label is six spaces.

Directory Start Address

This is a double word integer showing the address for the start of the directory in number of addressing units. The first word contains the high order bits and the second word contains the low order bits. The value of this field is two at the present time.

Word 6

This word is included to eliminate console messages on the System 3000. It must be written as shown.

Length of Directory

This double word integer entry is included to stop the directory from overwriting user storage space. This entry is decided on at disc initialization, thereby fixing directory size at that time. This entry contains the maximum allowable length of the directory in number of addressing units. Note, the length of the directory on discs produced by the Model 64000 is 14 sectors.

Version Number

This field is used to allow for implementation of extensions to the standard. It should be set to 0 for a base level implementation. For a list of capabilities added with each version, see the section on extensions that follows later. Note, the Model 64000 implements level 1 extensions. Discs formatted by the Model 64000 will have this field set to 1.

Words 21-126

These words are reserved for extension fields and for implementation dependent fields. See the later section on extensions for the currently implemented extensions.

Word 127

This word is reserved for the use of System 250 as a maintenance word. Tapes received from other machines must have this word set to 0.

Directory

The purpose of the directory is to allow data to be easily located. All information necessary for reading a file is contained in the directory, which is a linear list of directory entries each organized as follows.

WORD (16 BITS)	CONTENTS
0	FILE NAME (1-10 ASCII CHARS, TRAILING BLANKS)
1	
2	
3	
4	FILE TYPE
5	
6	STARTING ADDRESS
7	
8	LENGTH OF FILE
9	
10	TIME OF CREATION (12 BCD DIGITS)
11	
12	
13	L VOLUME NUMBER
14	IMPLEMENTATION
15	

Location

The directory is located starting on an addressing block boundary defined by the Directory Starting Location Field of the volume label. Directory location is further limited to the logical front of the medium. Therefore, user space cannot span the directory.

File Name

Characters (1-10 ASCII) are packed with the first byte of a pair in the most significant byte of the word. Characters are limited to digits (0-9), upper case letters (A-Z), and the underscore (_) character. The first character must be a letter. All nonpurged files must have at least one character in their file name and implementations must be capable of distinguishing all 10 characters. Padding is with trailing blanks.

Length

Length is a 16-bit integer showing the length of the record in bytes, but not including the length field in the count. A length of -1 will denote logical end of file. A file will have an end of file marker unless its length is at the maximum length defined in the directory. Also, the length field of the last record may exceed the physical space remaining; in this case, the file is terminated by physical length of file. If the last volume flag is set for the file, this is the end of file mark; otherwise, it is the end of the volume. A "0" length record is valid. In this case, the length field is followed by the length field for the next record. A record of length less than -1 is not allowed. Finally, if the length is odd, the record is rounded to the next word boundary and the extra byte is ignored.

The Model 64000 file system does not support 0 length records, records containing an odd number of bytes, or records longer than 256 characters. If one of these records is encountered, when reading an LIF file produced on another system, the Model 64000 reacts as follows.

A 0 length record is made into a record containing 2 space characters.

A record containing an odd number of characters is padded with a trailing space character to make an even number of characters.

LIF records longer than 256 bytes are divided into 2 or more records on the Model 64000. The division is made at 256 byte boundaries.

Data

File type 1: The record consists exclusively of 8-bit ASCII codes. All 8 bits are defined (i.e., foreign character sets) and no parity is checked or generated. Codes 0 through 127 have US ASCII meaning. Codes 128 to 255 are open. In addition, no FORTRAN carriage control, escape sequences, etc. are defined. Numeric data are formatted per ANSI standard X3.42, which is available from ANSI. Data records should be stored with one item per record. So an "n" element array would require n logical records for storage. Programs are stored with one source line per record. Records must not contain a trailing CR-LF unless it is a meaningful portion of the data item. Logical records are allowed to span physical record boundaries.

File type -2: The data format within the record is undefined and must be known to the user.

Extensions

An extension field is one that is defined to be of general interest to a large set of implementors, but is not required for implementation of the standard. No extension is allowed that would prohibit an implementation from reading, writing or creating a lower level implementation. Levels are hierarchical; that is, level N includes levels 0 to N-1.

File Type

This is a 16-bit signed integer. The presently defined file types follow.

0: Purged file - This standard makes no statement on directory names of purged files. However, the purged file name must be capable of being identical with a valid file. Implementation of this standard must not give duplicate file name errors for purged files.

