DATA FORMATS ON MAGNETIC TAPE.

All the standard magnetic tape software routines use a common data format to allow for data interchange between ALGOL, SIR and FORTRAN programs.

Blocks.

Blocks are the basic unit of data organization. Odd parity is used by all the standard software routines. A block consists of an integral number of 18 bit words, the first two of which are reserved for use by the software. Blocks may be of any length from 5 to 2047 words. A block must contain some non-zero words.

Blocks are written in sequence along a tape, starting from the load point. A block should only be written if it follows the load point or another block just read or written, or if it replaces a block just backspaced over. Any information beyond the block last written should be considered lost and an attempt to read past this block may lead to errors.

Block numbers.

When a block is written a block number is inserted in the first work and a block length (in words), plus check and indicator bits in the second. The first (header) block on a tape is numbered +1, the second block +2 and so on to the last block +N.

The second word is used thus:

Bit 18: =1 if the block is a header or label
=0 if the block is data.

Bit 17-13 contain a check character formed from the block number. The read routines checks this is a valid one written by standard software. On exit from a read routine, the copy of the block in store has these bits set to zero.

Bits 12-1 contain the block length as an integer in the range 0-2047. The length is the total number

of words in the block including the first two reserved words.

Layout of blocks on tape.

Block 1, following the load point, must be a header block written by an open file or initialization routine. Before a tape is unloaded, the last significant block on the tape should be an EOF block written by a close file routine. Between these two blocks there may be any number of blocks (or no blocks) written by the user. These intervening blocks may be in any format specified by the user and may include label blocks to separate different sections of data.

Files.

The term file is used to describe a quantity of data contained in a sequence of blocks on tape. It is convenient to have just one file per reel for the majority of applications. The header block contains the file name. Files may extend over any integral number of reels, but each file must start at the beginning of a reel. Reel numbering is the user's responsibility. Multiple files on a tape are not supported, but a file can be divided into sub-files using label blocks.

Records.

The SIR standard routines allow the use of records in addition to blocks. A record consists of an integral number of words, of length 2 or more. Records are backed into blocks and can be either 'fixed-length' or 'variable-length'. If 'fixed-length' all records in a file (or sub-file) must be of the same length. If 'variable-length', the records may be of any length, but a word is added to the beginning of each such record containing an integer giving its length.

The first word of a fixed length record should not be zero. For efficient use of tape with fixed length records, the number of words in a block should be an exact multiple of the number of words in a record (plus the two reserved words).

A file may consist of sub-file with data in block units of various kinds, with fixed or variable length records, but in

this case there must be label blocks separating each sub-file from its successor.

Header Blocks.

All the standard routines require that a header block of the correct format be written on a tape before it can be used. The only exception is MTINIT the tape initialization utility. A program can only obtain access to a file if the name of that file, held in the header block, is supplied by the user.

Header block layout.

Word number	Contents
0;	Block number
1;	/0 n (n = number of words in block)
2 to 5;	Name of file as 12 characters in SIR internal code, padded with zeros if not 12 characters long.
6;	Serial number.
7;	Generation number.
8;	Reel number for multi-reel file, 0 otherwise.
9;	Creation date (0 if none).
10;	Expiry date (0 if none).
11-14;	Reserved.
15;	=0 no record structure <0 variable length records in use >0 fixed records in length (and value is size of the records in words.
16;	Reserved.
17;	<0 file can only be opened for reading or updating, it cannot be opened for writing.
18-27;	owner's optional information. If word 18 is zero, the other words are undefined.

Dates are encoded in the form:

Bits 16-10:	Year number - 1900
Bits 9-6:	Month number
Bits 5-1:	Day number in month.

Label Blocks.

Label blocks are used to divide a file on tape into sub-files. Label blocks give information about the blocks following. Header blocks and end of tape blocks are special cases of a label block. A label block is indicated by the second word being negative (i.e., bit 18 set).

End of Tape Blocks.

The are written when required by the standard close routines. They have the following format:

0;	Block number.
1;	/0 n (n = number of words in block).
2;	Packed characters E, O, F (as SIR internal code)
3-5;	Reserved.

User Labels.

The user may introduce label blocks into files using the standard write routines and keeping to the following format:

0;	Block number.
1;	/0 n (n = number of words in block)
2-3;	Any alphanumeric name in SIR internal code.
4-27;	Any information inserted by the user in any format.

The name must start with a letter (making word 2 negative), and must not start with the letters EO.

Data labels.

Data labels are a special type of label block that may be used by standard software routines and utilities. They are distinguished by having a positive data type number in word 2 and a data name as six alphanumeric characters in words 3 and 4. The format of words 5 onwards is dependent on the data type.

The following data type numbers are allocated:

+1	Program dump
+5	Mnemonic program.
+6	Fixed length records

+7 Variable length records.

For data type 6, word 5; holds the number of words in a record as a positive integer.

Action on Detecting a Label.

In general a program reading a data file should detect the presence of a label by inspecting word 1; of the block in store. If a label is detected the program should check whether its type is of interest at that stage. If it is not blocks should be read and ignored until the next label is detected. If the label is an EOF indicator no more blocks should be read.

File Table Format.

The standard library routines require a data structure known as a file table to be set up for each handler in use. Most of the file table is only referred to on opening a file. Many items of information are not used by the standard software and the user can use them for his own purposes or ignore them. If an item is not used it should be set to zero.

The format is as follows:

0;	Handler number to be used (0-3).
1;	Address in store of the buffer from and to which data are read and written. The first two words of the buffer are reserved for the block number and length, set by software.
2;	The length of the buffer in the range 5-2047 giving the actual length of a block to be written or the maximum length of a block that can be read.
3-4;	Reserved.
5;	Address/marker for record handling.
6-9;	Reserved (affected when file is opened).
10-35;	Correspond exactly to word 2; to 27; of the header block format. When opened for writing the header block is copied from this position. When opened for reading the header block on tape is validated against the file name in this position.

File Protection.

The standard routines only allow programs to read or write on a file that has been 'opened', an operation that checks the identity of the file. An error is generated if a program attempts to access a file that has not been opened. The open file routines will only work if the name in the file header block matches that in the user's file table, or if the tape file name is 'SCRATCH'.