60495700

CONTROL DATA CORPORATION

CYBER RECORD MANAGER BASIC ACCESS METHODS VERSION 1.5 REFERENCE MANUAL

CDC® OPERATING SYSTEMS:

NOS 1
NOS 2
NOS/BE 1

REVISION RECORD

Revision	Description
A (11-01-75)	Original release.
B (03/05/76)	This revision reflects 7000 Record Manager as released under SCOPE 2.1.4: new features include F053 connected file flag. The revision also reflects CYBER Record Manager Version 1.4: new features include DM 119 FILE control statement cancel; and DM 135 internal changes which do not affect this manual. See the list of effective pages.
C (07/01/77)	This revision reflects CYBER Record Manager 1.4 at PSR level 452. All references to 7000 Record Manager have been eliminated.
D (03/31/78)	This revision reflects feature CP 091, CYBER Record Manager Basic Access Methods Version 1.5.
E (10/01/79)	This revision reflects miscellaneous technical/editorial changes and corrections. Obsoletes all previous editions.
F (01/30/81)	This revision reflects miscellaneous technical/editorial changes and corrections at PSR level 528.
G (05/17/82)	This revision reflects CYBER Record Manager Basic Access Methods Version 1.5 at PSR level 564. It supports NOS Version 2.0 and includes miscellaneous technical corrections.

REVISION LETTERS I, O, Q, AND X ARE NOT USED

Address comments concerning this manual to:

CONTROL DATA CORPORATION
Publications and Graphics Division
P. O. Box 3492
SUNNYVALE, CALIFORNIA 94088-3492

or use Comment Sheet in the back of this manual

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

Page	Revision
Front Cover	-
Title Page	-
ii	G
iii/iv	G
v	G
vi	G
vii	G
viii	G
ix	F
1-1 thru 1-3	F
2-1 thru 2-7	F
2-8	G
2-9 thru 2-12	F
3-1	F
3-2 thru 3-10	G
4-1 thru 4-7	F
5-1 thru 5-3	F
5-4	G
5-5	F
5-6	F
5-7	G
5-8	G
6-1	F
6-2	F
6-3	G
6-4	G
6-5	F
6-6 thru 6-9	G
A-1	F
A-2	F
A-3	E
A-4	E
B-1 thru B-10	F
B-11	G
B-12	F
B-13	F
C-1	G
C-2	F
C-3	F
D-1 thru D-8	F
D-9	G
D-10	F
D-11 thru D-14	F
E-1	F
E-2	F
F-1	F
G-1	F
H-1	F
Index-1 thru -5	G
Comment Sheet/Mailer	G
Back Cover	-

PREFACE

CYBER Record Manager Basic Access Methods (BAM) Version 1.5 operates under control of the following operating systems:

- NOS 1 and NOS 2 for the CONTROL DATA® CYBER 180 Series; CYBER 170 Series; CYBER 70 Models 71, 72, 73, 74; and 6000 Series Computer Systems.
- NOS/BE 1 for the CDC® CYBER 180 Series; CYBER 170 Series; CYBER 70 Models 71, 72, 73, 74; and 6000 Series Computer Systems.

BAM input and output facilities are available to users of COMPASS assembly language through macro calls; user programs, COBOL, FORTRAN, and Sort/Merge use BAM for input/output operations. The user programs communicate with BAM either through the compiler, using the calls supplied within the languages, or with BAM macros.

Intended as a primary document for COMPASS programmers, this manual presents background information and operational specifications for BAM. COBOL, FORTRAN, and Sort/Merge programmers can use this manual as a source for BAM terminology and concepts; specific language interfaces are detailed in the appropriate reference manuals. The user is assumed to be familiar with the operating system at the installation, and with file organization and manipulation.

The NOS 1, NOS 2, and NOS/BE Manual Abstracts are pocket-sized manuals containing brief descriptions of the contents and intended audience of all manuals for NOS 1, NOS 2, and NOS/BE and their product sets. The manual abstracts can be useful in determining which manuals are of greatest interest to a particular reader.

The Software Publications Release History serves as a guide in determining which revision level of software documentation corresponds to the Programming Systems Report (PSR) level of installed site software.

Information necessary for a complete understanding of BAM use is contained in the publications listed below. The applicable operating systems are also indicated.

The following manuals are of primary interest:

Publication	Publication Number	NOS 1	NOS 2	NOS/BE 1
COMPASS Version 3 Reference Manual	60492600	X	X	X
CYBER Record Manager Basic Access Methods Version 1.5 User's Guide	60495800	X	X	X
NOS Version 1 Reference Manual Volume 1 of 2	60435400	X		
NOS Version 1 Reference Manual Volume 2 of 2	60445300	X		
NOS Version 2 Reference Set, Volume 3 System Commands	60459680		X	
NOS Version 2 Reference Set, Volume 4 Program Interface	60459690		X	
NOS/BE Version 1 Reference Manual	60493800			X

The following manuals are of secondary interest:

Publication	Publication Number	NOS 1	NOS 2	NOS/BE 1
Common Memory Manager Version 1 Reference Manual	60499200	X	X	X
CYBER Loader Reference Manual	60429800	X	X	
CYBER Record Manager Advanced Access Methods Version 2 Reference Manual	60499300	X	X	X
CYBER Record Manager Advanced Access Methods Version 2 User's Guide	60499400	X	X	X
NOS Version 1 Diagnostic Index	60455720	X		
NOS Version 2 Diagnostic Index	60459390		X	
NOS/BE Version 1 Diagnostic Index	60456490			X
NOS Version 1 Manual Abstracts	84000420	X		
NOS Version 2 Manual Abstracts	60485500		X	
NOS/BE Version 1 Manual Abstracts	84000470			X
NOS/BE Version 1 System Programmer's Reference Manual	60494100			X
Software Publications Release History	60481000	X	X	X

CDC manuals can be ordered from Control Data Corporation, Literature and Distribution Services, 308 North Dale Street, St. Paul, Minnesota 55103.

