TABLE OF CONTENTS

6000/7000 FILE FORMAT STUDY

The intent of this document is to describe the tape file formats supported by the 6000 SCOPE 3.2, 3.3, 3.4; 7000 SCOPE 1.1 (operator's station), and 7000 SCOPE 2.0 (operator station, magnetic tape station and on-line) operating systems. Additional information has been included to describe label formats, tape structures, label utility programs, and end point processing. The information assembled in this document was taken in part from the 6000 SCOPE 3.2 Reference Manual, the 6000 SCOPE 3.3 ERS and Reference Manual, 7000 SCOPE V1 ERS, 7000 SCOPE V2 ERS, and notes taken from discussions with individuals from the respective development departments. Some of the material contained in this document is subject to change and modification as a function of future development plans. The content of this document is intended to be as accurate and as up-to-date as possible at the time of its generation.

1.0 DEFINITIONS

Section 1.0 defines or gives reference to many of the terms and acronyms used in this document. For consistency and continuity, the terminology as defined below applies throughout the entire document.

Record - A record is a collection of related items of data, which for operating system logic purposes is treated as a unit of information. The delineation of a record may be relatively arbitrary, and is determined by the designer of the information formats. Record and "logical" record are synonymous terms.

Block - A block is a group of contiguous characters recorded on and read from magnetic tape as a unit, i.e., the data contained between 2 successive inter-record gaps.

File - A file is a collection of information, consisting of the records pertaining to a general subject. The delineation of a file may be relatively arbitrary.

Volume - A volume is a physical unit of storage media. The term "volume" is synonymous with "reel of magnetic tape".

Label - A label is a block at the beginning or end of a volume or a file, which serves to identify and/or delimit that volume or file.

Label Group - A label group is a collection of contiguous labels and associated tape marks pertaining to a file that precede or follow that file (or part of that file) on one volume.

Tape Mark - A tape mark is a special hardware bit configuration recorded on magnetic tape. A tape mark indicates the boundary between files and labels, and also between certain label groups.

Character - A character is a unit of information which is either 6 or 8 bits in length.

Physical Record - Continuous data on tape between 2 successive inter-record gaps; physical record, physical block and block are interchangeable terms. A physical record may contain one or more logical records or may constitute only a portion of a logical record.

Block - The term "block" is synonymous to physical record.

Physical Record Unit (PRU) - A PRU is a physical record of the maximum size allowed for that format. A short PRU is a physical record of size less than the maximum. A zero length PRU is a special physical record containing no data; it is 8 characters in length and contains a level number, if appropriate. Zero length PRU's appear only on SCOPE standard tapes, on binary X-mode tapes, and 7000 Format tapes.

PRU sizes	SCOPE Standard	Binary	5120 characters
		Coded	1280 characters
	X tapes	Binary	5120 characters
		Coded	136 characters
	S tapes		IP.NOISE to 5120 characters
	L tapes		IP.NOISE to n characters (n is size of user buffer)
	7000 Format		5120 characters

Level Number Appendage - A level number appendage is the eight characters of added information which are appended to the last

physical record of every logical record on a standard SCOPE tape file (coded and binary). The low order 4 bits of the 8 characters is a binary number ranging from 0 -- 17_8.

These 4 binary bits are refered to as the level number. The level number allows the user to associate a hierarchy indentification with each logical record. The format of the 8 character level number as it appears on the tape is:

Coded tape - 20202020202020XX - For level 0, XX=20;
For levels 1 - 17_8, XX=01-17_8 respectively.

Binary tape - Under 6SC3.2 - 0000000000000YY
For levels 0 - 17_8, YY=0 - 17_8 respectively.
Under 6SC3.3, 6SC3.4 and 7SC2.0
552335522754ZZ
For levels 0 - 17_8, ZZ=0 - 17_8 respectively.

End of Line Terminator - An end of line terminator is a 12 bit binary zero byte in the low order byte of a 60 bit word. End of line terminator is applicable to coded standard SCOPE, coded external and Z record formats.

Staged Tape File - A staged tape file is a tape file which is copied from magnetic tape to the system mass storage device or copied from the system mass storage device to magnetic tape. The user accesses the file from the system mass storage copy when needed.

On-Line Tape File - An on-line tape file is a file which resides on magnetic tape and is accessed directly from the tape unit. i.e., not staged.

CMR - Central Memory Resident - Low core system tables under 6000 SCOPE.

EOR - End of Logical Record.

EOF - End of Logical (and possibly physical) File.

EOI - End of Information.

FIT - File Information Table - The name of a parameter list which defines many of the characteristics of a file under 7000 SCOPE V2 and 6000 SCOPE 3.4 when using 6RM. See 7000 SCOPE V2 ERS and 6RM ERS.

FET - File Environment Table - The name of a parameter list which defines many of the characteristics of a file under 6000 SCOPE and 7000 SCOPE 1.1. See 6000 SCOPE or 7000 SCOPE Reference Manual, respectively.

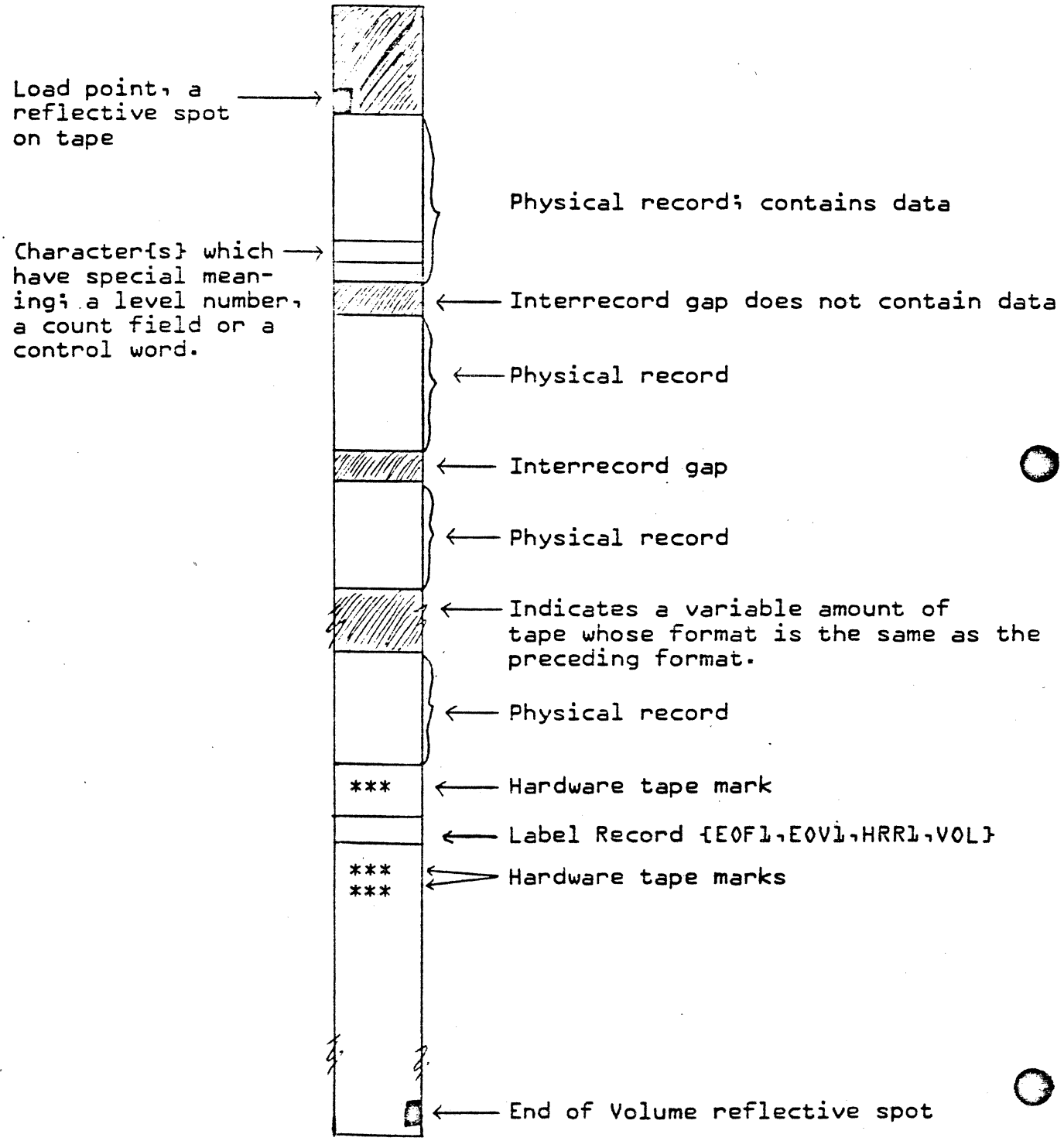
6RM - 6000 SCOPE 3.4 Record Manager.

RCW - Recovery Control Word - A 60 bit control word which 7000 SCOPE V2 inserts into the appropriate place of the mass storage image of a tape file to indicate an inter-record gap. For a detailed description of the RCW format, see Appendix D.

WCW - W record type control word - A 60 bit control word which prefixes every W type logical record. For a detailed description of the WCW format, see Appendix D.

ICW - Type I blocking control word - A 60 bit control word which appears in the first word of every I block, refer to Appendix D.

The following notation is used throughout the document to illustrate tape formats.



2.0 TAPE FILE RECORD FORMATS

A variety of record formats are supported by the five operating systems being considered (6000 SCOPE 3.2, 3.3, 3.4; 7000 SCOPE V1.1; 7000 SCOPE V2). Section 2.0 describes each of these record formats. To be consistent with existing concepts, paragraphs 2.1 thru 2.5 describe the format of the logical record and its associated file. Since the record types described in paragraphs 2.6 thru 2.14 are to a high degree independent of physical record type, very little, if any, mention is made of a file format.

2.1 SCOPE Standard Format

SCOPE standard format is the 6000 SCOPE 3.2, 3.3, and 3.4 system default format (i.e. unless another format is specified, SCOPE standard is assumed). A logical record is defined to be one or more PRU's terminated by a short (or zero length) PRU containing an 8 character level number appendage. A PRU is 5120 characters in binary mode or 1280 characters in coded mode. The end of the logical record is indicated by a short or zero length PRU. A short PRU is $10*n + 8$ characters where $0 < n \leq 511$ in binary mode or $0 < n \leq 127$ in coded mode; the last 8 characters containing the logical record level number is not passed to or from the user's CM buffer. Whenever a logical record is an integral number of whole PRU's (5120 characters for binary, 1280 characters for coded) a zero length PRU is written to indicate end of logical record.

The zero length PRU is only 8 characters in length, the 8 characters contain the level number. All physical records (and therefore logical records) are multiples of 10 characters in length, excluding the level number appendage. A binary SCOPE standard file is recorded in odd parity, a coded file in even parity. SCOPE standard tape files may be labeled or unlabeled. A group of logical records may be delimited by a zero length PRU containing a level number of 17_8. A level number of 17_8 may be written with the 6000 WRITEF or 7000 2.0 ENDFILE macro. Whenever a level number of 17_8 is detected when reading a SCOPE standard tape file, an EOF status is returned through the FET/FIT. Under 6SC3.2, a level number of 17_8 may appear in a physical record which contains data or in a zero length physical record (no data, only 8 characters). Under 6SC3.3, 6SC3.4 and 7SC2.0, a level number of 17_8 always appears in a zero length physical record. Coded SCOPE standard tape files may contain end of line terminators (12 bits of zero in the low order positions of a 60 bit word). On output, the 12 zero bits are converted such that they appear as 1632_8 in XBCD on the tape. On input, an external BCD code of 1632_8 in a multiple of the 5th byte is converted to 0000_8. This line terminator is used by such routines as FORTRAN, COBOL, UPDATE, EDITLIB, to delimit line images.

Figures 1 and 2 illustrate the format of a SCOPE standard tape file. Notice that the VOL1 label, HDR1 label and first hardware tapemark do not appear on unlabeled SCOPE standard tape files. However, the EOF1 trailer label is present on both the labeled and unlabeled SCOPE standard tape files.

In these illustrations, the first logical record is greater than one PRU in length. The first logical record is composed of a full PRU and a short PRU which contains the level number appendage. The second logical record, which is a multiple of the PRU size (2 PRU's), is written as 2 full PRU's followed by a zero length PRU containing the level number appendage. The third logical record, which is less than a full PRU in length, is written as a single short PRU including the level number appendage. The file is terminated by a hardware tapemark, an EOF1 trailer label and 2 more hardware tapemarks. This combination of tapemarks and EOF1 label defines end-of-information.

SCOPE STANDARD TAPE FORMAT (BINARY)

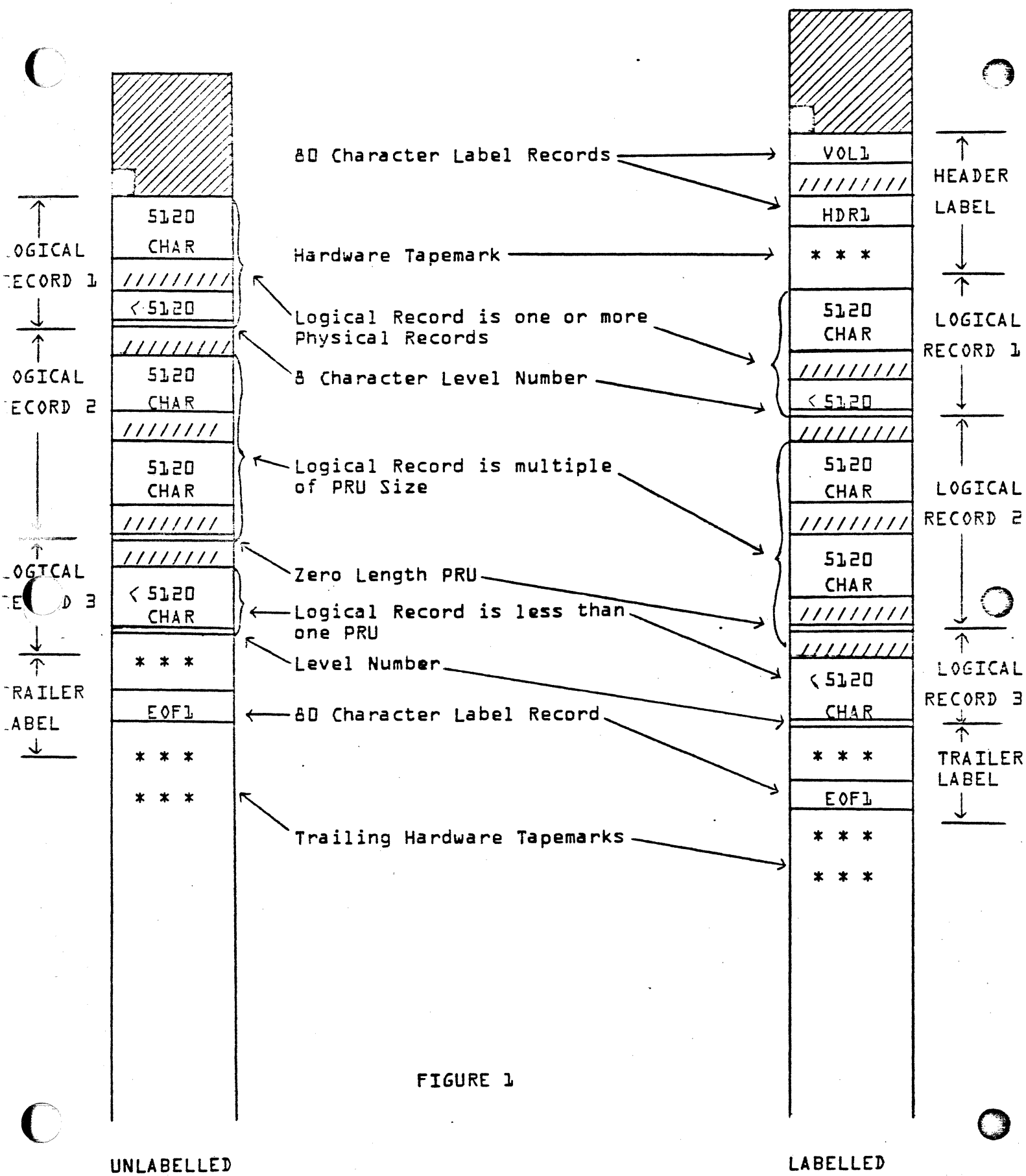


FIGURE 1

SCOPE STANDARD TAPE FORMAT (CODED)

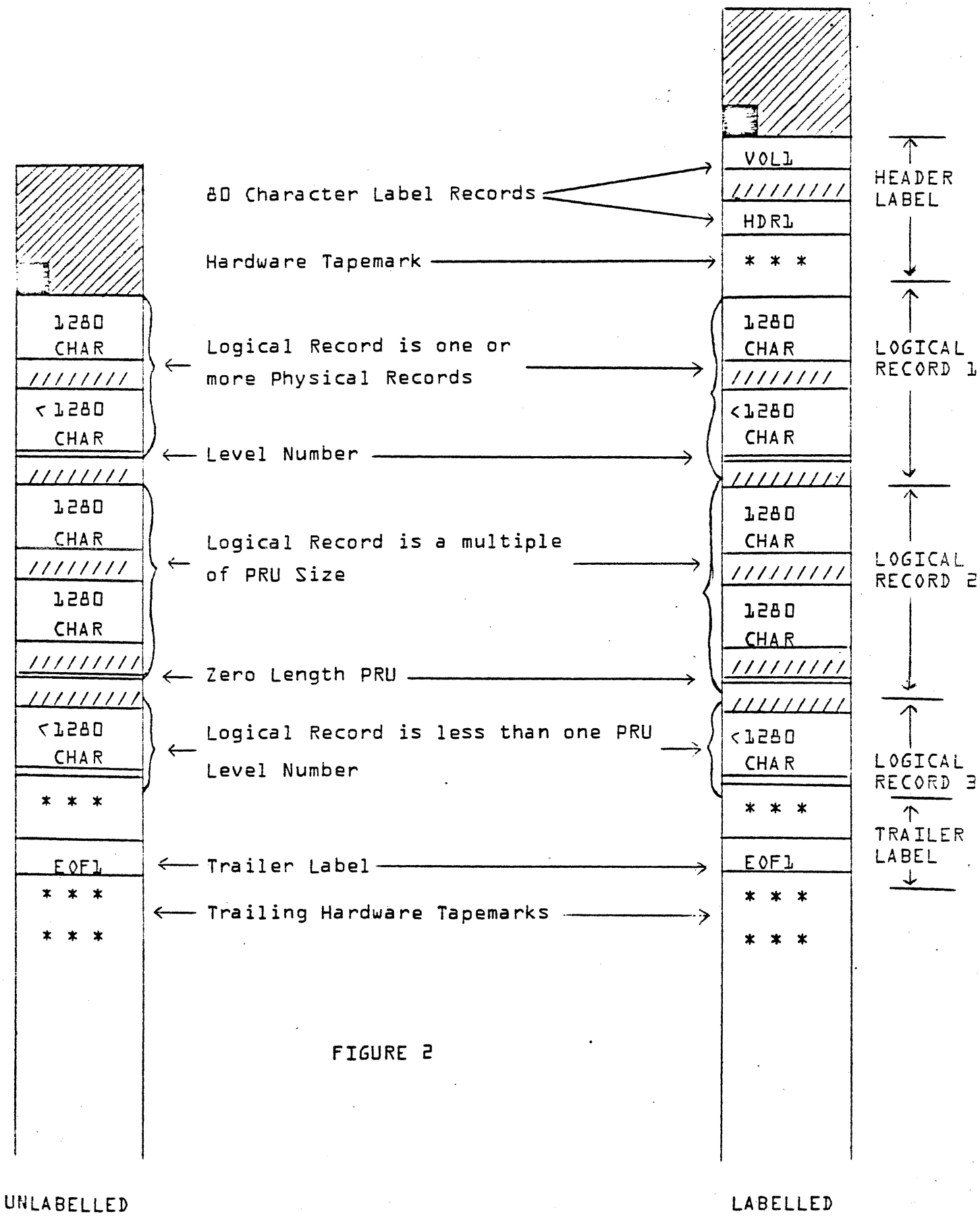


FIGURE 2

2.2 External (X) Tape Format

External tapes as supported by 6SC3.2 and 7SCV1.1 are compatible with tapes created under 6000 SCOPE 2.0 and earlier systems. External tapes may not be labeled. Binary external tapes are recorded in odd parity, coded files are recorded in even parity. Under 7SCV1.1, the binary format is refered to as an X tape and the coded format is called an E tape. Under 6SC3.2 both the binary and coded formats are referred to as the X format. For binary tapes, the maximum physical record size if 5120 characters. A physical record is a multiple of 10 characters. A logical record is one or more PRU's terminated by a short or zero length PRU; no 8 character level number appendage is included. A short PRU is a physical record of 10_n characters where $0 < n \leq 511$. A zero length PRU which is a physical record of 8 characters in length terminates a logical record which is an integral multiple of 5120 characters in length. Figure 3 illustrates the binary external tape format.