1: ASCII data file - This is the interchange file type. The associated file consists of 8-bit ASCII characters. See the next section, File Structure, for a complete description.

-1: Logical end of directory - See Directory Organization.

-2: Binary data file - See File Structure, which follows later, for details.

Other negative file types are reserved for system dependent applications. It is not necessary for any implementation to be able to read these file types for conformance. They are merely for flexibility and convenience and to be used by the experienced user.

Starting Address

The starting unit-of-addressing number for a file is a double-word integer. The most significant bits are in word 6 and the least significant bits are in word 7.

Length of File

A double-word integer shows the allocated space for the file (not current length) on this volume. Again, the most significant word comes first. The length is given as the number of 256-byte blocks. Lengths less than 0 are not allowed. The usage of free space is implementation dependent. Therefore, the length and start address fields of a purged file are not guaranteed to be accurate. Free space is computed from the start address and length fields of the two nearest valid files that surround the free space.

Time of Creation

This field is 12 BCD digits of the form YYMMDDHHMMSS. These words will be packed with the first digit in the most significant bits of word 9 and the last digit in the least significant bits of word 11. This field can be used for a version number on systems not using a real-time clock. If the year and month fields are 0, the other fields will be a version number. However, all digits must be valid BCD numbers. Also, if a particular file spans several volumes, the directory entries for each volume of this file must have the same time stamp or version number.

Last Volume Flag

If the last volume flag (L) = 0, it is not the last volume of the file. If the last volume flag (L) = 1, it is the last volume of the file.

Volume Number

This is a 14-bit unsigned integer containing the volume number of this file on this medium. Volume number = 0 is not a valid condition. All files must be contained on at least one volume. Volumes start with number 1 and are incremented by one for each subsequent volume.

Implementation

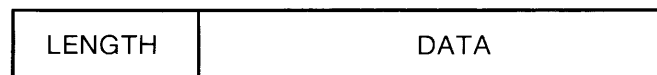
Words 14 and 15 are available for implementation dependent uses by file types -2000 to -77777. These fields must be set to 0 for all interchange file types and for file types -2 to -177.

Directory Organization

The directory is defined to be a linear list of directory entries. It has no fixed length, except as fixed at the time of initialization. A directory entry that has a file type of -1, is defined to be the end of logical directory. A logical end of directory mark is written unless the directory is filled. The physical end of directory is determined by adding the start of directory and length of directory fields. This address is considered an absolute end of directory indicator and thus precludes the need for a logical end of directory marker on full directories. Free space begins at Address # = (Starting Address # of last file entry + last file length). Implementations must also be aware of the absolute end of directory found in the volume label. Methods of packing and reallocating free space are not defined. However, entries must be stored so they are in order of strictly increasing starting addresses. Directory entries are undefined after the logical end of directory. Thus, when a file is appended, the next directory entry must be set to the logical end of the directory.

File Structure

A file is a list of records containing data. Defining the structure for file types 1 and -2 provides a mechanism for transferring files containing virtually any type of data. The file does not attempt to convey the meaning of the data as part of the file. This is best done by external means (naming conventions, attached hard-copy description etc.). A file consists of a linear list of records; explanations follow.



Location

A file location is determined by the value of the starting address field within the directory entry. A file always starts on an addressing block boundary. Further, a logical record will not traverse a volume. Thus, all volumes will start with a complete logical record making it possible to recover partial data.

Level 0 is defined to be the level with no extensions. This level must be supported for all implementations. The Model 64000 implements level 1 extensions when formatting LIF discs. The value of words 12-13 is 33, words 14-15 have a value of 2, and words 16-17 have a value of 16. Words 18-20 contain the date and time the volume was formatted. The following paragraphs explain the words in detail.

Words 12-17

These words contain physical attributes data. Words 12 and 13 contain the number of tracks per surface with the most significant word being word 12. Words 14 and 15 contain the number of surfaces per medium, with the most significant word being word 14. Words 16 and 17 contain the number of sectors per track with the most significant word being word 16.

Words 18-20

These words contain the volume stamp. This field is identical to the time stamp field of the directory listing, except this field reflects volume creation. All comments about valid BCD digits, version numbers, etc. still apply.

Flexible Disc Track Format

Table A-1 describes the physical track format for HP 5-1/4 inch minifloppies.