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or parameters.

CONTENTS

APPENDIXES

INDEX

FIGURES

TABLES

NOTATIONS

The following notations are used throughout the manual with consistent meaning:

UPPERCASE In language syntax, uppercase indicates a statement keyword or character that is to be written as shown.

lowercase In language syntax, lowercase indicates a name, number, or symbol that is to be supplied by the programmer.

[ ] In language syntax, brackets indicate an item that can be used or omitted.

{ } In language syntax, braces indicate that only one of the vertically stacked items can be used.

••• In language syntax, a horizontal ellipsis indicates that the preceding optional item in brackets can be repeated as necessary.

⋮ In program examples, a vertical ellipsis indicates that statements or parts of the program have not been shown.

Numbers that appear without a subscript are decimal values. Other value formats are denoted as:

n . . . n	Value is decimal
n . . . nB	Value is octal
n . . . nW	Value is decimal, specified in words

BAM FEATURES

The CYBER Record Manager Basic Access Methods (BAM) provide an interface between user programs and system input/output routines. BAM routines exist in NOS/BE and NOS operating systems.

BAM also provides:

- Consistent error processing
- Accommodation for various labeling conventions
- Maintenance of different file organizations

BAM routines are used by some compilers and are available for user programs. Use of BAM by compilers and user programs extends input/output compatibility to both the system and application program levels.

The primary task of BAM is record and block input/output for files on supported devices. Consequently, the various types of records, blocks, and file organizations must be identified for BAM. These and other file characteristics must be set by the user in a file information table (FIT). The FIT is divided into fields that describe certain aspects of the file. Refer to appendix D for the exact structure of the FIT.

CDC offers guidelines for the use of the software described in this manual. These guidelines appear in appendix H. Before using the software described in this manual, the reader is strongly urged to review the content of this appendix. The guidelines recommend use of this software in a manner that reduces the effort required to migrate application programs to future hardware or software systems.

REFERENCES

The following terms are relevant to BAM and related systems:

- Advanced Access Methods (AAM)

  A file manager that processes indexed sequential, direct access, and actual key file organizations and supports the Multiple-Index Processor.

- Basic Access Methods (BAM)

  A file manager that processes sequential and word addressable file organizations.

- CYBER Record Manager (CRM)

  A generic term relating to both BAM and AAM as they run under NOS/BE and NOS operating systems.

- Multiple-Index Processor (MIP)

  A processor that allows AAM files to be accessed by alternate keys.

FILE ORGANIZATIONS

Two file organizations are supported by BAM:

- Sequential (SQ)

  A collection of records stored in the same physical order in which they were generated.

- Word addressable (WA)

  A group of contiguous computer words that comprise a file. Records are accessed by a word number within the file.

MACROS

A FIT is established for each file by a FILE macro encountered at assembly time. This macro can contain the file name only, or it can have user-specified parameters describing a particular file. The FILE macro establishes the FIT in the using program's field length at the point at which it is called. FIT fields are assumed through default values when they are not specified as parameters in macros. The macros and functions are listed in table 1-1 according to their associated purposes:

- File creation and maintenance
- File initialization and termination
- Data transfer
- File updating
- File positioning
- Boundary conditions
- User label processing

TABLE 1-1. CYBER RECORD MANAGER MACROS

Function	Macro	Action Taken
File creation and maintenance	FILE	Creates a file information table (FIT). In addition to this macro, a FILE control statement is available to supply FIT information.
	FETCH	Retrieves the value of specified fields in the FIT.
	STORE	Sets values in fields of the FIT.
	SETFIT	Sets values in fields of the FIT with values supplied through the FILE control statement.
	FITDMP	Dumps the contents of a FIT to the error file.
File initialization and termination	OPENM	Prepares a file for processing; initiates label processing.
	CLOSEM	Terminates file or volume processing; initiates label processing.
Data transfer	GET	Transfers data from a file to the working storage area.
	PUT	Transfers data from the working storage area to a file.
	CHECK	Determines completion status of input/output operations.
File updating	REPLACE	Replaces a record in a file.
File positioning	SKIP	Repositions a file backward or forward.
	REWINDM	Rewinds the current volume to beginning-of-information (BOI).
Boundary conditions	ENDFILE	Records a partition terminator.
	WEOR	Records a section terminator.
	WTMK	Records a tapemark on a tape file.
User label processing	GETL	Retrieves the next label of a label string and delivers it to the label area.
	PUTL	Writes or checks a label in the label area.
	CLOSEL	Terminates label processing.

The applicability of some macros depends on the file organization established by the user. Table 1-2 presents macros as applicable to sequential and word addressable file organizations, the two supported by BAM.