For coded files a logical record is a physical record of 136 characters. A logical record is commonly referred to as a line image. If a line terminator is encountered before 136 characters of data, it is converted to blanks and the remainder of the 136 character PRU is set to blanks. If 136 characters occur before a line terminator they are written as a logical (physical) record. Characters 137 thru 140 are lost. Writing continues with character 141. No 8 character level number appendage is added. Figure 4 illustrates the external tape format for a coded file.

End of File is defined to be a single hardware tapemark. End of Information is not defined by 6000 SCOPE. It is the user's responsibility to know the structure of the tape (i.e., the number of files). When writing an external format tape, 6000 SCOPE appends 4 hardware tape marks to the data before the tape is rewound or unloaded. Under 7SC1.1, 2 hardware tapemarks are written at the end of the data to indicate End of Information. When the tape is staged in under 7SC1.1, the user defines the End of Information in the sense that he has the option of staging to a single tapemark or a double tapemark.

EXTERNAL TAPE FORMAT (BINARY)

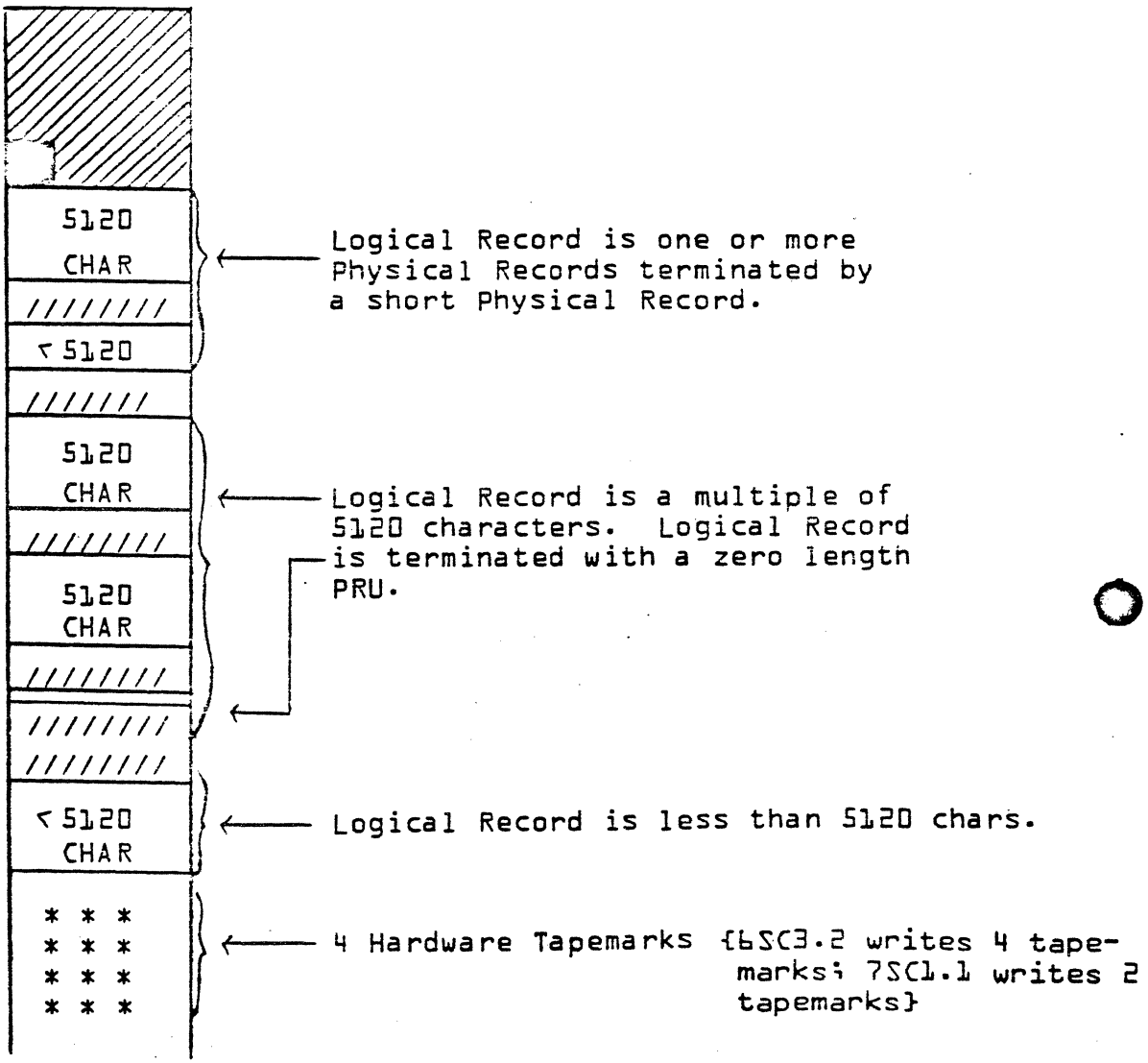


FIGURE 3

EXTERNAL TAPE FORMAT {CODED}

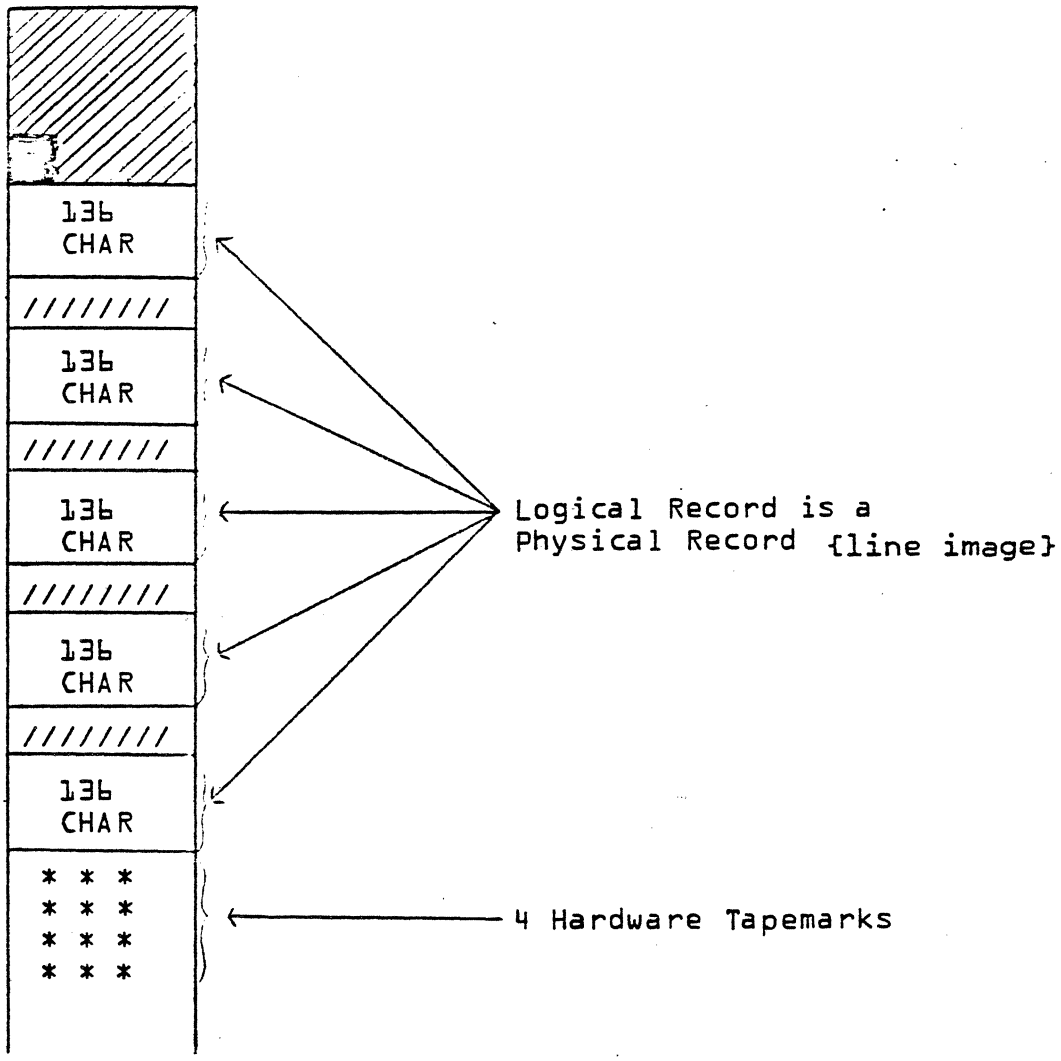


FIGURE 4

2.3 Stranger (S) Tape Format

The Stranger tape format does not define a "blocking format" or a "logical record format" other than the fact that the 6000 SCOPE CIO routine equates a logical record to a physical record. The S driver is a raw data processor; it does not recognize a logical data format or structure. It does code conversion where necessary, but the conversion is one-to-one* and no character or group of characters have any special meaning. The only data structure limitations are hardware imposed and consist of limiting a physical record to contain a multiple of 2 characters (12 bits), a minimum of n characters where n is an installation parameter, IP.NOISE, and a maximum of 5120 characters. The maximum record size of 5120 characters is a consequence of PP buffer size limitations and applies to both binary and coded records. No special significance is attached to any character or series of characters so that the data transferred to and from the user's buffer accurately reflects what appears on tape.* No level numbers are appended to data and no line terminators are recognized in data.

---

* Character code conversion is one for one with one exception. A DISPLAY code of 63_8 is converted to an External BCD code of 12_8 on output. An External BCD code of 12_8 is converted to a DISPLAY code of 33_8 on input. This exception holds true for 6SC3.2 using a 6681.

STRANGER TAPE FORMAT (BINARY OR CODED)

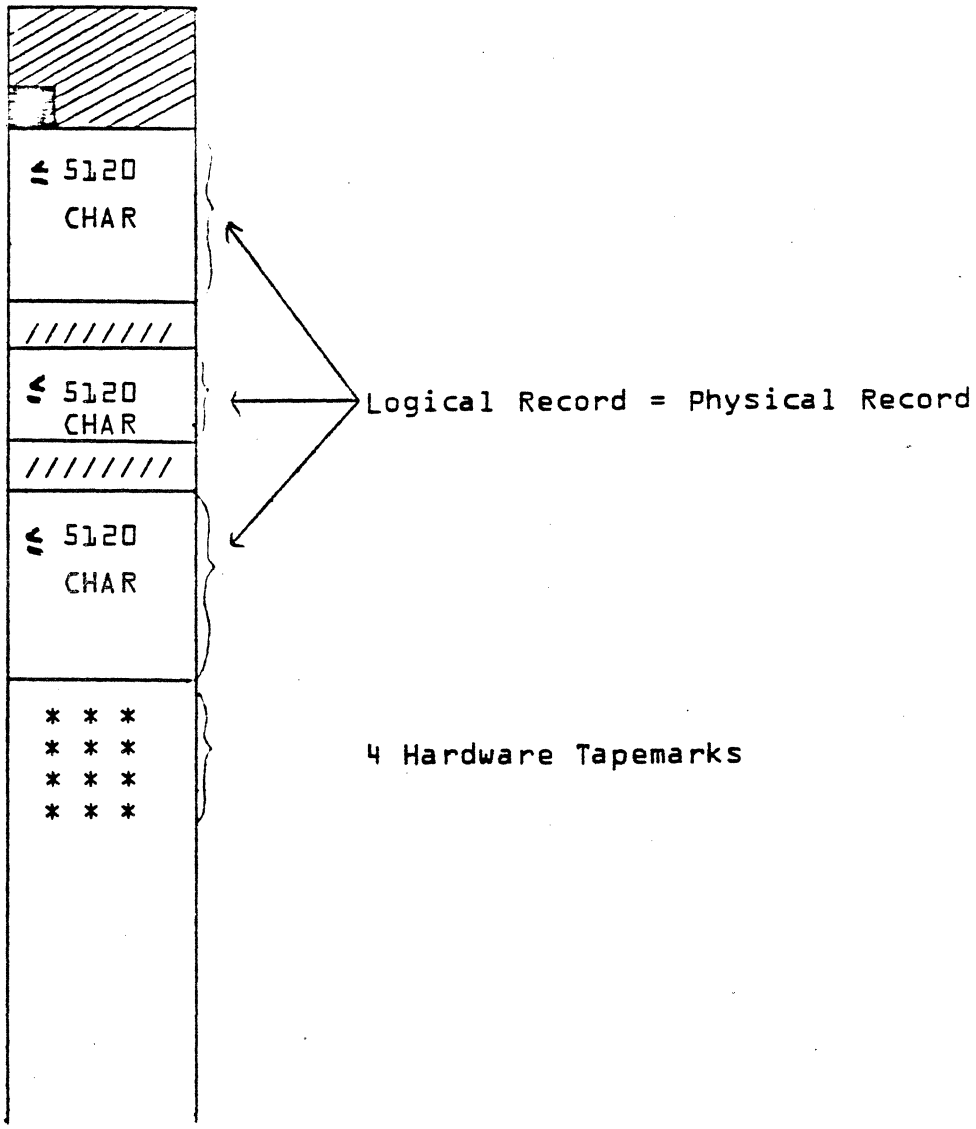


FIGURE 5

Figure 5 illustrates the unlabeled S tape format (binary or coded). Binary stranger tapes are recorded in odd parity; coded tapes are recorded in even parity. S tapes may be labeled (SCOPE standard or 3000 series labels) or unlabeled. Each physical record is a logical record. A physical record must be less than or equal to 5120 characters. A single tapemark indicates EOF. When the file is created under 6000 SCOPE, the system appends 4 hardware tapemarks at file close time. EOI is not defined. The user is responsible for knowing the structure of the tape, i.e., the number of files.

2.4 Long Stranger (L) Tape Format

The L format (coded or binary) is identical to the S format except the maximum size of the physical (logical) record is limited only by the user's CM buffer size. The physical record size is not limited to 5120 characters. The L tape read and write drivers transfer one 60-bit word from CM to the PP at a time and then one or two bytes at a time from the PP to the tape rather than a large number of 60-bit words (i.e., 512) as does the S tape drivers. As with S tapes, physical record is a multiple of 2 characters with a minimum size of n characters where n is an installation parameter, IP.NOISE. No special significance is associated with any character or group of characters

Figure 6 illustrates the unlabeled L tape format. L tapes may be unlabeled or labeled (standard SCOPE or 3000 series labels).

LONG STRANGER TAPE FORMAT (BINARY OR CODED)

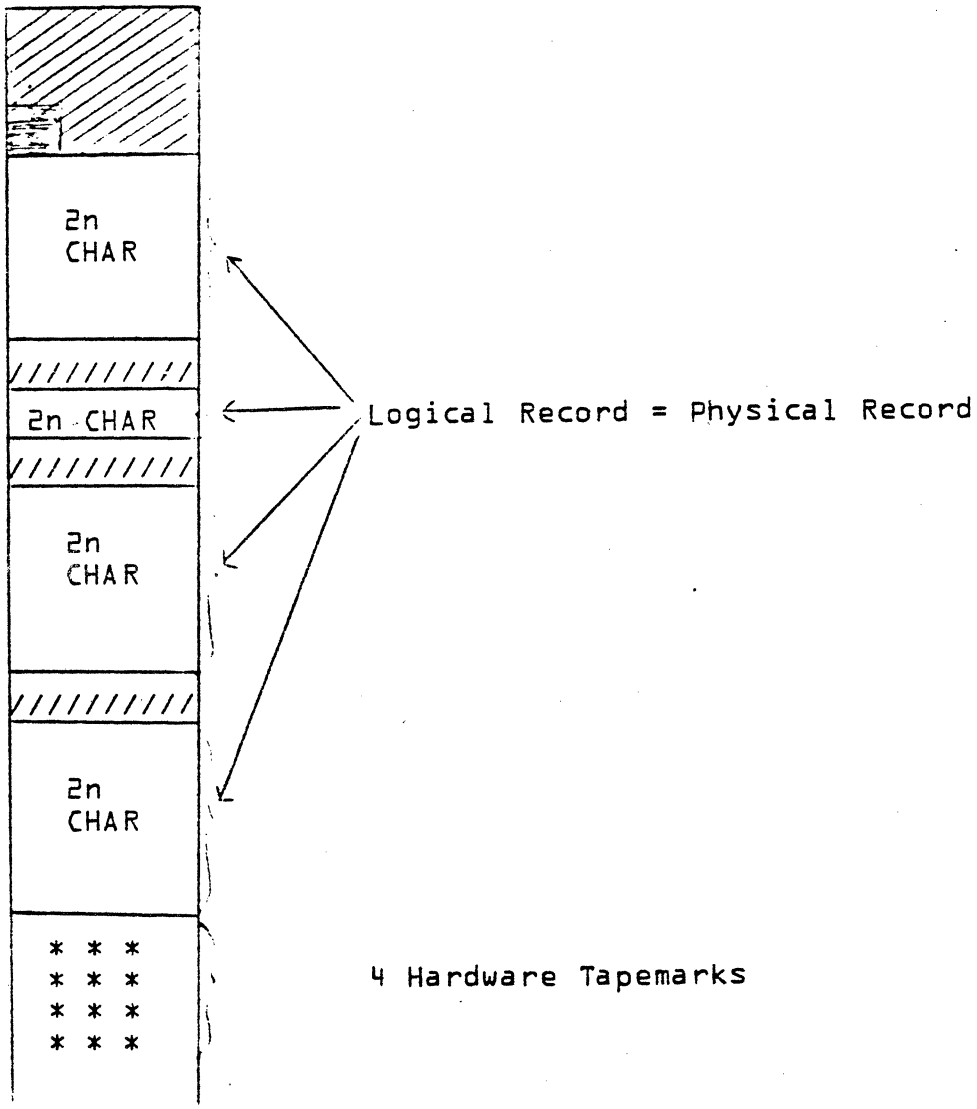
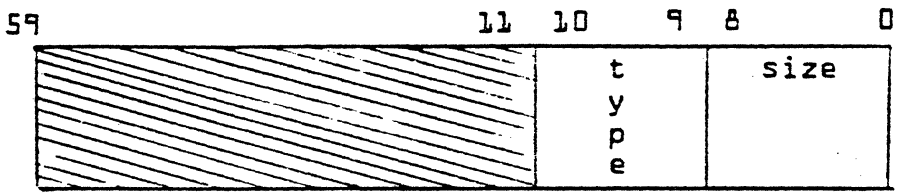


FIGURE 6

2.5 7000 Format

The 7000 format is the system default tape format under 7000 SCOPE V1.1. Each physical block (PRU) is 512 60 bit words in length except for the very last block on the tape. The tape is written in odd parity; labels are not allowed. The first 60 bit word of each block is a boundary control word. Each logical record, which is an integral number of 60 bit words in length, is followed by an end-of-record (EOR) control word. Each logical file is delimited by an end-of-file (EOF) control word. The end-of-information is indicated by an EOI control word and a short physical block. If the multi-file file is an integral number of PRU's in length, a physical record of 8 zero characters is appended to the file to indicate end-of-information (EOI). The last full word of the multi-file file is an EOI control word. Each control word contains a "size" field which points to the next control word. Figure 7 illustrates the control word format. Figure 8 illustrates a file written in 7000 format.