This manual discusses macro properties and generalizes processing whenever possible. Explanations are provided in section 4 for each macro according to file organization. Consequently, material is presented redundantly for the benefit of a programmer who uses this manual to reference particular features only.

Macro statements are coded in COMPASS format. Each statement can contain a location field, a macro name in the operation field, a variable field, and a comment field. Any field is terminated by one or more blanks. A macro statement begins at character position 1 of an 80-column card image and continues through column 72. Columns 73 through 80 are used for sequencing. COMPASS coding conventions are shown in figure 1-1.

Suggested column conventions are as follows:

1	Comma (continuation), asterisk (comments line), or other (beginning of new statements)
2 thru 9	Location field entry, left-justified
10	Blank
11 thru 16	Operation field entry, left-justified
17	Blank
18 thru 29	Variable field entry, left-justified
30	Beginning of comments

TABLE 1-2. MACROS AND RELATED FILE ORGANIZATIONS

Macro	File Organization	
	SQ	WA
CHECK	X	X
CHECKR	X	X
CLOSEL	X	
CLOSEM	X	X
ENDFILE	X	
FETCH	X	X
FILE	X	X
FITDMP	X	X
GET	X	X
GETL	X	
GETP	X	
GETWR	X	
OPENM	X	X
PUT	X	X
PUTL	X	
PUTP	X	
PUTWR	X	
REPLACE	X	
REWINDM	X	X
SETFIT	X	X
SKIP	X	
STORE	X	X
WEOR	X	
WTMK	X	

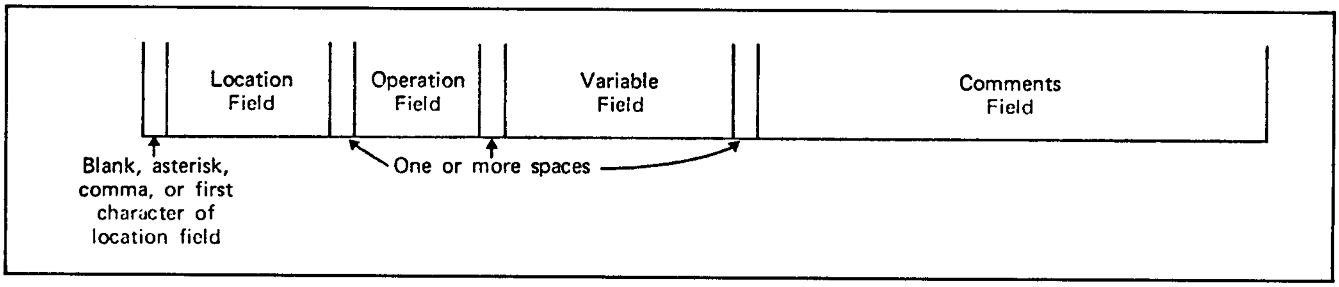


Figure 1-1. COMPASS Format

FILE STRUCTURES 2

Data structures progress hierarchically from the character level to the largest grouping of data, the file. The BAM user can describe file structure by file organization (FO), block type (BT), and record type (RT). This section presents these structures. Additionally, many of the file information table fields that must be set by the user are identified. They are explained in detail in section 3.

LOGICAL STRUCTURE

The logical structure of a file is user-controlled. The following definitions describe terms used throughout this manual that are applicable to the logical structure of a file:

- Record

  A record is a group of related characters. A character is represented in six bits as internal display code. A record or portion thereof is the smallest collection of information passed between BAM and the user. The user defines the structure and characteristics of records within a file by declaring a record format. The beginning and ending points of a record are implicit within each format. Records are grouped into files.

- Section

  A section consists of one or more records. Generally, a section is less than a partition and greater than a record, but it can be identical to either or both. A section begins with the first record after the end of the preceding section; a section ends when a special record or condition occurs. Only sequential files are grouped into sections.

- Partition

  A partition consists of one or more sections. Generally, it is less than a file and greater than a section, but it can be identical to either or both. A partition begins with the first record after the end of the preceding partition; a partition ends when a special record or condition occurs. Only sequential files are grouped into partitions.

- Block

  A block can contain partial records or one or more records. Block structure is interwoven with the physical recording format; unlike other logical file structure declarations, the block structure is transparent in use. Blocks are constructed from the records supplied by the user, and the user is supplied with records as required. The user is unaware of block boundaries. Only sequential files are grouped into blocks.

- File

  A file is a logically connected set of information; it is the largest collection of information that can be addressed by that file name. All data in a file is stored between the beginning-of-information (BOI) and the end-of-information (EOI). Label groups are not considered to be part of file data in the general case.

PHYSICAL STRUCTURE

The following definitions pertain to the physical means used to record files:

- Input/output device

  Any storage medium supported by the operating system.

- Rotation mass storage (RMS)

  Disk or disk pack.

- Mass storage device

  Disk, disk pack, or extended memory.

- Volume

  A volume is a reel of magnetic tape with sequential files. A file can be contained on more than one volume and a volume can contain more than one file.

- Level number

  A level number can range from 00 to 17_8 and is physically recorded on a physical record unit (PRU) device in an eight-character appendage to a short PRU. A short PRU consisting only of the eight-character level number appendage is called a zero-length PRU. The appendage is neither created by nor returned to the user. The level number value is available in the FIT on some input operations and can be specified by the user on some output operations.