7000 Format Control Word



type 0 - boundary word (first word of each physical block)

1 - End of Record (logical)

2 - End of File (logical)

3 - End of Information (last word in the multi-file file, size = 0)

size - number of 60 bit words between this control word and the next control word. $0 \leq size \leq 777$

FIGURE 7

7000 TAPE FORMAT

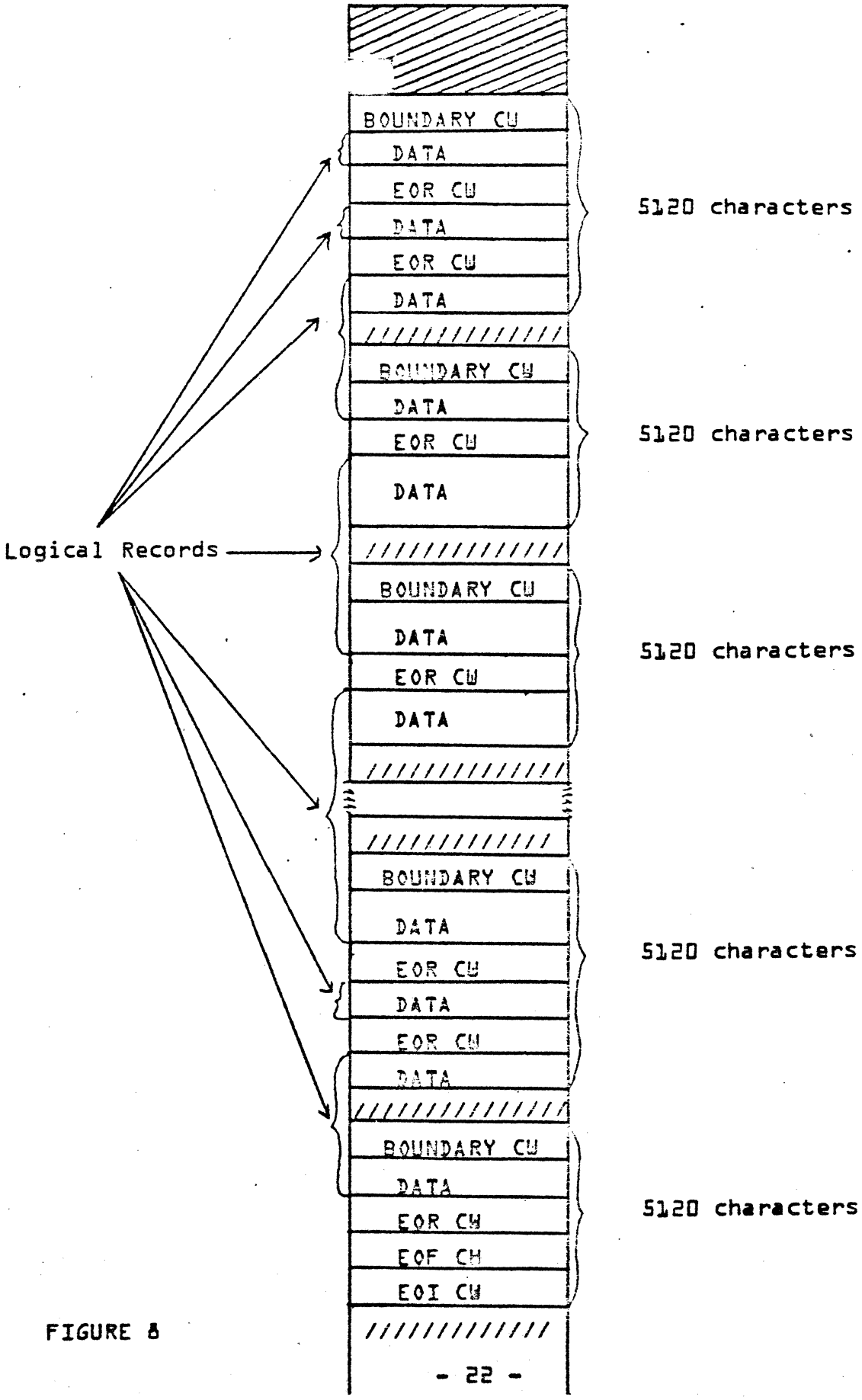
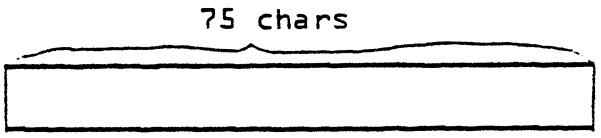


FIGURE 8

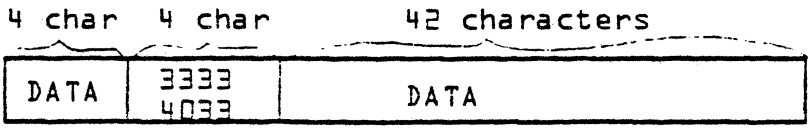
2.6 Fixed Length (F)

Fixed length records are defined as records whose length is constant. The size of the fixed length record is specified by the user in number of characters, FL. All records in the file are of equal size; records need not be an integral number of 60 bit words in length. An example of a fixed length record whose length, FL, is 75 is:



2.7 Decimal Character Count (D)

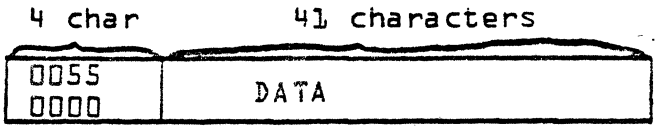
Decimal character count records are defined as records whose length is specified in a length field located within the record. The length field begins in character position LP (character positions begin with 0) and is LL ($1 \leq LL \leq 64$) characters in length. The record size including the length field is specified in the user's buffer as display coded decimal, right justified and coded zero filled within the length field. The records are variable length. However, the size must be less than or equal to a user specified length, MRL. An example of a D type record whose MRL=80, LP=4, LL=4 follows:



Characters 4 thru 7 are 33334033_8 which is 50 in display coded decimal, right justified, zero filled.

2.8 Binary Character Count (B)

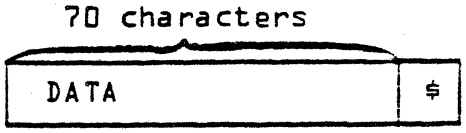
Binary character count records are defined as records whose length is specified in a length field which occupies the first four characters in the record. The record size, including the length field, is specified in the user's buffer as binary in the first two characters. The lower two characters are reserved for system use. The record size is variable. However, the record size must be less than or equal to a user specified limit, MRL. MRL must be less than or equal to 4095. An example of a binary character count record whose size is $45D=55_8$ and MRL=80 follows:



The first 4 characters are 00550000.

2.9 Record Mark (R)

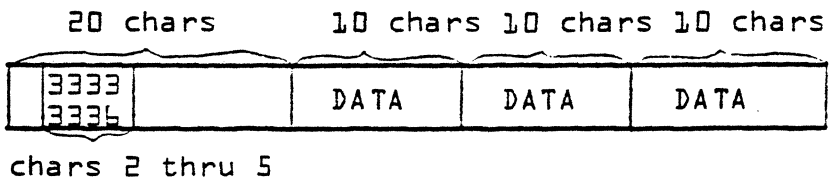
A record mark record is defined as a record whose length is determined by the appearance of a user specified character, RMK. The terminating character is passed to the user. The RMK character must not be used as data within the record. The user may specify a minimum record length, MNR, to enhance performance. Records may be variable length but must be less than or equal to a user specified limit, MRL. An example of an R type record whose size if 71, MRL=71, MNR=60, and RMK=53B follows:



53B is the display code for '$'.

2.10 Trailer Count (T)

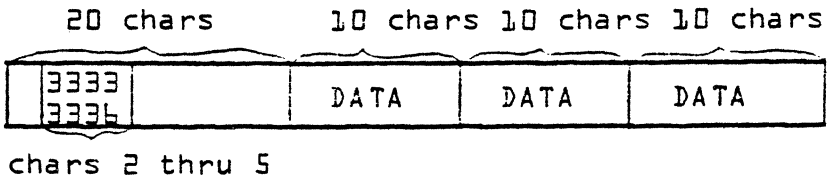
A trailer count record is defined as a record which is composed of a fixed length base and a variable number of fixed length trailer items. The lengths of the base, HL, and trailer item, TL, are specified by the user. The number of trailer items is specified in a count field contained within the base. The count fields starting character position, CP, (character positions begin with 0) and length, CL, ($1 \leq CL \leq 64$) are specified by the user. The value contained in the count field is represented in the user's buffer as display coded decimal, right justified and zero filled. Records are variable length but must be less than or equal to the user specified limit, MRL. A record does not need to start on a 60 bit word boundary. HL, TL, CL, CP and MRL are stored in the FIT. An example of the T type record whose MRL=60, CP=2, CL=4, HL=20, TL=10, and number of trailer items is 3 follows:



The count field contains 33333336_8 = display coded decimal 3, right justified, zero filled.

2.10 Trailer Count (T)

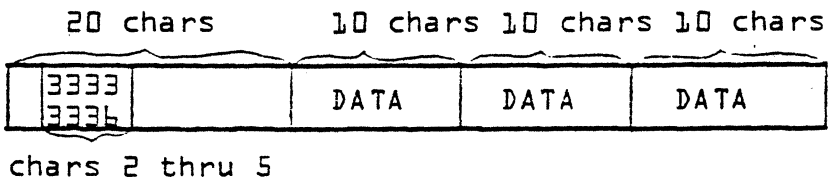
A trailer count record is defined as a record which is composed of a fixed length base and a variable number of fixed length trailer items. The lengths of the base, HL, and trailer item, TL, are specified by the user. The number of trailer items is specified in a count field contained within the base. The count fields starting character position, CP, (character positions begin with 0) and length, CL, ($1 \leq CL \leq 64$) are specified by the user. The value contained in the count field is represented in the user's buffer as display coded decimal, right justified and zero filled. Records are variable length but must be less than or equal to the user specified limit, MRL. A record does not need to start on a 60 bit word boundary. HL, TL, CL, CP and MRL are stored in the FIT. An example of the T type record whose MRL=60, CP=2, CL=4, HL=20, TL=10, and number of trailer items is 3 follows:



The count field contains 3333336_8 = display coded decimal 3, right justified, zero filled.

2.10 Trailer Count {T}

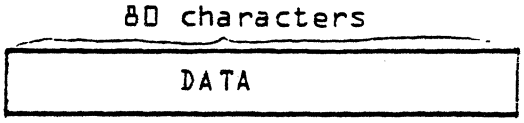
A trailer count record is defined as a record which is composed of a fixed length base and a variable number of fixed length trailer items. The lengths of the base, HL, and trailer item, TL, are specified by the user. The number of trailer items is specified in a count field contained within the base. The count fields starting character position, CP, {character positions begin with 0} and length, CL, {1 ≤ CL ≤ 64} are specified by the user. The value contained in the count field is represented in the user's buffer as display coded decimal, right justified and zero filled. Records are variable length but must be less than or equal to the user specified limit, MRL. A record does not need to start on a 60 bit word boundary. HL,TL,CL,CP and MRL are stored in the FIT. An example of the T type record whose MRL=60, CP=2, CL=4,HL=20, TL=10, and number of trailer items is 3 follows:



The count field contains 33333336_8 = display coded decimal 3, right justified, zero filled.

2.11 Undefined (U)

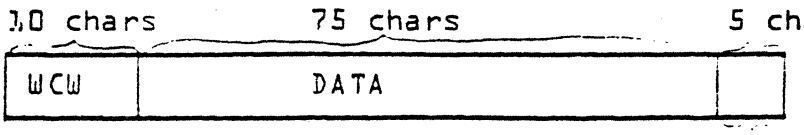
The undefined record is defined as a record whose format does not conform to any of the other record descriptions. The user specifies the maximum record size, MRL, except for input files where a physical block may be taken as one record. An example of a U type record whose MRL=80 and size is 80 follows:



2.12 Control Word (W)

The control word record is defined as a record which is composed of an integral number of 60 bit words prefixed by a control word (WCW) whose format is described in Appendix D. The control word which is the first 60 bit word of each record is furnished by the data manager on output and stripped off by the data manager on input. The control word specifies the record length in 60 bit words, the number of unused bits in the last 60 bit word, the number of 60 bit words in the previous record and three single bit fields which carry special meanings. Refer to Appendix D for a complete description of the WCW format.

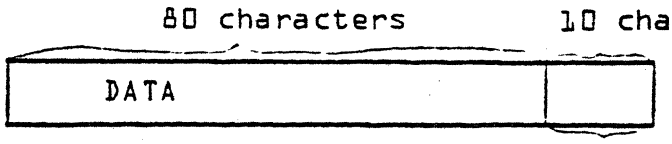
The record does not need to be a multiple of 60 bit words, but each record must begin on a 60 bit word boundary. (Records which are not an integral number of 60 bit words in length are padded to fill the last 60 bit word.) The W type record is the 7000 V2 and 6RM system default record type. The size of a W type record is variable but the size must be less than or equal to a user specified limit, MRL. An example of a W type record whose size is 75 and MRL=75 follows:



5 unused chars needed to pad the record to a 60 bit word boundary. (not necessarily blanks)

2.13 Zero Byte (Z)

The zero byte record is defined as a record whose length is determined by the appearance of a twelve bit zero byte in the low order twelve bits of a 60 bit word. Records are variable length but they must begin on a 60 bit word boundary and be less than or equal to a user specified maximum size, FL. On input, the zero byte is stripped from the record and blank padding is inserted to the user specified maximum record size, FL. On output, the terminating zero byte is supplied by either the user or the data manager. If the user supplied record is less than FL character, the user must supply the zero byte. If the user supplied record is equal to FL, the data manager strips any trailing blanks and supplies the terminating zero byte. An example of a Z type record whose FL=80, size is 80 characters, and the 80th character is non-blank follows:

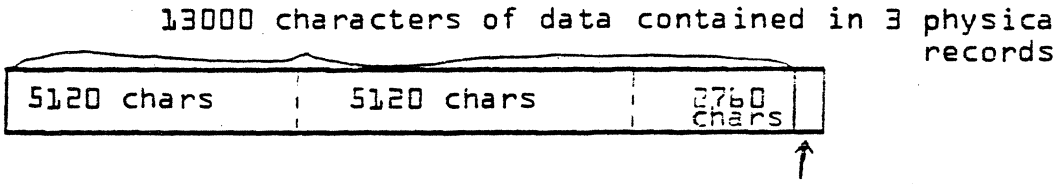


60 bits appended by the data manager.

In this example the 7SC2.0 Data Manager will append 8 blank characters followed by the 12 b it zero byte and transfer 90 characters to the file. When this record is read with FL=80, the trailing 8 blanks plus the 12 bit zero byte will be stripped by the Data Manager.

2.14 SCOPE Logical {S}

The SCOPE logical record is defined as a record which is composed of zero or more physical records of MBL characters followed by a terminating physical record of less than MBL characters. MBL is the maximum number of characters per physical block as specified by the user. MBL must be an integral number of 60 bit words. Each record consists of an integral number of 60 bit words followed by an 8 character level number appendage. The appendage is supplied by the data manager on output and is stripped off by the data manager on input. Level numbers 0 thru 16 indicate end of logical record; level number 17 indicates end of logical file and is always carried in a zero length physical record. An example of an S type record whose length is 13000 characters and MBL=5120 follows:



The record format described in this paragraph is identical to the 6000 SCOPE standard binary format as described in paragraph 2.1.

3.0 PHYSICAL BLOCK FORMATS

Four physical block formats are supported by the 5 operating systems being considered. This section defines each block format. Throughout this section the terms "physical record" and "block" may be interchanged. Refer to Appendix B for illustrations of blocking types.

3.1 Record Count (K)

The record count block format is defined as a physical record which contains a fixed number, RB, of whole logical records. Each physical record contains RB records except the last physical record of a file or a physical record before a logical ENDFILE. Blocks may be variable length (if logical record is variable length). Logical records do not span block boundaries. The K type block is an allowable ANSI tape interchange format.

3.2 Character Count (C)

The character count block format is defined as a physical record of length equal to or less than a specified number of characters, MBL. Each block is MBL characters except possible the last block and any block written to indicate a special condition (i.e., end of SCOPE logical record or file). Logical records may span block boundaries.

3.3 Exact Records (E)

The exact records block format is defined as a block which contains an integral number of logical records, as many whole records as can be contained in the specified maximum block size, MBL. Records may not span block boundaries. The E type block is an allowable format for ANSI standard tape interchange.

3.4 Internal Block {I}

The internal block format is defined as a physical record which consists of a 60 bit block control word followed by up to 511 60 bit words of the file image. The control word is refered to as the ICW as described in Appendix D. The ICW points to the first logical record starting in the block, indicates the number of the first record in the block and indicates the number of the block. All I type blocks are 512 60 bit words in length except possibly the last block, or a block before the logical ENDFILE.

4.0 OCCURRENCE OF RECORD TYPE AND BLOCK TYPE PER OPERATING SYSTEM

This section describes how a record and block type is specified and which record types and block types are supported by each of the 5 operating systems being considered. Table 1 indicates which record types are supported; Table 2 indicates which block types are supported. A set of footnotes are included to clarify those table entries which are not completely straightforward. The information presented in this section pertains primarily to 7-track tapes. Refer to section 9.0 for information concerning 9-track tapes.

The selection of the record and block format for a magnetic tape is not accomplished in the same manner for each of the 5 operating systems being considered. The various formats are specified in the following manner:

6SC3.2 REQUEST control card data format parameter:

X - implies external

S - implies stranger

L - implies long stranger

none of the above - SCOPE standard

The mode, binary or coded, is specified by bit 1 of word one of the FET, set up directly by the user or the utility. (for example, COPYBR)

6SC3.3 REQUEST or LABEL control card device type parameter:

S - implies stranger

L - implies long stranger

neither S nor L - implies SCOPE standard

The mode, coded or binary, is specified as under 6SC3.2.

6SC3.4 The format may be specified by a combination of control cards and macros.

a) REQUEST control card device type parameter selects format. The format is specified the same as under 6SC3.3; mode, coded or binary, is specified as under 6SC3.2.

b) If 6RM is to be used, the FILE control card and/or macro further defines the format:

Record type is specified by RT=F,D,R,T,Z,U, S, or W.

Block type is specified by BT=K,I,C, or E.

7SCV1.1 REQUEST or TAPE control card mode parameter:

X - implies External Binary

E - implies External Coded

Z - implies 7000 Format

7SCV2 FILE control card or macro

Record type, RT = F,D,B,R,T,U,S,W, or Z

Block type, BT = K,C,I,E

If BT and RT are not specified, BT=I and RT=W is assigned.

If RT=S, BT=C is assigned.

If BT and RB are not specified and RT ≠ W or S, BT=K with RB=1 is assigned.