- Physical record

  A physical record is defined only on magnetic tape; it consists of the data between interrecord gaps. A physical record need not contain a fixed amount of data.

- S/L tape

  S/L tape must be declared by the user. The physical structure of a file on an S/L tape depends entirely on the logical structures selected by the user; no operating system structure is superimposed. Physical record size is limited only by the buffer size on an L tape; physical record size on an S tape cannot be greater than 5120 characters. Noise size is an installation default on NOS/BE; noise size is an option on REQUEST and LABEL control statements on NOS. On S/L tapes, a block and a physical record are the same.

- PRU device

  All mass storage devices and non-S/L tapes are PRU devices; a physical structure is superimposed over the user-declared file structure by the operating system on all files that reside on PRU devices.

- Physical record unit (PRU)

  A PRU is the smallest unit of information that can be transferred between a peripheral storage device and central memory. The PRU size is permanently fixed for PRU devices; the PRU concept does not apply to S/L tapes. PRU device sizes are:

  Mass storage devices - 640 characters

  Binary SI tapes - 5120 characters

  Coded SI tapes - 1280 characters (supported under NOS/BE only)

  I tapes - 5120 characters (supported under NOS only)

- Short PRU

  A short PRU contains less than the number of characters defined for a PRU on a PRU device. An eight-character level number appendage is always part of a short PRU.

- System-logical-record

  A system-logical-record is defined only on PRU devices. It consists of a group of PRUs terminated by a short or zero-length PRU. A system-logical-record can be simulated on an S/L tape by writing a series of physical records of the same length as a PRU, followed by a physical record of a length less than a PRU and with a level number appendage. However, because of the installation parameter that defines noise (IP.NOISE=), no PRU smaller than the installation definition or operating system default can be written on an S/L tape. (The default on NOS/BE is 8 characters; the default on NOS is 18 characters.)

BAM controls the physical file position while the user controls only the logical file position. Physical and logical positions are not guaranteed to agree after a given operation unless S type records are being used.

FILE ORGANIZATIONS

BAM supports two file organizations: sequential and word addressable. Once the file organization is set for a BAM file, it must not be changed to an AAM file organization in the same job step. It is possible that the AAM interface routines are not loaded and that internal FIT fields have been initialized based on the BAM file organization. The following is a description of the structure of each organization and its applicable record and block types.

NOTE

> Refer to appendix H for recommendations on the selection of file organization.

SEQUENTIAL FILES

Sequential files are tape-like in structure. Records are placed in the order of presentation; physically, a record follows the previous record. Given the location of one record, the location of the next record is determined in relation to the given record only. A sequential file can extend across any number of volumes and can be accessed sequentially only.

A sequential file can reside either on a magnetic tape or on mass storage. Tape files, punch card or printer files, and some mass storage files are classified as sequential. A mass storage sequential file is not necessarily maintained internally in sequential order by CIO; however, records are presented to the user in sequential order. All sequential files are blocked through the block type parameter specification, regardless of device type, except for S type records.

The logical structure of a sequential file is shown in figure 2-1. The physical structure of a sequential file is shown under the discussion of the various block types.

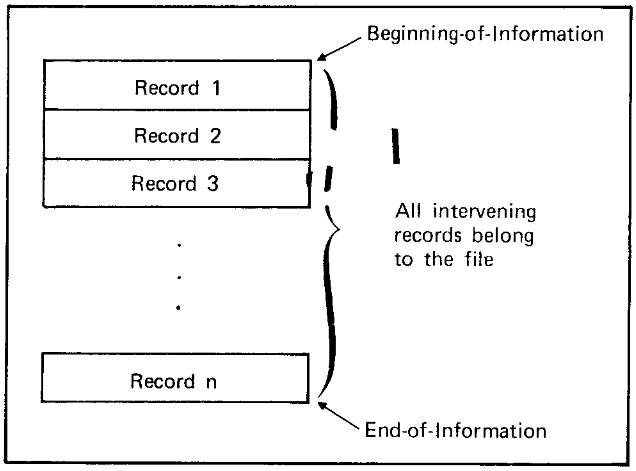


Figure 2-1. Logical Structure of a Sequential File

Block Types for Sequential Files

Sequential file blocking is, essentially, the concept of compressing actual records into contiguous record groups, thereby saving storage that would otherwise be wasted for interrecord gaps. Blocks can be various types, as explained in the following discussion. BAM supports four block types identified as I, C, K, or E. These block types are applicable to sequential files. A summary of block types and physical recording formats is represented in table 2-1.

NOTE

> Refer to appendix H for recommendations on the selection of block type.

Internal Blocking Type I

I type blocks begin with a block control word, which contains block and record identification. Contents of the block control word include a pointer to the first record beginning in the block. I type blocks can contain only W type records. I type block size is 5120 characters; however, the last block of the section, partition, or file can be shorter.

A file with I type blocks can be recorded on either a PRU device or an S/L tape. On a PRU device, a short I block is recorded as a short PRU, which is the end of a system-logical-record. On an S/L tape, I type blocks are not an allowable American National Standards Institute (ANSI) interchange format because ANSI does not define W type records.

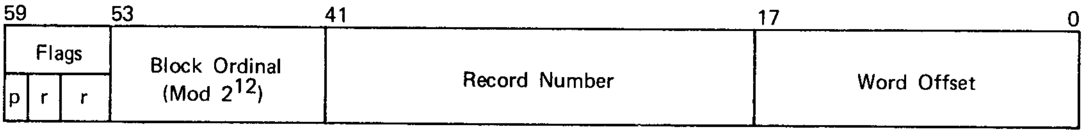
The block control word format is shown in figure 2-2. Blocks and records are numbered consecutively from 1. The record number includes all records that are physically present whether they are logically present or not. If no record begins in the block, word offset and record number equal zero. The block control word is word zero of the block.

Character Count Block Type C

Each C type block contains the number of characters specified by the value of the maximum block length (MBL) field of the FIT; however, the last block of the section, partition, or file can be shorter. Except for S type records, records can span block boundaries as shown in figure 2-3; for this reason it is important to specify noise size for S/L tape less than 10 characters, perferably NS=8. C type blocks can contain any record type.

TABLE 2-1. BLOCK TYPE USAGE

Block Type	Physical Recording Format	
	PRU Device	S/L Tape
I	I block size is 5120 characters; section or partition is a single system-logical-record.	I block size is 5120 characters; last block in section, partition, or file can be shorter.
C	C block size is equal to 0 (unblocked) or a multiple of PRU size; section or partition is a single system-logical-record.	C block size equals 5120 characters for S tapes and a maximum of the value of the BFS field in characters minus 20 for L tapes; last block of section, partition, or file can be shorter.
K	K blocking on PRU devices is prohibited.	Each K block is written as a physical record.
E	E blocking on PRU devices is prohibited.	Each E block is written as a physical record.



59 Parity bit, used to maintain odd parity within the control word.

58, 57 Reserved for CDC.

56 thru 54 Reserved for users.

53 thru 42 Ordinal of the current block (modulus 4096).

41 thru 18 Ordinal of the first record beginning in this block (modulus 2^{24}, if necessary).

17 thru 0 Word number of the control word of the first W type record in the block.

Figure 2-2. Block Control Word Format for I Type Blocks

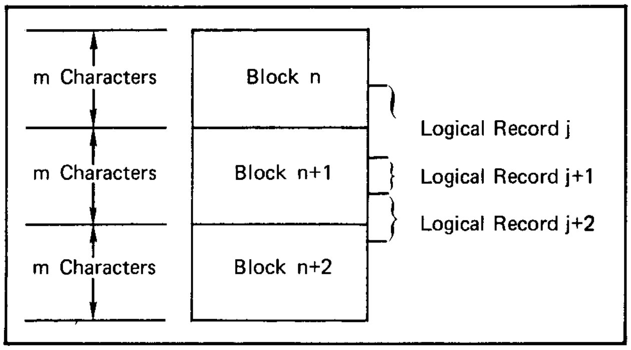


Figure 2-3. C Type Block Structure

If the MBL field is not specified, the default values are set as follows:

S tapes MBL=5120 characters

L tapes MBL=value of the buffer size (BFS) field in characters minus 20

PRU devices MBL=0 (unblocked)

If a value is specified for the MBL field, it can be a maximum of 5120 characters for S tapes and a maximum of the value of the BFS field in characters minus 20 for L tapes. The most efficient value of the MBL field for PRU devices is 0; however, it can be set to a multiple of PRU size. If the value specified is not 0 or a multiple of PRU size, the value is rounded down to a multiple. The MBL field set to PRU size facilitates parity error recovery for W type records because a boundary condition would exist, and a control word is written at all block boundaries.

When record type is S and block type is C, any user value for MBL is not changed for files on any device. S type records cannot be blocked. On an S/L tape, one S type record is one tape block. The C type block on an S/L tape is not an allowable ANSI interchange format because BAM does not support the ANSI spanned record type.

Record Count Block Type K

For K type blocks, each variable-length block contains the same number of records. Records cannot span blocks. Noise size for S/L tapes should be less than minimum record length (MNR). The last block of a partition or file can contain fewer than the value specified in the number of records per block (RB) field of the FIT. K type blocks are prohibited on PRU devices; they are valid for S/L tapes only. K type blocks can contain any type record except S or W type records.

Padding can be inserted when a K type block is written. The three FIT fields concerned with padding are the padding character (PC), the multiple of characters per block (MUL), and the minimum block length (MNB). The value of the MNB field takes precedence over the value of the MUL field. Padding is inserted so that each block, except possibly the last one on a file or volume, is a multiple of MUL characters and is at least the number of characters specified by the value of the MNB field in length. The last block of a partition can contain fewer than the number of characters specified by the value of the MNB field; padding is not added to the last block because the GET macro cannot distinguish padding from a valid record.

When writing K type blocks, the value of the RB field of the FIT is used to construct blocks of exactly that number of records. When reading K type blocks, each block need not be exactly the number of records specified by the value of the RB field because blocks are physically delimited and boundaries are readily detected. However, if the RB field of the FIT is set to a value less than the number of records physically present, only the number of records specified by the value of the RB field are returned to the working storage area; other records physically present are assumed to be padding and are not returned to the working storage area.

K type blocks are recorded as tape physical records. To ensure that the last block in a file is interpreted correctly, minimum record size should be greater than noise record size because it is possible for the last block to contain only a single record.

The K type block is an allowable format for ANSI standard tape interchange. The structure of a K type block is shown in figure 2-4.