4.1 Record Type vs. Operating System

To insure the interchangeability of a particular format between two different operating systems, an understanding of the format and structure is necessary. The existence of characters having special meaning or the occurrence of special character encoding/decoding may prohibit a format from being read under a different operating system. Any remark made in this document concerning the interchangeability of a format is qualified by the 2 preceding statements. For an example, a SCOPE coded format written by FORTRAN under 6SC3.2 cannot be read meaningful by 7SCV2 due to the special treatment by 6SC3.2 of the line terminator. {On a coded SCOPE format, a 12 bit zero byte in the low order positions of a 60 bit word are converted to 1632_8 XBCD on output and the 1632 XBCD is converted back to 0000_8 on input}. However, a SCOPE coded format written under 6SC3.2 by a COMPASS user may be read successfully under 7SCV2 by properly defining the format providing the data contains no line terminators. At this time, this document does not address the topic of which formats are generated by the various product sets using the various write functions.

Table 1 indicates which record formats are supported by the various operating systems.

RECORD TYPE/OPERATING SYSTEM

RECORD TYPE \ OPERATING SYSTEM	6SC3.2	6SC3.3	6SC3.4	7SCV1.1	7SCV2
F	No	No	Yes	No	Yes
B	No	No	No	No	Yes
D	No	No	Yes	No	Yes
R	No	No	Yes	No	Yes
T	No	No	Yes	NO	Yes
U	Yes [A]	Yes [A]	Yes	Yes [A]	Yes
W	No	No	Yes	No	Yes
Z	Yes [B]	Yes [B]	Yes	No	Yes
S	Yes [C]	Yes [C]	Yes [C]	No	Yes
Long Stranger	Yes	Yes	Yes	No	Yes [A]
SCOPE Binary	Yes	Yes	Yes	Yes [H]	Yes [C]
SCOPE Coded	Yes	Yes	Yes	No	No
External Binary	Yes	No [D,G]	No [D,G]	Yes	Yes [E,D]
External Coded	Yes	Yes [G]	Yes [F,G]	Yes	Yes [F]
Stranger	Yes	Yes	Yes	Yes [I]	Yes [A]
7000 Format	No	No	No	Yes	No

TABLE 1

A. By definition the Stranger, Long Stranger, coded external, and U (with 1 logical record per physical record) record types are alike except for the maximum physical record size limitations. Maximum physical record sizes are: stranger - 5120 chars, long stranger - user buffer length, coded external - 136 chars; U - station dependent. These record types may be interchanged among operating systems so long as the physical records are the same size and the file is properly defined. For example a stranger tape written by 6SC3.3 may be read by 7SCV2 if defined, as BT=K, RB=1, MBL=5120, RT=U, MRL=5120.

B. The Z type record format is written by COBOL under 6SC3.2 and 6SC3.3 where the COBOL user REQUEST's a SCOPE standard tape format and specifies RECORDING MODE IS BINARY. The format which COBOL writes is equivalent to the 7SCV2 format defined by RT=Z, BT=C, and MBL=5120.

C. By definition, the S (SCOPE logical) and SCOPE standard binary record formats are equivalent, including level number.

D. 6SC3.3, 6SC3.4 and 7SC2.0 include a utility program called COPYXS which will copy n number of files from an External binary formatted file to a SCOPE binary formatted file.

E. An external binary formatted tape may be read by 7SCV2 if the user defines the file as BT=C, RT=S, and MBL=5120. However, since the data manager expects a level number appendage at the end of each logical record, control will be passed to the user error exit with insufficient data status indicated in the FIT when end of logical record is encountered.

F. An external coded format written by 6SC3.2 or 7SCV1.1 may be read by 6SC3.4 and 7SCV2 if the file is defined as BT=K, RB=1, RT=F, FL=136, parity is even and character conversion is required.

G. External binary and external coded format tapes generated by 6SC3.2 and 7SC1.1 may be read by 6SC3.3 and 6SC3.4 as Stranger tapes. However, the S driver does not recognize the external mode's short or zero length PRU as the logical end of record.

H. 7SC1.1 can read 6000 SCOPE Standard binary tape files if defined as X format tapes. 7SC1.1 cannot write a tape in the 6000 SCOPE standard binary tape format.

I. The 6000 SCOPE binary Stranger format is a special case of the 7SC1.1 X format. The 6000 SCOPE binary Stranger format can be read and written by 7SC1.1 provided the user insures that each logical record of the X format/stranger format tape is less than 512 60 bit words in length.

4.2 Blocking Type/Operating System

	6SC3.2	6SC3.3	6SC3.4	7SCV1.1	7SCV2
K	yes; for external coded, stranger and long stranger record types	yes; for stranger and long stranger record types	yes; for stranger, long stranger, F, D, R, T, U, Z, and W record types	yes; for external coded record type	yes; for F, D, B, R, T, U, X, and W record types
C	yes; for SCOPE binary, External binary and SCOPE coded record types	yes; for SCOPE binary and SCOPE coded record types	yes; for SCOPE binary, SCOPE coded, F, D, R, T, U, Z, W, and S record types	yes; for External binary, and 7000 format record types	yes; for F, D, B, R, T, U, X, W, and S record types
E	no	no	yes; for F, D, R, T, U, Z and W record types	no	yes; for F, D, B, R, T, U, Z, and W record types
I	no	no	yes; for W type records only	no	yes; for W type records only

5.0 TAPE FILE STRUCTURE

The structure of a magnetic tape file is a function of the data format, whether the tape file is labelled or unlabelled and the number of reels necessary to contain the file. Sections 5.1 and 5.2 discuss labelled and unlabelled tape file structures respectively.

5.1 Labelled Tape File Structures

The structure of a labelled tape file is determined by the number of files and the number of reels necessary to contain the files. Four configurations are defined: Single-File/Single-Reel, Single-File/Multi-Reel, Multi-File/Single-Reel and Multi-File/Multi-Reel. The structure of each configuration is shown below. Required labels are indicated by a 4-character identifier. (Refer to Section 6 for the definition of the label format). Hardware tapemarks are indicated by an asterisk.

Single-File/Single-Reel

VOL1 HDR1*---Data Blocks---*EOF1**

Single-File/Multi-Reel

Reel 1 VOL1 HDR1*---First Volume Data---*EOV1**

Reel 2 VOL1 HDR1*---Last Volume Data---*EOF1**

Multi-File/Single-Reel

VOL1 HDR1*---File A---*EOF1*HDR1*---File B---*EOF1**

Multi-File/Multi-Reel

Reel 1 VOL1 HDR1*---File A---*EOF1*HDR1*---File B---*EOV1**

Reel 2 VOL1 HDR1*---Continuation of File B---*EOV1**

Reel 3 VOL1 HDR1*---Remainder of File B---*EOF1*HDR1*---File C---*EOF1**

Volume Header Label

The first physical record in the reel must be a volume header label; it may not appear elsewhere.

File Header Label

Every file must be preceded by a file header label and every file header must be preceded by a tape mark or a volume header label. When a volume ends within a file, the continuation of that file in the next reel must also be preceded by a file header.

File Trailer Label

A file trailer label is required as the last block of every file. A file trailer must be preceded and followed by a tape mark, and if it is the last file trailer in the reel, two following tapemarks are required.

Volume Trailer Label

When a reel ends within a file, the last physical record of the file in the first reel must be followed by a volume trailer label which must be preceded by a single hardware tapemark and followed by 2 hardware tapemarks.

Special case end-of-tape conditions may occur when the reflective spot is encountered during the writing of a label group on a multi-file tape. Depending upon when the reflective spot appears, one of the following configurations will be generated.

1. Reel 1 ---File A---*EOV1**

   Reel 2 VOL1 HDR1**EOF1*HDR1*---File B---

2. Reel 1 ---File A---*EOF1*HDR1**EOV1**

   Reel 2 VOL1 HDR1*---File B---

3. Reel 1 ---File A---*EOF1*EOV1**

   Reel 2 VOL1*HDR1*---File B---

5.1.1 Labelled Tape File Structures Per Operating System

The five operating systems being considered do not support tape label structures as described in Section 5.1 to the same degree. The following table indicates which structures are supported by the various operating systems. 7SCV1.1 does not support labelled tapes.

	6SC3.2	6SC3.3	6SC3.4	7SCV1.1	7SCV2
Single-File/ Single-Reel	yes	yes	yes	no	yes
Multi-File/ Single-Reel	yes *	yes *	yes *	no	no
Single-File/ Multi-Reel	yes	yes	yes	no	yes
Multi-File/ Multi-Reel	yes *	yes *	yes *	no	no

* The multi-file/single-reel and multi-file/multi-reel structures are not allowed with 3000 computer series labels.

5.2 Unlabelled Tape File Structure

The structure of an unlabelled tape file is determined by the number of reels needed to contain the file and the data format of the file. The concept of a single-file/single-reel and single-file/multi-reel is valid and well defined for an unlabelled tape file. The term "multi-file" is normally associated with labelled files. However, the concept of a "multi-file" file has meaning when describing unlabelled tapes. An unlabelled tape file may be partitioned into "logical files" by end-of-file indicators. The end-of-file indicator is dependent upon the data format. When the end-of-file indicator is detected upon reading the file, end of file status is returned to the user.

The general formats of the various unlabelled structures are shown below. The end of file indicator is indicated by EOF, end of volume is indicated by EOV and end of information is indicated by EOI. EOF, EOV and EOI are defined for the various record types in Table 3.

Single-File/Single-Reel

---File A---EOI

Single-File/Multi-Reel

---File A {first part}---EOV

---File A {last part}---EOI

Multi-File/Single-Reel

---File A---EOF---File B---EOI

Multi-File/Multi-Reel

```
---File A---EOF---File B {First part}---EOV      Reel 1
---File B {last part}---EOF---File C---EOI       Reel 2
```

Table 3 defines the EOF, EOV, and EOI indicators for the various record types.

TABLE 3

EOF, EOV, AND EOI DEFINITIONS PER RECORD TYPE

	EOF	EOV	EOI
SCOPE Std.	Zero length level 17 PRU	*EOV1**	*EOF1**
External {6SC3.2}	*	****	**** 1
External {7SCV1.1}	*	End of tape	**
Stranger	*	****	**** 1
Long Stranger	*	****	**** 1
7000 Format	EOF control word	End of tape	Short PRU containing EOI control word
Fixed Length	*	**	**
Decimal Char.	*	**	**
Binary Char.	*	**	**
Record Mark	*	**	**
Trailer Count	*	**	**
Undefined	*	**	**
Control Word	EOF control word	**	**
Zero Byte with K and I block types	*	**	**
Zero Byte with C type block	Zero length level 17 PRU	**	**
SCOPE Logical	Zero length level 17 PRU	**	**

* = Hardware Tapemark

** = Double Hardware Tapemark

NOTE 1: EOI is not defined for this format. However, when the tape is written, 4 tapemarks are written at the end of the data.

6.0 LABEL FORMATS

Four of the five operation systems being considered provide label processing for magnetic tape files. The following table indicates which label formats are supported for 7 and 9 track tapes per operating system.

Label Formats \ Operating System	6SC3.2	6SC3.3	6SC3.4	7SC1.1	7SC2.0
3000 Series Computer Labels	7 track	7 & 9 track	7 & 9 track	no	no
SCOPE Standard	7 track	7 & 9 track	7 & 9 track	no	7 & 9 track

The 3000 Series Computer Labels are the standard labels supported by CDC 3000 Series computer operating systems. A description of the 3000 Series computer labels is given in Section 6.3. The SCOPE Standard labels as supported by 6SC3.2 and by 6SC3.3 are identical; they are designed similar to the proposed USA Standard for Magnetic Tape labels and File Structure for Information Interchange submitted by the X.3.2/457 Committee on November 28, 1966 and revised by the X.3.2/513 Committee July 18, 1968. A description of the required labels is given in Section 6.2.

The SCOPE Standard labels as supported by 6SC3.4 and 7SC2.0 are identical, they conform to the American National Standard for Magnetic Tape Labels for Information Interchange submitted by the X3.27 Committee on October 20, 1969. A description of the required labels is given in Section 6.2. 7SC1.1 does not provide the capability to process labels on magnetic tape files.

6.1 Label Types per Operating System

VOL1, HDR1, EOF1 and EOV1 labels are defined as required labels. All other defined labels (HDR2-9, UHLa, UVL1-9, EOV2-9, UTLa and EOF2-9) are considered to be optional labels.

6SC3.2 and 6SC3.3 support the 4 required lables (VOL1, HDR1, EOF1 and EOV1). The HDR2-9 and UHLa labels cannot be written and they are skipped on input. The remaining optional labels (UVL1-9, EOV2-9, UTLa and EOF2-9) are not allowed.

Under 6000 SCOPE 3.4 all required labels are checked by the operating system on input and are generated by the operating system on output if the user does not supply them. All optional labels must be provided by the user to the operating system on output and are returned to the user on input.

7000 SCOPE 2.0 supports the processing of the 4 required labels but processing of the optional labels is not allowed.

6.2 SCOPE Standard Label Format Description

Section 6.2 describes the format for the 4 SCOPE Standard labels VOL1, HDR1, EOF1 and EOV1. The column titled Description 1 describes the field as defined by 6SC3.2 and 6SC3.3. The column titled Description 2 describes the various fields as defined by 6SC3.4 and 7SC2.0. The letter "n" means any numeric digit 0 through 9 and the letter "a" means any member of the character set in which the label is written.

VOLUME HEADER LABEL

FIELD	LENGTH	POSITIONS	NAME	DESCRIPTION 1	DESCRIPTION 2
1	3	1-3	Label Identifier	Must be VOL	Must be VOL
2	1	4	Label Number	Must be 1	Must be 1
3	6	5-10	Volume Serial Number	Six "n" characters which identify this physical reel of tape	Six "a" characters which identify this physical reel of tape
4	1	11	Accessibility	Blank = not security protected Non-blank = security protected	Must be "spaces"
5	1	12	Volume Density	Density of file information on tape, blank or 00 = 556 bpi 1 = 200 bpi 2 = 800 bpi	Must be "spaces"
6	19	13-31	Reserved for future standardization	Must be "spaces"	Must be "spaces"
7	6	32-37	Reserved for future standardization	Must be "spaces"	Must be "spaces"
8	14	38-51	Owner Identification	Must be "spaces"	Must be "spaces"
9	28	52-79	Reserved for future standardization	Must be "spaces"	Must be "spaces"
10	1	80	Label Standard Level	Must be "spaces"	Must be 1

FILE HEADER LABEL

FIELD	LENGTH	POSITION	NAME	DESCRIPTION 1	DESCRIPTION 2
1	3	1-3	Label Identifier	Must be HDR	Must be HDR
2	1	4	Label Number	Must be 1	Must be 1
3	17	5-21	File Identifier	Any "a" characters which identify this file	Any "a" characters which identify this file
4	6	22-27	Multi-file Set Identification	Any "a" characters to identify the set of files of which this is one. This identification must be the same for all files of a multi-file set	Any "a" characters to identify the set of files of which this is one. This identification must be the same for all files of a multi-file set.
5	4	28-31	File Section Number	The file section number of the first header label of each file is 0001. This applies to the first or only file on a volume and to subsequent files on a multi-file volume. This field is incremented by one on each subsequent volume of the file.	The file section number of the first header label of each file is 0001. This applies to the first or only file on a volume and to subsequent files on a multi-file volume. This field is incremented by one on each subsequent volume of the file.

FILE HEADER LABEL (Con't.)

FIELD	LENGTH	POSITION	NAME	DESCRIPTION 1	DESCRIPTION 2
6	4	32-35	File Sequence Number	Four "n" characters denoting the sequence (that is, 0001, 0002, etc.) of files within the volume or set of volumes. In all the labels for a given file, this field will contain the same number.	Four "n" characters * denoting the sequence (that is, 0001, 0002, etc.) of files within the volume or set of volumes. In all the labels for a given file, this field will contain the same number.
7	4	36-39	Generation Number (optional)	Must be "spaces"	Four "n" characters denoting the current stage in the succession of one file generation by the next. When a file is first created, its generation number is 0001.
8	2	40-41	Generation Version Number (optional)	Two "n" characters distinguishing successive iterations of the same generation. The generation version number of the first attempt to produce a file is 00.	Two "n" characters distinguishing successive iterations of the same generation. The generation version number of the first attempt to produce a file is 00.

* Does not imply the support of multi-file volumes.

FILE HEADER LABEL (con't.)

FIELD	LENGTH	POSITION	NAME	DESCRIPTION 1	DESCRIPTION 2
9	6	42-47	Creation Date	A "space" followed by two "n" characters for the year, followed by three "n" characters for the day (001 to 366) within the year.	A "space" followed by two "n" characters for the year, followed by three "n" characters for the day (001 to 366) within the year.
10	6	48-53	Expiration Date	Same format as field 9. This file is regarded as "expired" when today's date is equal to, or later than, the date given in this field. When this condition is satisfied, the remainder of this volume may be overwritten.	Same format as field 9. This file is regarded as "expired" when today's date is equal to, or later than, the date given in this field. When this condition is satisfied, the ramainder of this volume may be overwritten.
11	1	54	Accessibility	Same as Field 4 of the Volume Header Label	Must be "spaces"
12	6	55-60	Block Count	Must be binary zero	Must be binary zero
13	13	61-73	System code (optional)	Must be "spaces"	Must be "spaces"
14	7	74-80	Reserved for future standardization	Must be "spaces"	Must be "spaces"

FILE TRAILER LABEL

FIELD	LENGTH	POSITION	NAME	DESCRIPTION 1	DESCRIPTION 2
1	3	1-3	Label Identifier	Must be EOF	Must be EOF
2	1	4	Label Number	Must be 1	Must be 1
3-11	50	5-54	Same as corresponding fields in File Header (optional)	Same as corresponding fields in the File Header Label	Same as corresponding fields in the File Header Label
12	6	55-60	Block Count	Six n characters, number of data blocks (including labels and tape marks) written since last File Header Label	Six "n" characters denoting the number of data blocks (exclusive of labels and tape marks) since the preceding HDR label group
13,14	20	61-80	Same as corresponding fields in the first file header label (optional)	Must be "spaces"	Must be "spaces"

VOLUME TRAILER LABEL

FIELD	LENGTH	POSITIONS	NAME	DESCRIPTION 1	DESCRIPTION 2
1	3	1-3	Label Identifier	Must be EOV	Must be EOV
2	1	4	Label Number	Must be 1	Must be 1
3-11	50	5-54	Same as corresponding field in the first file header label (optional)	Same as corresponding fields in File Header Label	Same as corresponding fields in File Header Label
12	6	55-60	Block Count	6 n characters, number of data blocks (excluding labels and tape marks) written since preceding volume label	Six "n" characters denoting the number of data blocks (exclusive of labels and tape marks) since the preceding HDR label group
13,14	20	61-80	Same as corresponding fields in file header label (optional)	Must be "spaces"	Must be "spaces"

6.3 3000 Series Computer Labels

The following 3000 Series Computer Labels are supported by 6000 SCOPE.

1. FILE HEADER

Field	Name	Length	Position	Description
1	Density	1	1	Density at which label & data are recorded. 2 = 200 bpi 5 = 556 bpi 8 = 800 bpi
2	Label ID	2	2-3	{ }
3	Logical Unit Number	2	4-5	Logical unit number; not checked by SCOPE
4	Retention Cycle	3	6-8	3 n characters specifying number of days tape is to be protected from accidental destruction.
5	File Name	14	9-22	Any a characters to identify this file.
6	Reel Number	2	23-24	2 n characters to denote which in a series of reels comprising the file
7	Date	6	25-30	Date file was created; MMDDYY.
8	Edition	2	31-32	2 n characters distinguishing successive iterations of the same file.
9	User Information	48	33-80	Any a characters; SCOPE does not check this field.

2. FILE TRAILER (EOF)

Field	Name	Length	Position	Description
1	Label ID	3	1-3	EOF
2	Block Count	5	4-8	5 n characters, number of data blocks written since file header label.
3	Unused	72	9-80	Unused

3. VOLUME TRAILER (EOT)

Field	Name	Length	Position	Description
1	Label ID	3	1-3	EOT
2	Block Count	5	4-8	5 n characters, number of data blocks written since file header label
3	Unused	72	9-80	Unused

6.4 Label Characteristics Summary

All labels are 80 characters in length.