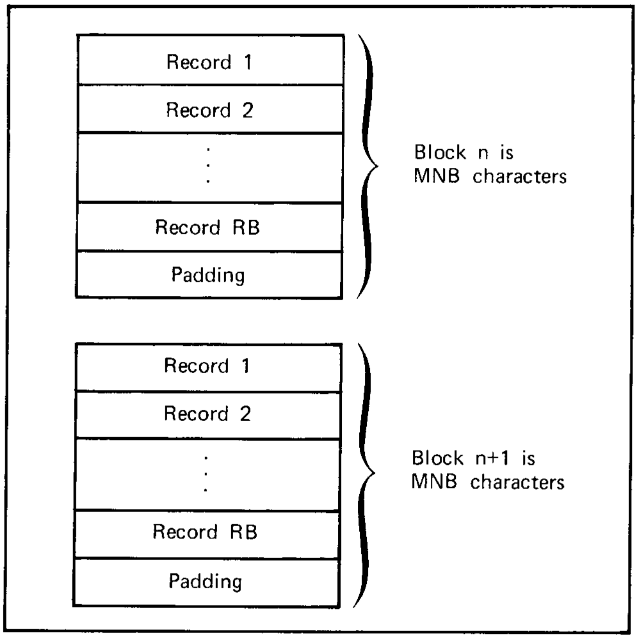


Figure 2-4. K Type Block Structure

Exact Records Block Type E

Each E type block contains as many whole records as can be contained in the block size, which is the number of characters specified by the value of the maximum block length (MBL) field of the FIT. E type blocks are prohibited on PRU devices; they are valid only for S/L tapes. Any type record, except S or W type records, can be contained in E type blocks. Noise size for S/L tapes should be less than minimum record length (MNR).

Padding can be inserted when an E type block is written. The three FIT fields concerned with padding are the padding character (PC), the multiple of characters per block (MUL), and the minimum block length (MNB). The value of the MNB field takes precedence over the value of the MUL field. Padding is inserted so that each block, except possibly the last one on a file or volume, is a multiple of MUL characters and is at least the number of characters specified by the value of the MNB field in length. The last block of a partition can contain fewer than the number of characters specified by the value of the MNB field. To ensure that the last block in a file is interpreted correctly, minimum record length should be greater than noise record size because it is possible for the last block to contain only a single record.

When specifying E type blocks with padding, the following restriction must be observed or E type blocks can be constructed in which padding cannot be distinguished from data. The value of the MBL field minus the value of the MNB field must be greater than the value of the MRL field minus the value of the MNR field; and the value of the MUL field must be less than the value of the MNR field.

The E type block is an allowable format for ANSI standard tape interchange. E type block structure is shown in figure 2-5.

File Boundaries

The beginning-of-information is that point in a file before which no data exists. The end-of-information is that point in a file after which no data exists. Table 2-2 shows the various file boundary conditions for sequential files.

Partition Boundaries

A partition begins at beginning-of-information or after a preceding end-of-partition (EOP). A partition ends at end-of-information (EOI) or on the occurrence of an end-of-partition boundary. End-of-partition boundaries vary depending on device, block type, and record type, as shown in table 2-3.

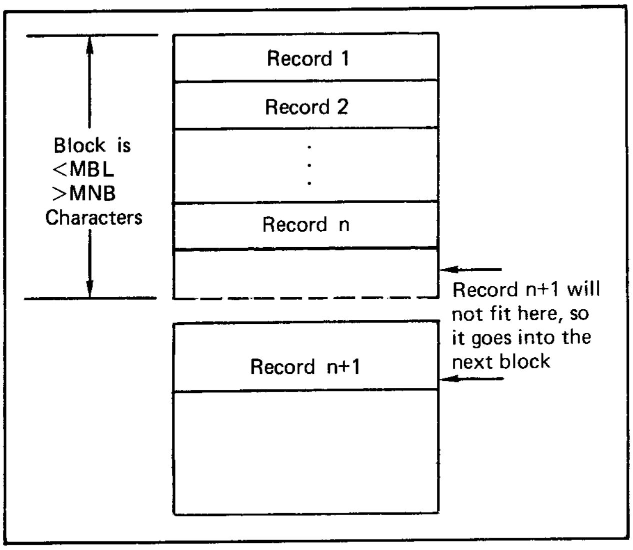

Record 1
Record 2
Block is
<MBL
>MNB
Characters
Record n
Record n+1 will not fit here, so it goes into the next block
Record n+1

Figure 2-5. E Type Block Structure

Section Boundaries

A section begins at beginning-of-information, or after a preceding end-of-partition, or after a preceding end-of-section (EOS). A section ends at end-of-information, or end-of-partition, or at the occurrence of an end-of-section boundary. End-of-section boundaries vary depending on device, block type, and record type, as shown in table 2-4.

S type records are a special case for section identification. Although an S type record is defined as a record terminated by a short PRU of level less than 17_8, an S type record is never considered to be a section. When S type records are read, the file position (FP) field of the FIT is set to end-of-record, never to end-of-section.

Volume Boundaries

Volume boundaries are defined only for sequential files on magnetic tape. The user of such files can elect to ignore volume boundaries or to be notified

TABLE 2-2. SEQUENTIAL FILE BOUNDARY CONDITIONS

Device	Boundary	
	Beginning-of-Information	End-of-Information
Mass storage	Before the first record written.	After the last record written.
Labeled tape	Between the file header label group and the first record written.	Between the last record written and the file trailer label group.
Unlabeled S/L tape	Between load point and the first record written.	Undefined.
Unlabeled SI tape	Between load point and the first record written.	Between the last record written and the file trailer label group.
Unlabeled I tape	Between load point and the first record written.	Between the last record written and the file trailer label group.

when volume boundaries occur. A volume boundary has no necessary relationship to any logical boundary and can occur at any point within a file. The beginning-of-volume of the first volume is synonymous with beginning-of-information. Thereafter, beginning-of-volume is located before the first data block on second and subsequent volumes. An end-of-volume condition exists when one of the conditions shown in table 2-5 occurs.

WORD ADDRESSABLE FILES

Word addressable files are mass storage files containing continuous data or space for data. Words within the file are numbered from 1 to n, each word containing 10 characters. Data is read or written within the file, starting at a word specified by the word number, called the word address.

TABLE 2-3. END-OF-PARTITION BOUNDARIES

Device	Block Type	Record Type	End-of-Partition Boundary
PRU Device	I	W	One-word deleted record pointing back to the last I block boundary; control word with the EOP flag; terminate the system-logical-record with level 0.
	C	W	Control word with an EOP flag; terminate the system-logical-record with level 0.
	C	All but W	Terminate the system-logical-record with level 0; zero-length PRU with level 17_8.
S/L Tape	I	W	Zero-length deleted records to exceed noise record size; one-word deleted record pointing back to the I block boundary; control word with an EOP flag; terminate the block.
	C	W	Zero-length deleted records to exceed noise record size; control word with an EOP flag; terminate the block.
	C,K,E	All but W	Terminate the block; tapemark.

TABLE 2-4. END-OF-SECTION BOUNDARIES

Device	Block Type	Record Type	End-of-Section Boundary
PRU Device	I	W	One-word deleted record pointing back to the last I block boundary; control word with EOS flags; terminate the system-logical-record with level 0.
	C	W	Control word with EOS flags; terminate the system-logical-record with level 0.
	C	All but W	Terminate the system-logical-record with level less than 17_8.
S/L Tape	I	W	Zero-length deleted records to exceed noise record size; one-word deleted record pointing back to the I block boundary; control word with EOS flags; terminate the block.
	C	W	Zero-length deleted records to exceed noise record size; control word with EOS flags; terminate the block.
	C,K,E	All but W	Terminate the block (undefined on a read).

TABLE 2-5. END-OF-VOLUME BOUNDARIES

Device	End-of-Volume Boundary
Labeled tape	Between the last record on tape and the volume trailer label group.
Unlabeled tape	Between the last record on tape and the volume trailer label group (PRU device only) or the first tapemark after the reflective spot (S/L tapes).
Nonstandard labeled tape	Between the last record on tape and the nonstandard end-of-volume label which is controlled by the user.

Reading beyond the current end-of-information limit is not allowed. For writing, word addressable files are automatically extended if the write results in an address beyond the end-of-information. Word addressable files can be accessed either sequentially or randomly by word address. The user should recognize that a sequential read is valid only if data is contiguous. The supplied word address for random access points to a location in the file that is on a word boundary; therefore, all records begin on a word boundary.

Although word addressable files must reside on mass storage for processing, the COPYBR or COPYBF utility can be used to copy a word addressable file to tape. The COPYBR utility is preferable. Any level 17_8 information written by the copy is ignored when the file is restored to mass storage and the user is writing data on the file. A read of level 17_8 written by the copy utility returns an end-of-partition status.

Only W, F, and U type records are possible in word addressable files. The logical structure of a word addressable file is shown in figure 2-6.

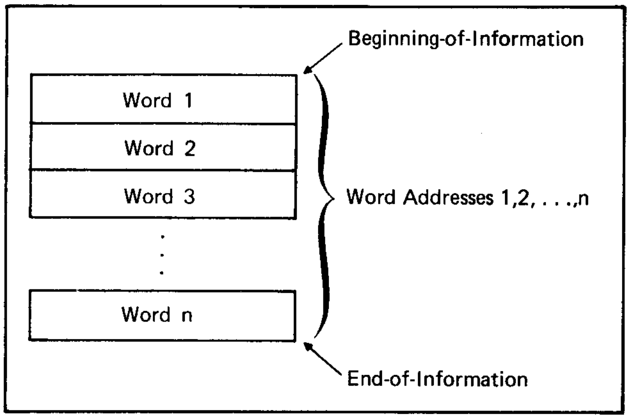


Figure 2-6. Logical Structure of a Word Addressable File

RECORD TYPES

BAM supports eight record types. The eight record types and a corresponding explanation of their lengths are listed in table 2-6.

The numbering conventions for describing a record or the position of a control field in a record are summarized in figure 2-7. All record lengths are specified by character count. Values normally are unsigned positive decimal integers.

The record types allowed for each block type are shown in table 2-7. An X indicates applicable record types. The use of Z type records in K and E blocks and R type records in all block types is discouraged. S type records are not blocked, but block type can be set to C for compatibility with SCOPE 2 type files.

NOTE

Refer to appendix H for recommendations on the selection of record type.