	SCOPE Std.	3000 Series
7-track, 6SC3.2 and 6SC3.3		
Parity	Even	Even
Density	IP.LDEN	Same as file
Character Code	BCD	BCD
7-track, 6SC3.4		
Parity	Even	Even
Density	Same as file	Same as file
Character Code	BCD	BCD
7-track 7SC2.0		
Parity	Even	
Density	Same as file	
Character Code	BCD	
9-track 6SC3.3 and 6SC3.4		
Parity	Odd	Odd
Density	Same as file	Same as file
Character Code	USASCII or EBCDIC	USASCII or EBCDIC
9-track 7SC2.0		
Parity	Odd	
Density	Same as file	
Character Code	ANSI8, EBCDIC	

7.0 LABEL UTILITIES

The operating systems being considered which support tape labels provide the user with special macros and/or utility functions to assist in processing tape labels. The following list indicates which tools are available per operating system:

6SC3.2	Generation of FET Label Field Information macro
6SC3.3	Generation of FET Label Field Information macro Label Processing Utility Program, LABEL
6SC3.4	Multi-file Set List Utility, LISTMF Generation of FET Label Field Information macro Label Processing Utility Program, LABEL Multi-file Set List Utility, LISTMF
7SCV2	Label Information Generation macro/control card, LABEL

The following paragraphs describe the various utilities provided by the respective operating systems.

7.1 6SC3.2

To cause the 6SC3.2 operating system to process tape labels, the user must specify E or N on the REQUEST control card and supply certain label information in the FET at file OPEN time. To assist the user in inserting the label information in the 13 word FET, 6SC3.2 provides an FET label field generation macro called LABEL. The format of the macro is as follows:

```
lfn LABEL fln,ed,ret,create,reel,mfn,pos
```

The LABEL macro may be used to generate FET label field information. If used, the LABEL macro must immediately follow the FILEx macro to which it pertains. The LABEL macro is simply a data generation statement and does not directly cause any action on the file. The following descriptions indicate the values each parameter may assume. In all cases if the parameter is absent, the field is set to binary zero. Alphanumeric values are left-justified within the field; numeric values are right-justified within the field. If the parameter is smaller than the field size, the fill character used will be binary zero.

If FET label field information is generated by means other than the LABEL macro, display code zero may be used as the fill character in numeric fields and display code blank may be used as fill in alphanumeric fields.

File Label Name {fln}:	1-17 alphanumeric characters specifying file identification.
Edition Number {ed}:	1-2 digits specifying which version of the file.
Retention Cycle {ret}:	1-3 digits specifying number of days a tape is to be protected.
Creation Date {create}:	5 digits specifying creation date in Julian format {YYDDD}.
Reel Number {reel}:	1-4 digits specifying which reel of the file.
Multifile Name {mfn}:	1-6 alphanumeric characters specifying the logical set name of which the current file is a member.
Position Number {pos}:	1-3 digits specifying the relative position of the current file within a multi-file set.

7.2 6SC3.3

6SC3.3 provides the user with two label processing utility programs, LABEL and LISTMF and an FET label field information generation macro, LABEL. The LABEL utility program allows the user to define and process labels. The LISTMF utility allows the user to list the labels for member files of a multi-file set. The LABEL macro which generates FET label information is the same described in paragraph 7.1. The LABEL and LISTMF utility programs are described below.

LABEL,lfn,{R / W},Y,L=fln,V=reel,E=ed,T=ret,C=create,F=df,
M=mfn,P=pos,D=density,N=charset,X=dc.

The LABEL utility program enables users to write and to check labels and to position within a multi-file set. It is a central processor program, callable through standard program call cards, and has no special capabilities which are not available to any user program.

The following list defines briefly the meaning of parameters which may appear on the LABEL card. lfn is required and must be the first parameter.
The remaining parameters are optional, and may appear in any order.

lfn	Logical file name, 1-7 characters, lfn must be specified, and must be the first parameter.
R or W	Read or write label. If absent, W is assumed.
Y	Indicates that a 3000 series label
fln	File label name, 1-17 characters. If lfn is absent, a default name will be assumed for a New Tape and no checking will be done on an Existing Tape. Any special characters except $, . } { : may be used.
reel	Reel number, 1-4 digits. If reel is absent, reel 0001 is assumed for a New Tape and no checking will be done on an Existing Tape.

ed — Edition number, 1-2 digits. If ed is absent, edition 01 is assumed for a New Tape and no checking of edition number will be done on an existing tape.

ret — Retention cycle, 1-3 digits. If ret is absent, an installation-defined value will be written on a New Tape and no checking of retention cycle will be done on an Existing Tape. If ret equals 999, an expiration date of 99999 will be written in the label of a New Tape, effectively establishing permanent retention.

create — The 5-character Julian creation date. If create is absent, today's date will be used on a New Tape and no checking of creation date will be done on an Existing Tape.

df — The format of the data in the file.

absent	Data will be read/written in SCOPE standard format.
S	S-tape format.
L	L-tape format.

mfn — Multi-file set name, 1-6 characters, mfn is a required parameter on utility calls that process multi-file set tapes.

pos — Position number, 1-3 digits, of a file within a multi-file set.

density — The density at which data is to be recorded.

absent	installation-defined density.
LO	200 bpi/7-track tape
HI	556 bpi/7-track tape
HY	800 bpi/7-track tape
HD	800 cpi/9-track tape
PE	1600 cpi/9-track tape

charset — The character set that is to be used for coded data conversion on 9-track tapes.

absent	installation-defined default character set.
US	USASCII code
EB	EBCDIC code

dc — Disposition code

absent	no disposition code.
IU	inhibit unload
SV	save
CK	checkpoint dump tape
CI	checkpoint tape and inhibit unload
CS	checkpoint tape and save

Single-File (neither M nor P declared)

The LABEL utility program may be used to read and check (R is declared) or write (W is declared) standard SCOPE or 3000 Series Y labels. Normally LABEL will be the first reference to a file within a job. A REQUEST card is not needed for single-file tapes since the LABEL utility will issue a REQUEST function for a labeled tape (7 or 9 track) having the logical file name, data format, density, disposition code, and character set as specified on the LABEL control card. The LABEL utility will then build an FET using the label information as specified on the LABEL control card and issue an OPEN (with rewind) function. System default label parameters will be used whenever the LABEL card does not specify a parameter. The field content of the label actually written/read will be noted in the dayfile. The tape is left positioned at beginning of information. Any subsequent program wishing to inspect the label may do so by issuing an OPEN function using a 13 word FET.

Multi-File Tapes (M is declared, P may or may not be declared)

The LABEL utility program may be used to position and check (R is declared) or position and write (W is declared) standard SCOPE labels on a member file of a multi-file set. 3000 Series Y labels are not allowed for multi-file sets. The multi-file set must have been previously assigned by a REQUEST card/function. The LABEL utility will build an FET using the parameters as specified on the LABEL card and issue the appropriate OPEN function. The multi-file set will be

positioned as specified by the P parameter on the LABEL card. The field content of the label {read or written} will be recorded in the dayfile.

LISTMF LISTMF,M-mfn,P=p

LISTMF provides the ability to obtain a list of the contents of multi-file set tape, mfn. The multi-file set must have existing status.

LISTMF rewinds the multi-file set, mfn, and then positions to the p^{th} file by specifying position number p. The contents of the label are extracted from the buffer and written to the OUTPUT file. Each succeeding file is positioned specifying previous position number plus one. Files are positioned and labels are listed until "end-of-multi-file-set" status is returned {code 21_8}. The set is left positioned at end-of-set.

If p is absent, a value of one is assumed.

7.3 6SC3.4

6SC3.4 provides the user with the LABEL and LISTMF utility programs as described in Section 7.2 as well as the FET label field information generation macro as described in Section 7.1.

7.4 7SCV2

7SCV2 provides the user with the LABEL control card/macro. The LABEL control card/macro is a means of supplying label information which is used to check or create the HDR1,EOF1 and EOV1 labels on standard labelled files. When the LABEL control card or macro is encountered, the label information is stored in a label table. At file OPEN time the label information is used to process the HDR1 label. At file and volume CLOSE time, the label information is used to process the trailer labels.

The format and parameter description of the LABEL control card and macro is given below. The first parameter is lfn. The remaining parameters are optional and may appear in any order.

LABEL lfn, C=p2, D=p3, E=p4, G=p5, L=p6, M=p7,
P=p8, T=p9, U=p10, V=p11

KEYWORD	EXPLANATION
LFN	Logical file name {or an X register containing generalized core address of the FIT}.
C	Creation date. Parameter {p2}: Creation date in Julian format yyddd.
D	Checked or created. Parameter {p3}: R=check existing label, W=create non-existing label } I-O Seq. Files Only
E	Version number. Parameter {p4}: A 2 digit version number of the generation.
G	Generation number. Parameter {p5}: A 4 digit generation number.
L	File identifier. Parameter {p6}: A 1 to 17 character value enclosed by $ signs used to supply/check the associated HDR1 label field.
M	Multifile* name. Parameter {p7}: A 6 character name that associates all files of a file set.
P	File sequence number. Parameter {p8}: Gives the sequence number of a file in a multi-file set.
T	Expiration time. Parameter {p9}: A period in days from creation date in format dddd.
U	Expiration date. Parameter {p10}: An expiration date in Julian format yyddd.
V	File section number. Parameter {p11}: A 1 to 4 digit number that gives the volume no. of the file at which processing is to commence.

* Refers to ANSI standard but does not imply support of multi-file name as 6000 SCOPE.

8.0 END POINT PROCESSING

8.1 6000 SCOPE

End-of-File (Tape Mark) Procedures

A physical end-of-file (tape mark) can appear on a SCOPE standard tape only as part of a label, the WRITEF function writes a zero-length logical record with level 17. End-of-file marks may be written on X, L and S tapes with the WRITEF function. On labeled tapes, end-of-file marks are written as part of the label. On an input tape, the I/O system determines whether it is part of the label. For a SCOPE standard tape, the end-of-file mark indicates a label, since trailer labels are always written on SCOPE standard tapes (labeled or unlabeled). X tapes cannot be labeled. For S or L tapes, SCOPE determines whether a labeled tape has been declared. If so the I/O system determines if a label record is next. If so, end-of-reel or end-of-information procedures are performed. If the next record is not a label record or if the tape is not labeled, the end-of-file mark is treated the same as a zero-length record of level 17; the end-of-file bit is set in the FET status field and the function encountered is completed normally. Since X tapes cannot be labeled, when and end-of-file mark is read, the end-of-file bit is set in the FET status field and the function is completed.

End-of-Reel Procedures

End-of-reel procedures for an output tape are performed when the end-of-reel reflective spot is encountered according to the following table:

SCOPE Standard	y	y	y	y	n	n	n	n
Labeled	n	y	n	y	n	n	y	y
UP bit set	n	n	y	y	n	y	n	y
Backspace over last physical record	x	x						
Write EOV trailer label	x	x					x	
Write 4 tape marks					x			
Rewind unload reel	x	x			x		x	
Locate next reel	x	x			x		x	
Write header label		x					x	
Continue function	x	x			x		x	
Set end-of-reel bit in FET			x	x		x		x
Exit to user			x	x		x		x

y = yes

n = no

End-of-reel procedures for an input tape are performed when and EOV label is encountered on a labeled or SCOPE standard tape or when the end-of-reel reflective spot is encountered on an unlabeled tape. If the UP bit is not set the next reel is obtained, label checking is performed if the tape is labeled and the function continues normally on the next reel.

WARNING: When the UP bit is set and control is given to the user, the zero length PRU may not have been written if the longest record was an exact multiple of the PRU size. Any further writing may appear as part of the previous record. The user is responsible and should terminate his file with an end-of-file mark or some other action which will permit proper reading of the file.

End of Information Procedures

For an output tape, before backward motion takes place, an EOF trailer label is written on labeled or SCOPE standard tapes or four end-of-file marks are written on unlabeled tapes.

For an input tape, end of information is defined only for labeled and SCOPE standard tapes when the EOF trailer label is encountered. The end-of-information bit is set in the FET as long as the file remains positioned at the end of information. End of information for unlabeled tapes is not defined. It is the user's responsibility to determine by some other means when he has processed to the end of information.

8.2 7000 SCOPE 1.1

End of File - The TAPE control card and its associated directives provide the user with the capability to stage data to or from a magnetic tape on either a record or a file basis. The End of File indicator for a file that resides on a magnetic tape is defined by the Operator Station to be a hardware tapemark for an X or E format tape, a hardware tapemark or a level number of 17_8 for a 6000 SCOPE standard tape being read as an X format tape, and a short PRU for a Z format tape. Since all tape files are staged under 7SCV1.1, the end-of-file indicator on tape appears as an EOF control word on disk. When an EOF control word is sensed on a read function, the EOF status is returned to the user in the FET. The user may write an EOF control word with the WRITEF macro.

When the file is staged to tape, the EOF control word is converted, if necessary, to the proper end of file indicator as defined by the format.

When Z format file is staged out, the station writes a hardware tapemark after the file and leaves the tape positioned between the last physical record of the file (a short PRU) and the tapemark. When an X or E format file is staged out, 2 hardware tapemarks are written behind the data and the tape is left positioned between the tapemarks.

End of Reel - End-of-Reel processing is handled by the operators station. The central site is never aware of an end-of-reel condition. If the end-of-reel reflective spot

is sensed on a write, the reel is unloaded, and the last physical record is rewritten on the same unit once the next reel is mounted and Ready condition reoccurs. If the end-of-reel condition is sensed on a read function, the reel is unloaded, the last physical record is discarded and the function continues on the same unit once a new reel is mounted and the Ready condition reoccurs.

End-of-Information - End-of-Information is defined by the 7611-1 Operator Station to be a short PRU for the 7000 Format and a double hardware tapemark for the External Format. On input, the tape file may be staged through EOI. The EOI condition is indicated on the system disk as an EOI control word. When the EOI condition is sensed on a read operation or if the user attempts to read past the EOI indicator, EOI status is returned in the FET. On output, the system appends the EOI control word on the disk image of the file. When the file is staged out, the EOI control word causes the proper EOI indicator to be written on the tape.

8.3 7000 SCOPE 2

End of File - The end of file indicator is dependent upon the data format. Table 3 (Section 5.2) defines the EOF indicator for each of the unlabelled data formats. For labelled tape files, the EOF1 trailer label indicates EOI. Whenever the end of file condition is sensed, control is transferred to the user specified End of Data address +1. Prior to this transfer of control, the data manager stores a JP instruction in the user specified End of Data address. This JP instruction is constructed to pass control to the instruction following the macro call that detected the EOF condition, thus allowing

the user to return to normal program execution if desired. If no End of Data address is supplied by the user and the EOF condition is detected, control is returned to the user following the macro call giving rise to the EOF condition. In either case, the file position flags are set in the FIT. These status indicators may be retrieved from the FIT by use of the FETCH macro. The conditions which cause control to be transferred to the End of Data exit are:

End of Section

a) deleted W flag record (generated by WEOR macro on W format files and by 7-8-9 cards on an INPUT file.)

b) level number appendage less the 17B on files with C blocking and Z record types

End of Partition

c) single tapemark, or first tapemark of pair on unlabeled tape files

d) single tapemark not followed by EOV1 or EOF1 label on standard labelled tape file.

e) level 17 zero length block of files with C blocking and Z or S record types

f) non-deleted W flag record (as written by the ENDFILE macro for W type records and the 6-7-8-9 card on an INPUT file.

End of Information

g) a tapemark followed by an EOF1 label on standard labelled tape files.

h) second tapemark of pair on unlabelled tape file.

End of Volume - The End of Volume condition is automatically processed by the system in all cases except two. The two exceptional cases are a) reading an unlabelled on-line tape and b) reading an unlabelled staged tape. Control is transferred to the user End of Data address +1 with the appropriate file position indicators set in the FIT whenever a double tapemark is detected. The user must know if this double tapemark is End of Volume or End of Information (i.e., Is the file a multi-reel file, or a single-reel file.) If the file is a single-reel file, the user would issue a normal CLOSE FILE request. If the file is a multi-reel file, the user must issue a CLOSE VOLUME request to cause the data manager to request the next reel to continue the read operation.

Automatic End of Volume processing involves the following: For labelled files, the end of volume label group is written/checked, the volume is positioned as indicated by the user, the next volume is requested, the header label group is written/checked and control is returned to the user following the write/read operation which reached the end of volume condition. For unlabelled files, two hardware tapemarks terminating the current volume are written, the volume is positioned as indicated by the user, the next volume is requested and control is returned to the user following the read/write operation which reached the end-of-volume condition.

The user program is completely unaware of the end of volume condition when reading or writing staged labelled tape files, writing staged unlabelled tape file, reading or writing labelled on-line tape files and writing unlabelled on-line tape files. When reading an unlabelled tape file (staged or on-line), a double tapemark condition must be processed by the user.

End of Information- The End of Information indicator is dependent upon the data format. For labelled tape files, the EOF1 label group indicates EOI. For unlabelled tape files, Table 3 indicates the EOI indicator. For staged tape files, the EOI indicator will appear on the system disk as a control word. Whenever EOI is sensed, the file position field of the FIT is set appropriately and control is transferred to the user in the same manner as when the EOF condition is sensed.

9.0 9 TRACK TAPES

The level to which 9 track tapes are supported by the operating systems being considered varies per operating system. 6SC3.2 and 7SCV1.1 do not support 9 track tapes. 6SC3.3 and 6SC3.4 support 9 track tapes in the same manner except that 6SC3.4 will allow dumping and loading of permanent files to/from 9 track tape. 6SC3.4 will incorporate more extensive error recovery techniques. 7SCV2 will support 9 track tapes at the same level as 7 track tapes provided certain precautions are adhered to. Paragraphs 9.1 and 9.2 discuss 9 track tape characteristics under 6SC3.3/6SC3.4 and 7SCV2 respectively.

9.1 Characteristics of 1/2-inch 9-track magnetic tape, 6SC3.3 and 6SC3.4

Densities:

Only two densities, 800 BPI* and 1600 BPI, may be used on 9-track tapes. The density can be set only at load point so that the density of the file will be the same as the density of the first record read or written.

Modes:

In 9-track tapes, mode is a term that is used to describe the intrinsic structure of the data and, unlike 7-track tapes, does not imply a change in tape parity. All 9-track tapes are recorded in odd parity. The two modes supported by SCOPE are conversion mode and packed mode. Conversion mode allows the CP program to use display code internally, since the 6-bit display code characters will be mapped {reversibly} into 8-bit bytes for transcription on tape. Packed mode transfers 8-bit bytes directly to central memory at a density of 7 1/2 bytes per CM word with no conversion or alteration performed on the data.

Packed mode writing assumes that the buffer contains 8-bit bytes packed 7 1/2 per CM word.

Character Sets

The two character sets provided by SCOPE for use with conversion mode I/O are 64 character subsets of USASCII and EBCDIC. On a write operation, each member of the 64 character set is mapped to the corresponding character in the 64 character subset of EBCDIC or USASCII. On a read operation, each member of the 8 bit character set is mapped to one of the 64 DISPLAY code characters.

Labels

6000 and 3000 labels will be supported on 9-track tapes. However, the density of the labels will be the same as the file density. Since conversion mode is used in label processing all labels will appear in CM as 80 6-bit display code characters and on tape as 80 8-bit bytes in either USASCII or EBCDIC.

Data Format

Data on 9-track tape will be analogous to that of "S" style 7-track tapes. In conversion mode (similar to 7-track BCD) the record length may be any even number ranging from just above noise size up to 5120 display code characters. Thus a record of 512 CM words will be represented on tape by 5120 8-bit bytes. The 6 bit to/from 8 bit code conversion is performed in the tape controller.

* BPI stands for "8-bit bytes per inch".

In packed mode (similar to 7-track binary) the record length may range from just above noise size up to 3840 8-bit bytes. Thus a record of 512 CM words will be represented on tape by 384 8-bit bytes. On all 9-track tapes, an end-of-file is represented by a hardware file mark.

Macros

Any macro that can be used on a Stranger tape can also be used on a 9-track tape. For conversion mode I/O, all fields and parameters have the same meaning that they have on 7-track "S" tapes. For packed mode I/O the UBC and MLR fields in the FET will be used to denote the number of 8-bit bytes in a record. The same is true of the unused bit field and word count field in the header words of READN and WRITEN records. The mode of the request is conversion mode if bit 1 is 0 in the function code, and it is packed mode if bit 1 is 1 in the function code.

9.2 7SCV2 9 Track Tape Characteristics

Densities:

Two densities are supported: 800 BPI and 1600 BPI. The density setting as specified on the REQUEST or STAGE control card is used when writing a file. The hardware automatically selects the proper density setting for a read operation.

Character Sets:

Two external 8 bit character sets are supported for 9 track tape. They are EBCDIC and ANSI8. The character set is specified on the FILE control card by the EC parameter.

Labels:

Standard ANSI labels will be supported for single file/single reel and single-file/multi-reel tapes. Paragraph 6.4 describes the physical characteristics of the labels.

Macros:

All macros which apply to 7-track tapes apply as well to 9-track tapes. Macros such as LABEL, FILE, GET, PUT,--- are applicable to 9-track tapes.

Data Formats/Mode:

Two modes of writing/reading 9-track tapes are supported under 7SCV2. They are conversion mode and non-conversion (or binary) mode. A third mode which may be refered to as an 8-bit internal mode may be used as long as certain restrictions are observed.

Conversion Mode:

Conversion mode causes character conversion between an internal 6 bit code of DISPLAY (in user's SCM buffer) and an external 8-bit code of either EBCDIC or ANSI8 (8 bit code physically on tape). Conversion mode is selected by specifying CM=yes and EC=EBCDIC or ANSI8 on the FILE control card or macro. In conversion mode, 7SCV2 can write only a 64 character subset of EBCDIC and ANSI8, but can read the full 256 character set. (On a read, each of the 256 characters is mapped to one of the 64 character subset characters). The many combinations of record type and block type are supported for 9-track conversion mode. Since device drivers do not recognize logical record boundaries nor block formats, the conversion of block and record formats which contain binary control information does not produce meaningful results as recorded on tape. However, when the data is reconverted on the read operation, it regains its meaningfulness.

Binary Mode:

Binary mode causes no character code conversion to occur when data is transfered to or from the 9 track tape. The user selects binary mode by specifying CM=NO and by not specifying the EC parameter on the FILE control card or macro. Data appears in the user's buffer as 60 significant bits per word, i.e 7 1/2 8-bit frames are written from or read to each 60 bit word.

The operating system does not prohibit the use of the many combinations of record type and block type. However, to ensure that meaningful transfers are performed, certain restrictions and considerations must be adhered to. These restructions and considerations are:

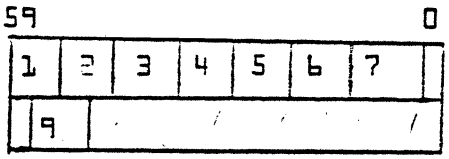
1. Physical blocks must be of a length that is an integral multiple of 8 bits.
2. Physical blocks, logical records and partial records must be of a length which is a multiple of 6 bits. (The data manager works with 6 bit character or 60 bit word lengths only).
3. To satisfy the above requirements, the user must compensate for the special block and record processing provided by the data manager in such cases as: a) prefixing control word and trailing padding for W format records b) trailing blank suppression and zero byte insertion for Z format records, c) prefixing control word for I format blocks d) short block generation for C and I block formats at logical and physical ENDFILE.

Therefore, it is recommended that the user read/write logical records which are an integral number of 60 bit words. This is the case for S,W, and X record formats. Whenever the physical block length is an even number of 60 bit words (or an even number plus the 48 bit S format level number appendage) the data can be written on the tape as an integral number of 8 bit frames. Whenever the physical record length is an odd number of 60 bit words (or odd number plus 48 bits) an extra

4 bits of padding is supplied by the driver to allow an integral number of 8 bit frames to be written. When this physical record is read, the data manager drops the 4 bits of padding.

If the user wishs to write a logical record which is not a multiple of a 60 bit, he must carefully adhere to the above restrictions. Consider the following two simple examples.

a) Suppose the user wishes to write a 9-track tape file with the following characteristics: fixed length logical records of 9 8-bit characters (12 6-bit characters) in length, and 1 logical record per physical record. The user's FILE control card or macro must specify BT=K, RB=1,RT=F,FL=12,CM=N0. The user's REQUEST or STAGE control card must specify NT, density (HT or PE), parity (EVEN or ODD) and other desired parameters. The user would issue a PUT macro specifying lfn and user record/data area. The data manager would pick up the first 2 60-bit words of the user's record area and send them (immediately or at stage out time) to the PP with an unused bit count of 48. The PP would in turn write the 6X12=72 bits of data as 9 8 bit frames as indicated below.



Character number 8 is contained in bits 0 thru 3 of the first word and 56 thru 59 of the second word.

When this physical record is read, the reverse operation occurs: The PP sends the data manager 2 60-bit words with an unused bit count of 48, the data manager passes the data to the users buffer with the record length field of the FIT set to 12. (12x6=72=9x8)

b) Suppose the user wishes to write a logical record whose length is 16 8-bit frames, i.e. 128 bits. Since the data manager requires character lengths to be a multiple of 6 bits, and 128 is not a multiple of 6, the user cannot use the PUT and GET macros to process this size physical record.

Eight-Bit Internal Mode

The 7SCV2 data manager does not formally recognize internal codes of 8 bits. However, the user is able to transfer such data if he adheres to certain restrictions.

1. The FILE control card or macro must specify CM=NO and EC=ANSI8/EBCDIC.
2. The data must be represented internally in bits 55-0 of each 60 bit word. Bits 59-56 are unused.
3. The file must be defined as BT=K, RB=1. If the user wishes to block logical records, the user must perform the blocking.
4. If partial transfers are to be used (GETP, PUTP), the transfers must be for a whole number of 60 bit words.
5. The length of each physical block must be an integral number of 8 bit characters (excluding high order padding).

6. The length of each physical block must be an integral number of 6 bit characters (including high order padding in each word).

7. Restrictions 5 and 6 imply that blocks must be 7n+0,1,4 8-bit character (which can be expressed as 10m+0,2 or 6 bit characters).

No character code conversion occurs under 8-bit internal mode.

The following example may best explain the manner in which the 8 bit internal mode is used.

Suppose the user wishes to write a physical record of 15 8-bit frames. The FILE control card/macro must specify RT=F,FL=22,BT=K,RB=1,CM=N0,EC=ANSI8/EBCDIC. Before the user issues the PUT macro his data area would appear as the following:

59 56	48	40	32	24	16	8	0	
''//	1	2	3	4	5	6	7	word 1
//''	8	9	10	11	12	13	14	word 2
.////	15							word 3

The data manager will pass the 3 60 bit words along with an unused bit count equal to 48 (22X6=132, remember that the data manager always deals with 6-bit character lengths) to the PP. As a result of CM=N0 and EC=ANSI8/EBCDIC on the FILE control card, the PP driver expects the data to be in the format shown above (i.e., bit 59-56 are unused). As a result, the PP will write the 15 8-bit characters as shown above. Upon reading the file, the reverse operation will occur if the file is defined in the same manner.

APPENDICES

APPENDIX A

CHARACTER SETS

DISPLAY	CHARACTER	HOLLERITH (026)	HOLLERITH (029)	BCD Internal	BCD External	ANSI8	EBCDIC
00	: Note 1	8-2	8-2	12		72	172
01	A	12-1	12-1	21	61	101	301
02	B	12-2	12-2	22	62	102	302
03	C	12-3	12-3	23	63	103	303
04	D	12-4	12-4	24	64	104	304
05	E	12-5	12-5	25	65	105	305
06	F	12-6	12-6	26	66	106	306
07	G	12-7	12-7	27	67	107	307
10	H	12-8	12-8	30	70	110	310
11	I	12-9	12-9	31	71	111	311
12	J	11-1	11-1	41	41	112	321
13	K	11-2	11-2	42	42	113	322
14	L	11-3	11-3	43	43	114	323
15	M	11-4	11-4	44	44	115	324
16	N	11-5	11-5	45	45	116	325
17	O	11-6	11-6	46	46	117	326
20	P	11-7	11-7	47	47	120	327
21	Q	11-8	11-8	50	50	121	330
22	R	11-9	11-9	51	51	122	331
23	S	0-2	0-2	62	22	123	342
24	T	0-3	0-3	63	23	124	343
25	U	0-4	0-4	64	24	125	344
26	V	0-5	0-5	65	25	126	345
27	W	0-6	0-6	66	26	127	346
30	X	0-7	0-7	67	27	130	347
31	Y	0-8	0-8	70	30	131	350
32	Z	0-9	0-9	71	31	132	351
33	0	0	0	00	12	60	360
34	1	1	1	01	01	61	361
35	2	2	2	02	02	62	362
36	3	3	3	03	03	63	363
37	4	4	4	04	04	64	364
40	5	5	5	05	05	65	365
41	6	6	6	06	06	66	366
42	7	7	7	07	07	67	367
43	8	8	8	10	10	70	370
44	9	9	9	11	11	71	371
45	+	12	12-8-6	20	60	53	116
46	-	11	11	40	40	55	140
47	*	11-8-4	11-8-4	54	54	52	134
50	/	0-1	0-1	61	21	57	141
51	(	0-8-4	12-8-5	74	34	50	115
52	)	12-8-4	11-8-5	34	74	51	135
53	$	11-8-3	11-8-3	53	53	44	133
54	=	8-3	8-6	13	13	75	176
55	space	blank	blank	60	20	40	100
56	,	0-8-3	0-8-3	73	33	54	153
57	.	12-8-3	12-8-3	33	73	56	113

DISPLAY	CHARACTER	HOLLERITH	HOLLERITH	BCD	BCD	ANSI8	EBCDIC
60	≡	0-8-6	8-3	76	36	43	173
61	£	8-7	8-5	17	17	47	175
62	⊐	0-8-2	12-8-7	72	32	41	117
63	% Note 2	8-6	0-8-4	16	16	45	154
64	≠	8-4	8-7	14	14	42	177
65	→	0-8-5	0-8-5	75	35	137	155
66	∨	11 or 11-8-2	11 or 11-8-2	52	52	135	320
67	∧	0-8-7	12	77	37	46	120
70	↑	11-8-5	8-4	55	55	100	174
71	↓	11-8-6	0-8-7	56	56	77	157
72	<	12-0 or 12-8-2	12-0 or 12-8-2	32	72	133	300
73	>	11-8-7	0-8-6	57	57	76	156
74	≤	8-5	12-8-4	15	15	74	114
75	≥	12-8-5	0-8-2	35	75	134	340
76	¬	12-8-6	11-8-7	36	76	136	137
77	;	12-8-7	11-8-6	37	77	73	136

NOTE 1: The following table indicates the printed character for a 00_8 code.

6SC3.2	space
6SC3.3	:
6SC3.4	:
7SC1.1	:
7SC2.0	:

NOTE 2: The following table indicates the printer character for a 63_8 code.

6SC3.2	:
6SC3.3	%
6SC3.4	%
7SC1.1	%
7SC2.0	%

APPENDIX B: RECORD TYPE/BLOCK TYPE EXAMPLES

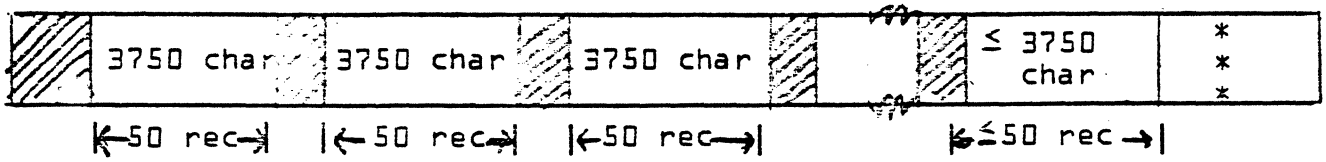
RECORD TYPE: F

K BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=K, RB=50, RT=F, FL=75.
```

Associated Tape Format:

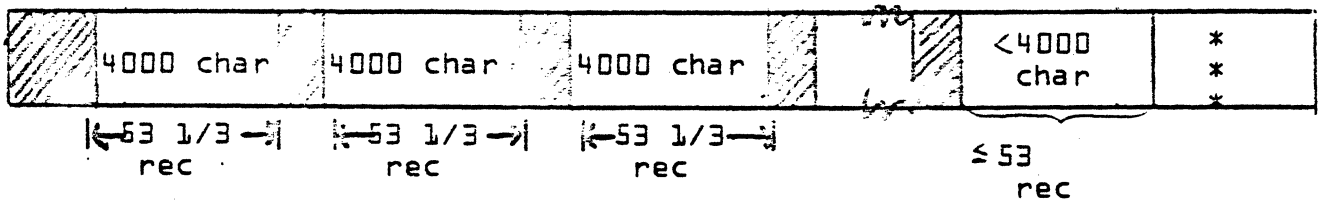


C BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=C, MBL=4000, RT=F, FL=75.
```

Associated Tape Format:

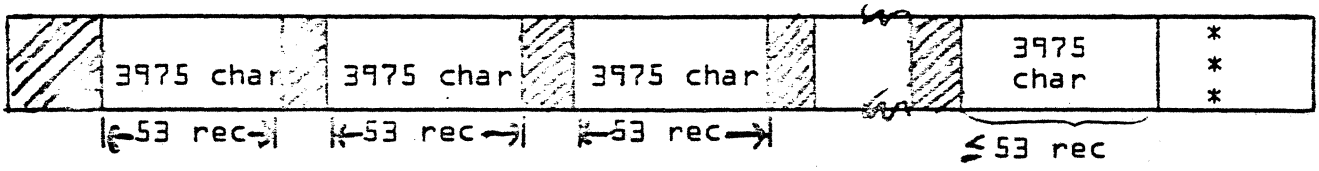


E BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=E, MBL=4000, RT=F, FL=75.
```

Associated Tape Format:



RECORD TYPE: B

Assume:

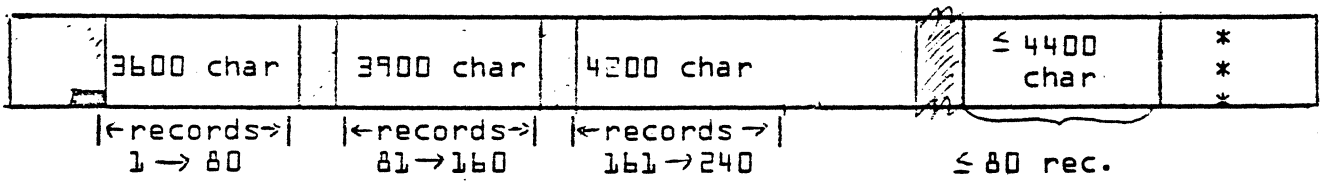
Record number	Record length
1 → 100	45 chars
101 → 200	50 chars
over 200	55 chars

K BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=K, RB=80, RT=B, MRL=55.
```

Associated Tape Format:

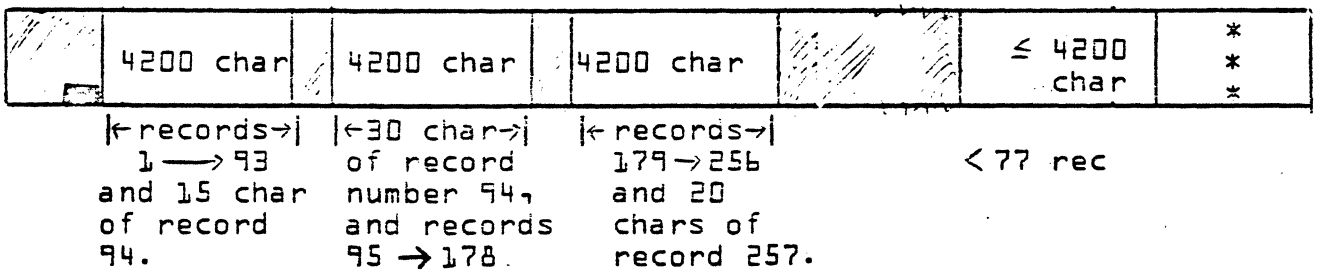


C BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=C, MBL=4200, RT=B, MRL=55.
```

Associated Tape Format:

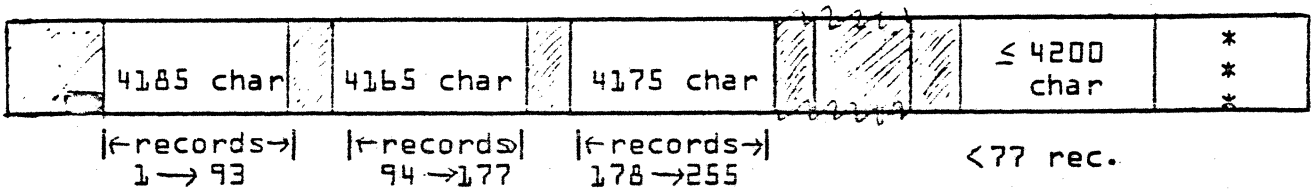


E BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=E, MBL=4200, RT=B, MRL=55.
```

Associated Tape Format:



RECORD TYPE: D

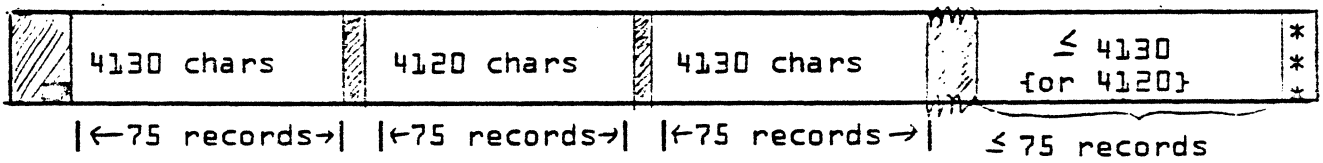
For simplicity assume: even numbered records are 50 characters in length and odd numbered records are 60 characters in length. Record numbers begin with 1; lengths include length field.

K BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=K, RB=75, RT=D, MRL=80, LP=0, LL=4.
```

Associated Tape Format:



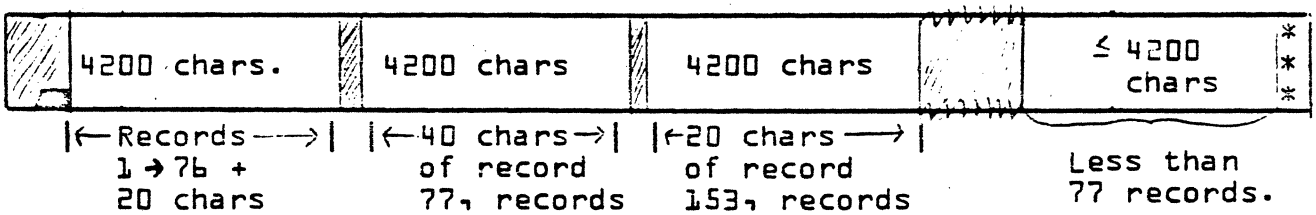
NOTE: The first 2 characters of each logical record specifies the length of the record.

C BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=C, MBL=4200, RT=D, MRL=60, LP=0, LL=4.
```

Associated Tape Format:

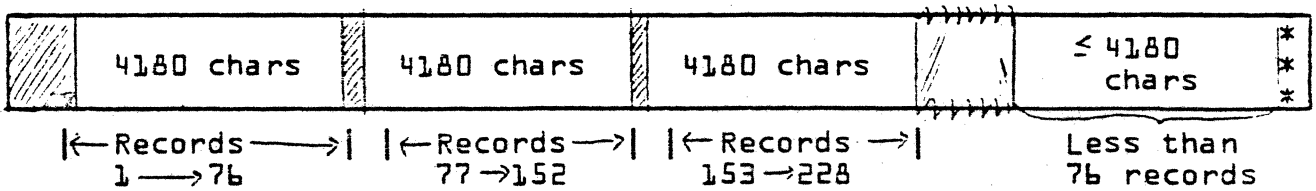


E BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=E, MBL=4800, RT=D, MRL=60, LP=0, LL=4.
```

Associated Tape Format:



RECORD TYPE: R

Assume:
Record Number
1 ⟶ 25
26 ⟶ 50
over 50

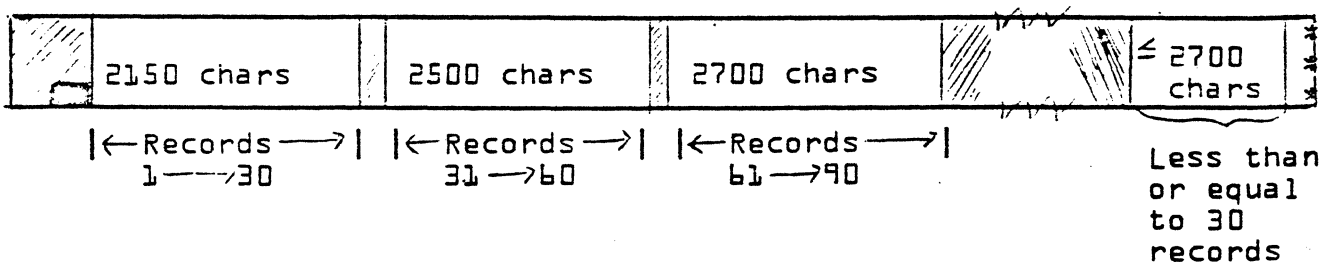
Record Length
69 char + RMK
79 char + RMK
89 char + RMK
chars

K BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=K, RB=30, RT=R, MRL=90, MNR=69, RMK= 53B.
```

Associated Tape Format:



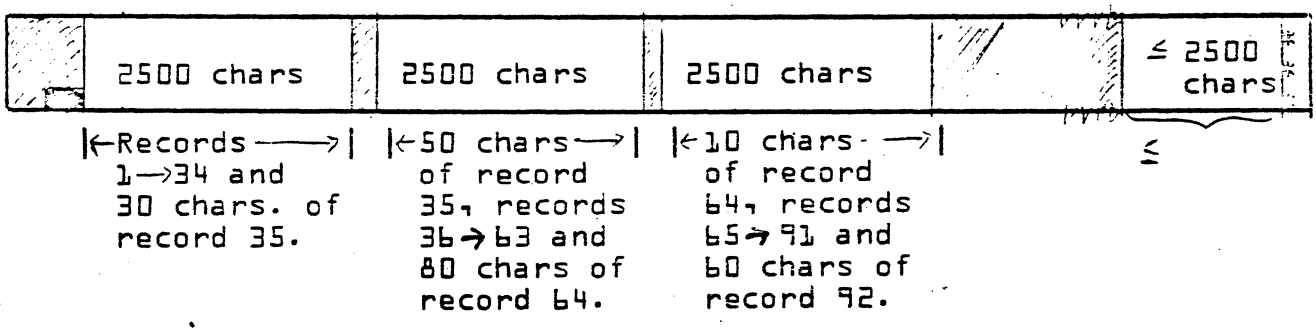
NOTE: In all 3 of these examples, the RMK characters are included in the figures indicating number of characters.

C BLOCKING

FILE Cotnrol Card:

```
/FILE MYFILE, BT=C, MBL=2500, RT=R, MRL=90, MNR=69, RMK=53B.
```

Associated Tape Format:

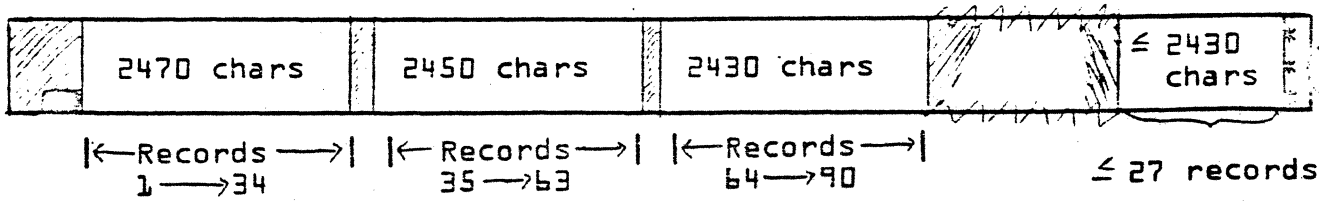


E BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=E, MBL=2500, RT=R, MRL=90, MNR=69, RMK=53B.
```

Associated Tape Format:



RECORD TYPE: T

Assume:

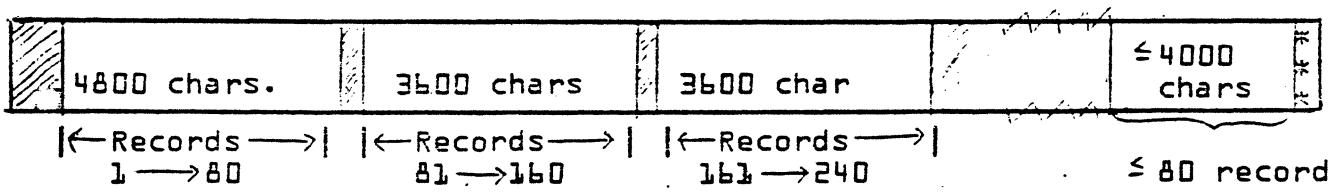
Record Number	Number of Trailer Items
1 → 100	4
101 → 200	2
over 200	3

K BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=K, RB=80, RT=T, MRL=60, CP=2, CL=4, HL=20, TL=10.
```

Associated Tape Format:

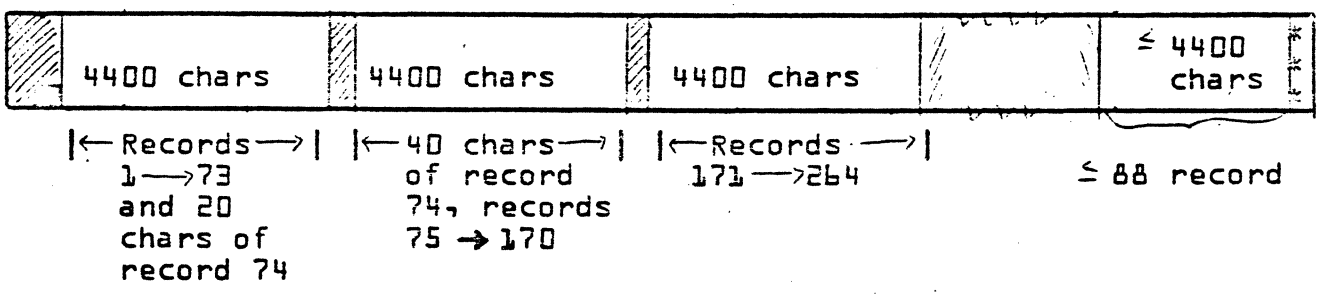


C BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=C, MBL=4400, RT=T, MRL=60, CP=2, CL=4, HL=20, TL=10.
```

Associated Tape Format:

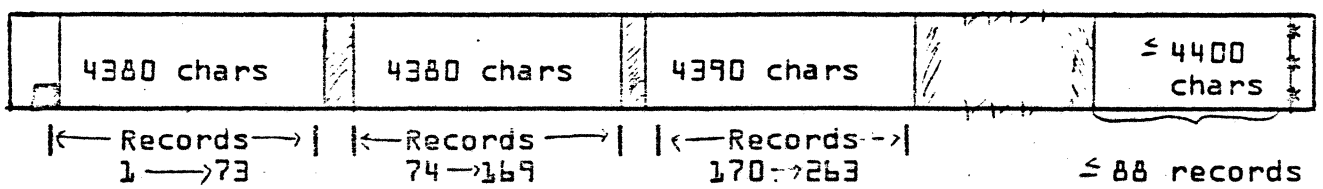


E BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=E, MBL=4400, RT=T, MRL=60, CP=2, CL=4, HL=20, TL=10.
```

Associated Tape Format:



RECORD TYPE: U

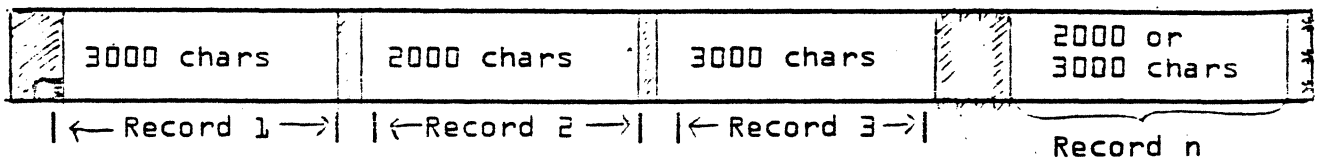
Assume: even numbered records are 2000 chars, odd numbered records are 3000 chars, record numbers begin with 1.

K BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=K, RB=1, RT=U, MRL=3000.
```

Associated Tape Format:

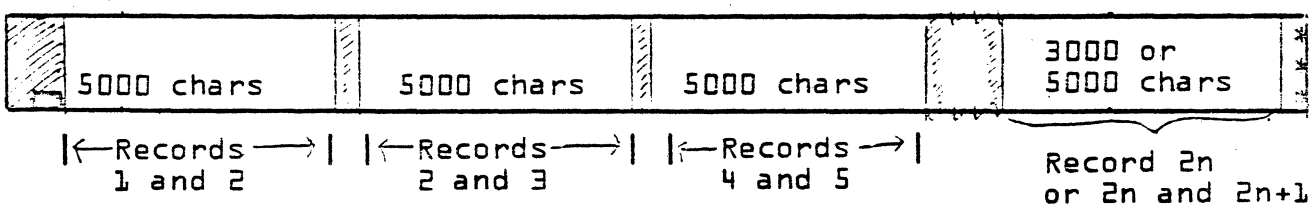


C BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=C, MBL=5000, RT=U, MRL=3000.
```

Associated Tape Format:

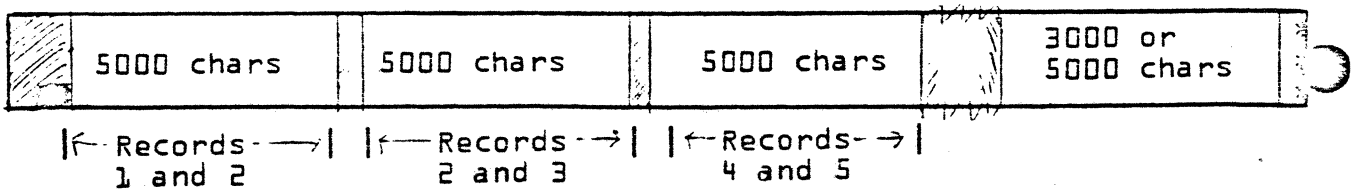


E BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=E, MBL=6000, RT=U, MRL=3000.
```

Associated Tape Format:



NOTE: In all 3 of the examples below, the figures indicating number of characters includes the 12 bit zero bytes. (12 bits of zero = 2 characters).

RECORD TYPE: Z

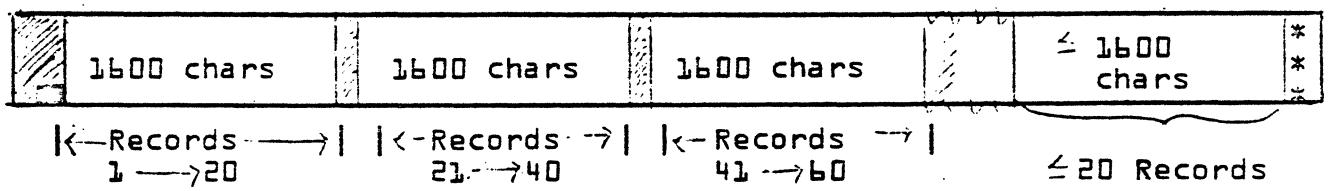
Assume: All records are 80 characters (78 characters of data and 2 characters of zero), last data character is non-blank.

K BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=K, RB=20, RT=Z, FL=78.
```

Associated Tape Format:

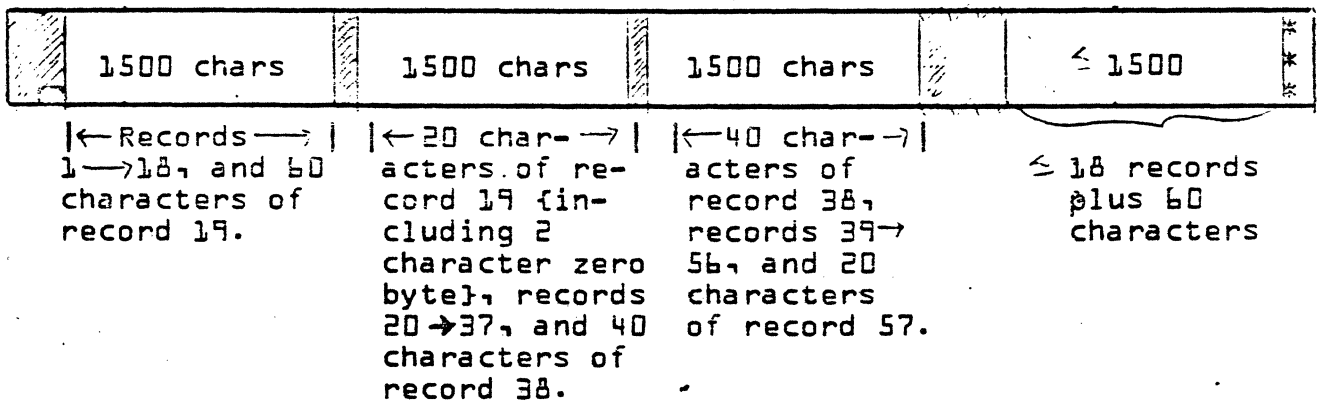


C BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=C, MBL=1500, RT=Z, FL=78.
```

Associated Tape Format:

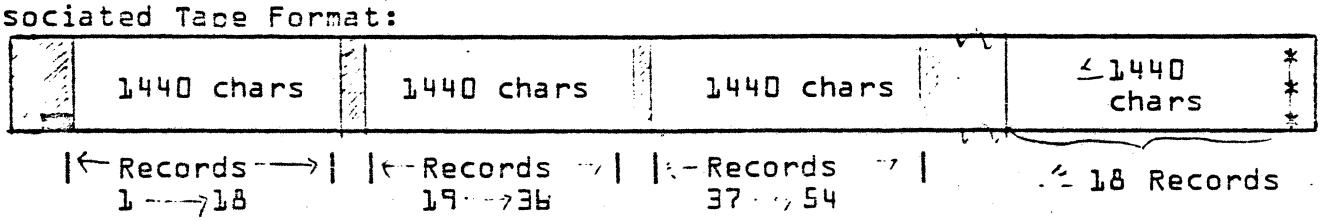


E BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=E, MBL=1500, RT=Z, FL=78.
```

Associated Tape Format:



RECORD TYPE: W

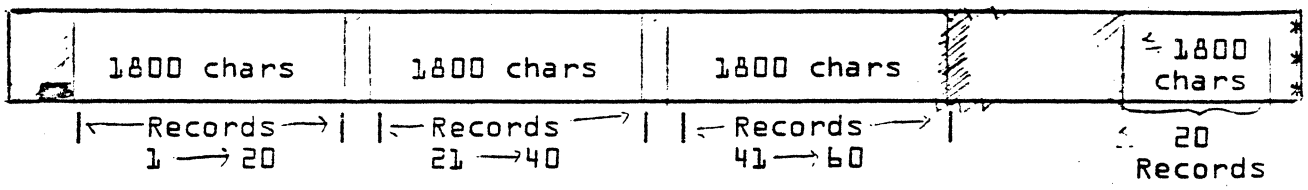
Assume: All records are 75 characters in length not including the control word ➡ 90 characters are needed to represent the record.

K BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=K, RB=20, RT=W, MRL=75.
```

Associated Tape Format:

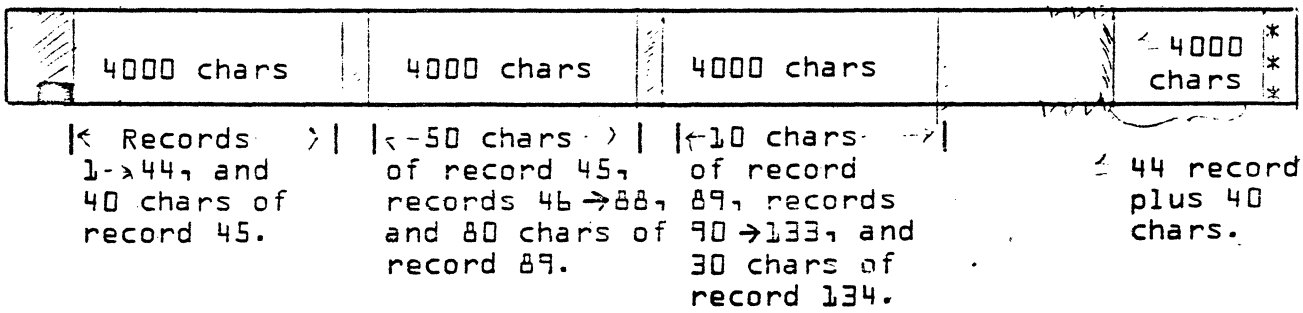


C BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=C, MBL=4000, RT=W, MRL=75.
```

Associated Tape Format:

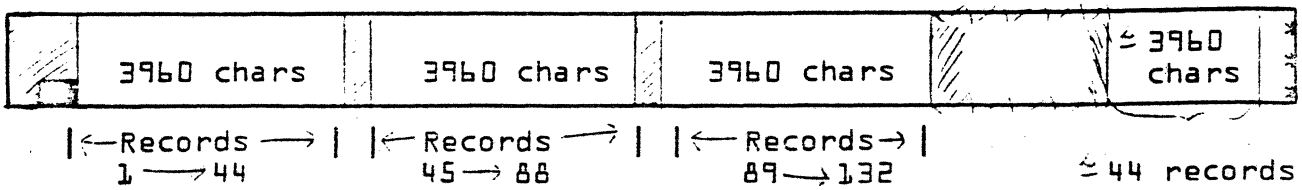


E BLOCKING

FILE Control Card:

```
/FILE MYFILE, BT=E, MBL=4000, RT=W, MRL=75.
```

Associated Tape Format:



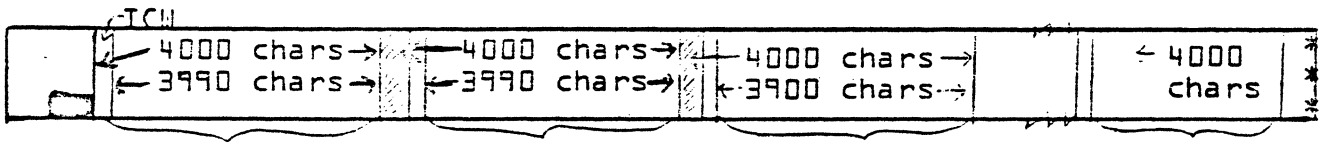
RECORD TYPE: W

BLOCKING TYPE: I

FILE Control Card:

```
/FILE MYFILE, BT=I, MBL=4000, RT=W, MRL=75.
```

Associated Tape Format:



Records 1→44, and 30 chars of record 45

60 chars of record 45, records 46→88 and 60 chars of record 89.

30 chars of record 89 and records 90→133.

44 record plus 30 chars.

Assume the file is composed
of 3 records.
record 1 = 5120 chars.
record 2 = 4600 chars.
record 3 = 1000 chars.

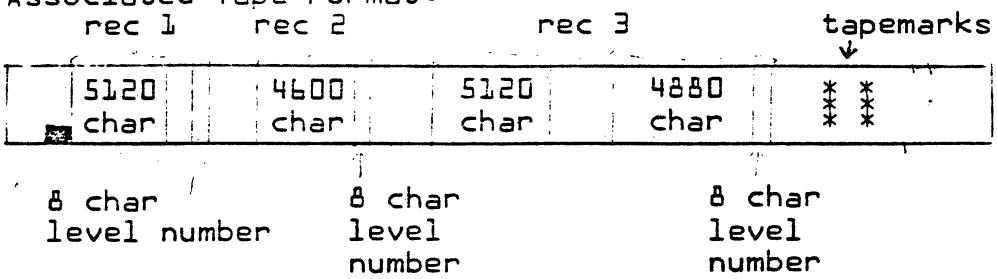
RECORD TYPE: S

C BLOCKING ONLY

FILE Control Card:

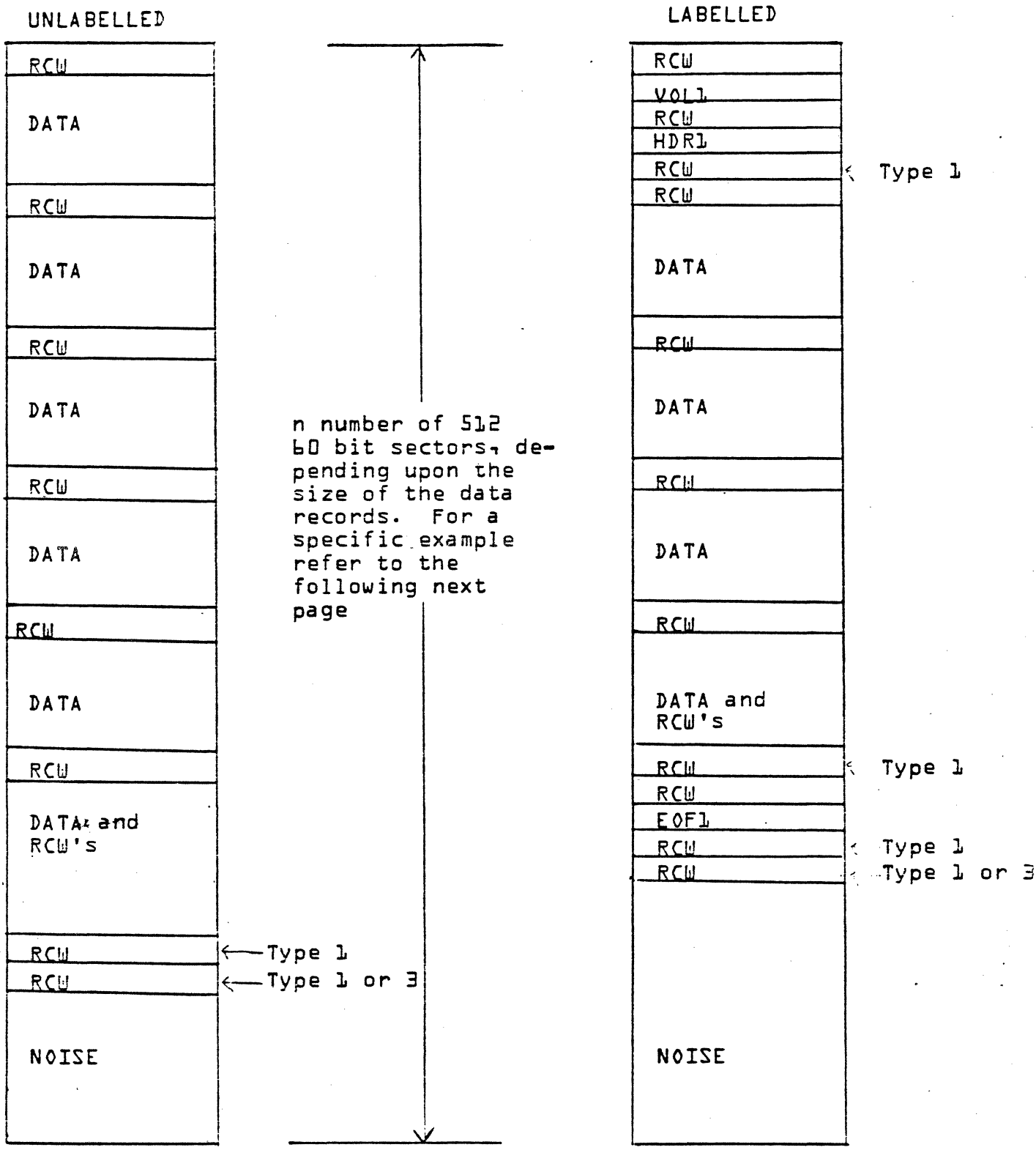
```
FILE MYFILE, BT=C, MBL=5120, RT=S.
```

Associated Tape Format:



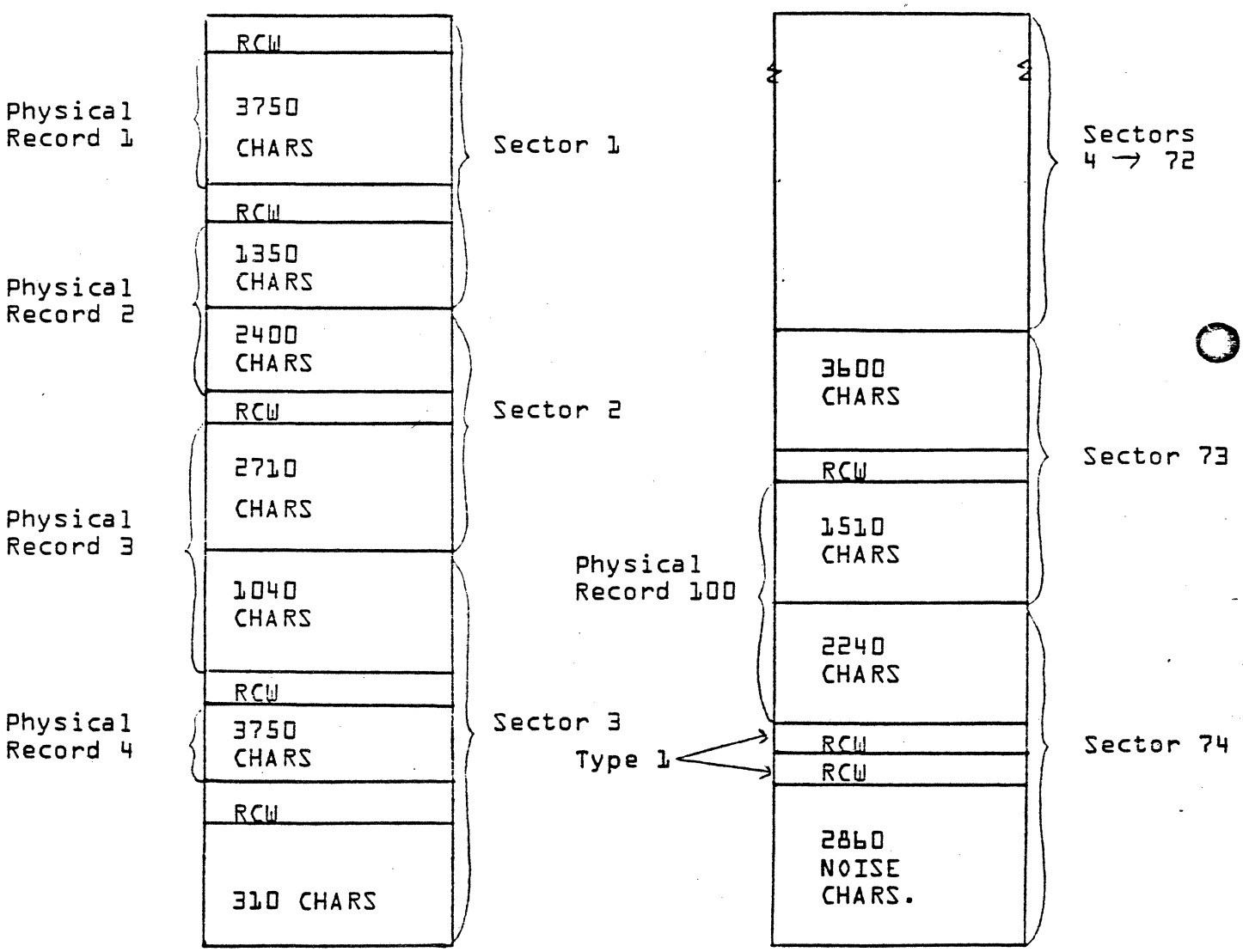
APPENDIX C

MASS STORAGE IMAGE OF 7SCV2 STAGED TAPE FILE

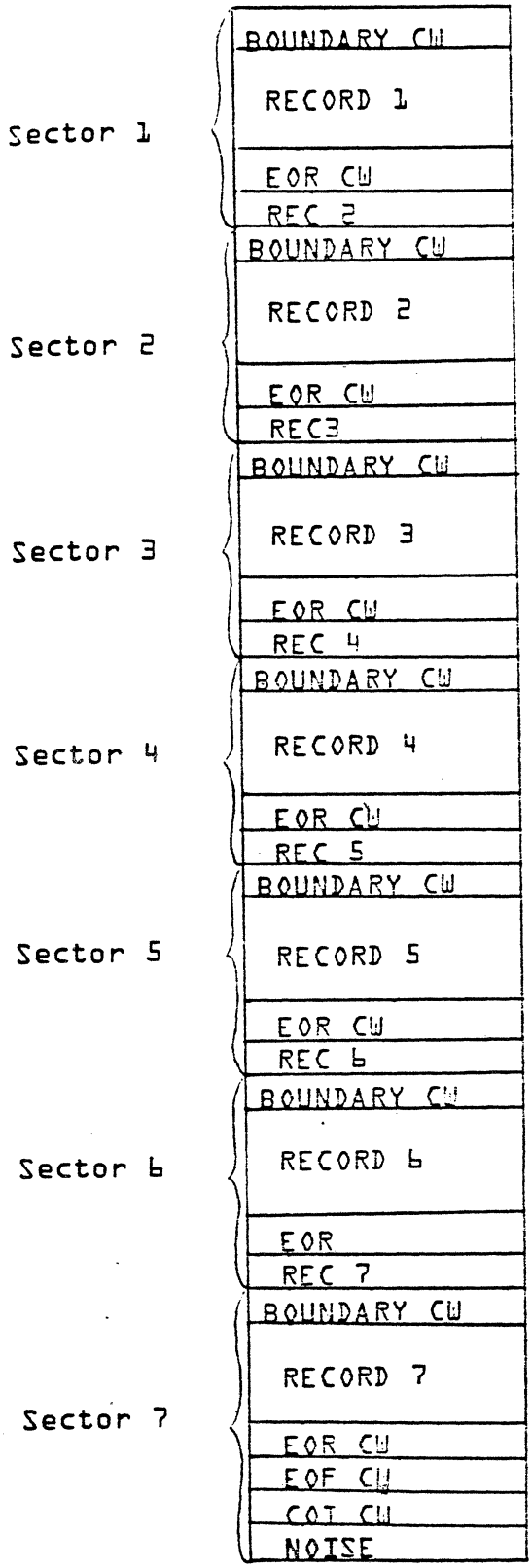


EXAMPLE OF MASS STORAGE IMAGE OF A 7SCV2 TAPE FILE

The following example illustrates the mass storage image of the tape file illustrated in the first example in Appendix B. BT=K, RB=50, RT=F, FL=75. Assume the file contains 5000 logical records. The mass storage image would be the following: A sector is 512 60 bit words.

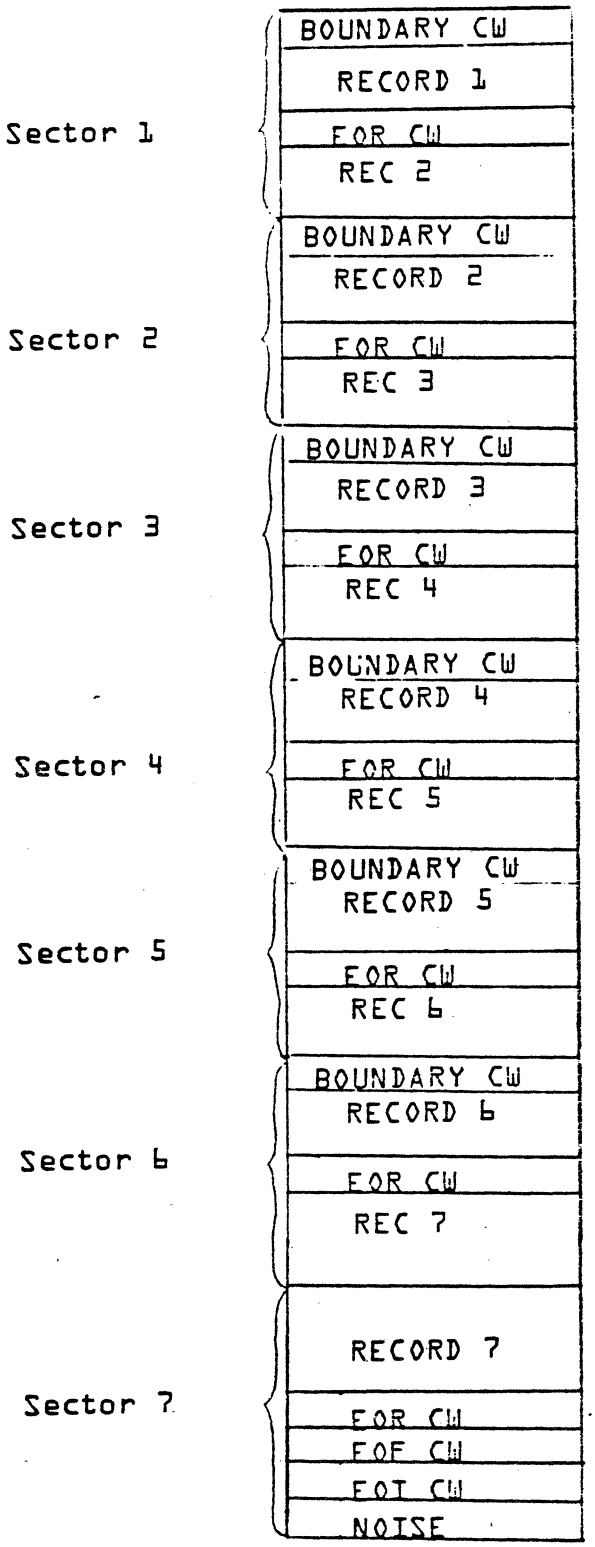


7SCV1 MASS STORAGE IMAGE OF STAGED TAPE FILES



7000 FORMAT FILE

The file is composed of 8 5000 character records. Since each physical block is 512 60 bit words in length, the first word of each sector is a Boundary CW.



EXTERNAL MODE BINARY

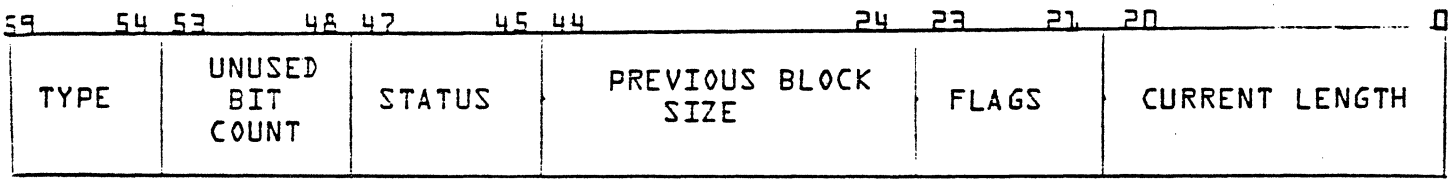
The file is composed of 8 5000 charact logical records. Since each logical (and physical) record is 5000 characters, 502 60 bit words are needed to represent each logical record. The Boundary CW is not the first word in each sector.

APPENDIX D

7000 SCOPE V 2

CONTROL WORD FORMATS

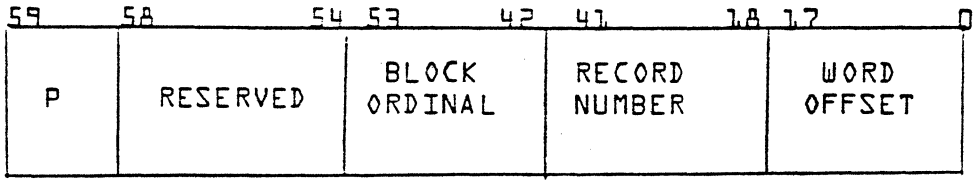
RCW Recovery Control Word (prefixes blocks on Mass Storage files in staged tape format.)



Bits 57-54 TYPE
- 0 - physical record
- 1 - hardware filemark
- 3 - second hardware filemark of a pair (EOV/EOI)

53-48 Unused bit count in last 60 bit word of block
47 Hardware malfunction status
46 Unrecoverable parity error status
45 Lost data error status
44-24 Length of previous block in 60 bit words including RCW
23 Recovered parity error status
22 End of volume reflective spot during read
21 reserved
20-0 Length of this block in 60 bit words excluding RCW. This will always be zero for types 1 and 3.

ICW I - blocking Control Word



Bits 59 odd parity bit

58-54 reserved

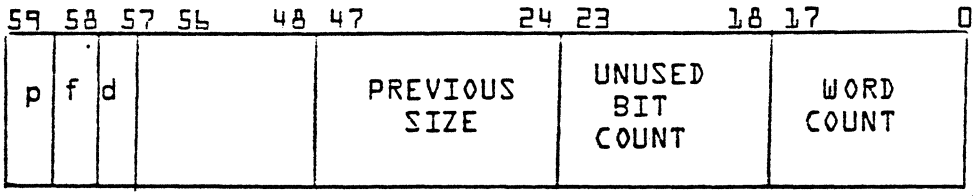
53-42 block ordinal (mod 2^{12}=4096)

41-18 record number of first record in this block

17-0 word number, relative to the start of the block, of the control word for the first record which begins in this block.

First word of I-type block.

WCW W - format record Control Word



Bits 59 p odd parity bit

58 f flag bit (end of record group)

57 d delete bit (set implies record is not physically there)

56-48 reserved

47-24 length of previous record including WCW

23-18 unused bit count of last 60 bit word in block

17-0 record length excluding the WCW

Delimits sections, and logical records in W-format record files.

The following table indicates the use of the d and f bits

d	f	Use
0	0	end of data record
0	1	generated by the ENDFILE macro
1	1	generated by WEOR macro and 7/8/9 card

APPENDIX E

The following table presents the PP driver names for the various 6000 tape formats.

7 track

FORMAT	READ	WRITE
SCOPE std.	1RT	1WI
X	1RT	1WX
S	1RS	1WS
L	1MT	1MT

9 track

Read - 1NR

Write - 1NW

Write parity error - 1PX

Read parity error is included in 1NR

BIBLIOGRAPHY

PUBLICATION/ PRODUCT NUMBER	TITLE
60189400	6000 SCOPE 32. Reference Manual
M628a	6000 SCOPE 3.2 Internal Maintenance Documentation
E013*3.3	6000 SCOPE 3.3 External Reference Specification
60305200	6000 SCOPE 3.3 Reference Manual
60281200A	7000 SCOPE 1.1 Reference Manual
60281600A	7611-1 I/O Station Operator Guide
E130*2	7000 SCOPE 2.0 External Reference Specification