DECIMAL CHARACTER COUNT TYPE D

The record length for D type records is specified in a length field located within the record. The two fields of the FIT that specify the position of the length field are the length field beginning character position (LP) field, numbering from 0, and the length field length (LL) field, which is the number of characters in the length field, one to six characters. The FIT fields that must be specified for D type records are:

LP Length field beginning character position (counting from 0)

LL Length field length (one to six characters)

MRL Maximum record length

TABLE 2-6. RECORD TYPES AND LENGTH DESCRIPTIONS

Record Type	Length Description
Decimal Character Count (D)	Length is given as character count, by the length field contained within the record.
Fixed Length (F)	Fixed length.
Record Mark (R)	Terminated by a record mark character specified by the user.
System Record (S)	Length of the system-logical-record depends on the PRU device; on an S/L tape, one S type record is a physical record.
Trailer Count (T)	Fixed length header followed by a variable number of fixed length trailers; the header contains the trailer count field.
Undefined (U)	Length is defined by the user.
Control Word (W)	Length is contained in a control word prefixed to a record by BAM.
Zero Byte (Z)	Terminated by a 12-bit zero byte in the low-order byte position of a 60-bit word.

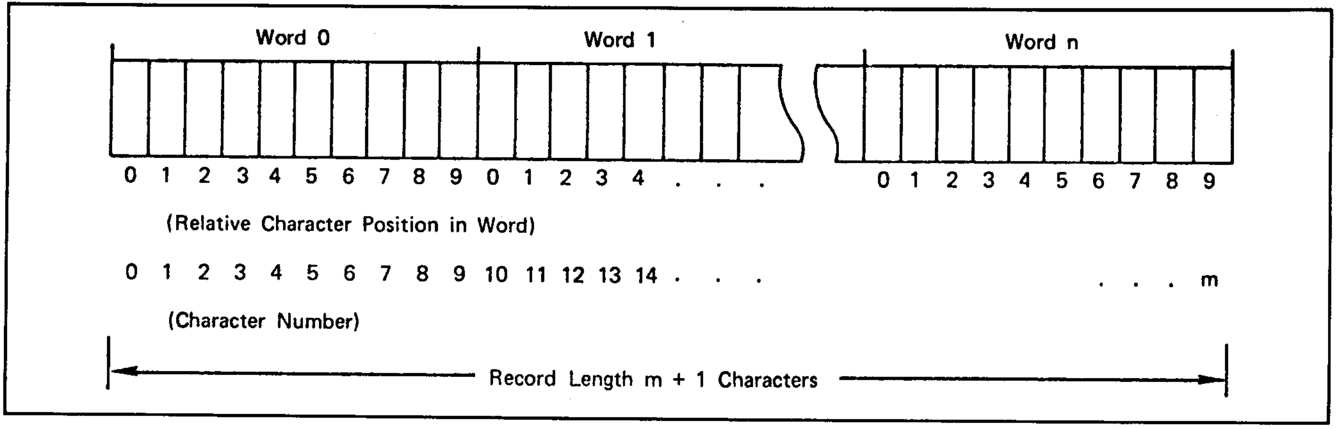


Figure 2-7. Numbering Conventions

TABLE 2-7. RECORD TYPE AND BLOCK TYPE ASSOCIATIONS

Block Type	Record Type							
	F	D	R	T	U	W	Z	S
I						X		
C	X	X	X	X	X	X	X	X
K	X	X	X	X	X		X	
E	X	X	X	X	X		X	

The record length specified must be less than or equal to the number of characters specified by the value of MRL. Maximum record length that can be specified is 1310710 characters. Minimum record length (MNR) must be large enough to contain the length field and should be at least 10 characters to ensure a correct detection of end-of-data conditions.

If MNR or the sum of LL and LP (if MNR = 0) is less than or equal to nine characters, padding from the last word of a block can be read as valid data for a PRU device.

The record length specified in the length field is right-justified display code filled with zeros or blanks. The length field located within the record is interpreted as a binary number if the C1 field of the FIT is set to YES, or as a sign-overpunch if the sign-overpunch (SB) field of the FIT is set to YES.

For the first GETP or PUTP macro issued for a given record, the minimum number of characters that can be transferred is the value specified in the minimum record length (MNR) field of the FIT. If the user does not supply a value for the MNR field, the sum of the values of the LL field and the LP field is used.

In the example in figure 2-8, the length field is three characters beginning with character position 22. The minimum number of characters that can be transferred for a partial read or write is 25.

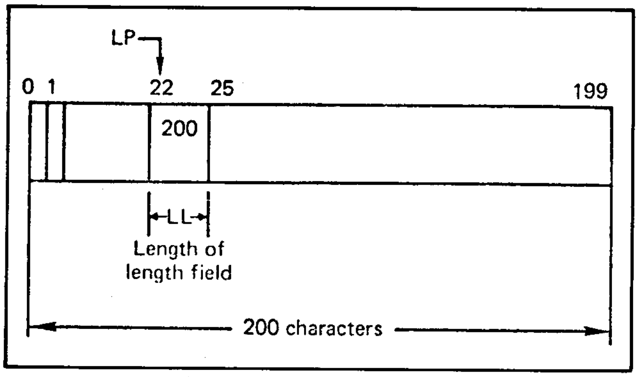


Figure 2-8. D Type Record Example

FIXED LENGTH TYPE F

Fixed length records are defined as records that are the number of characters in length specified by the value of the fixed length (FL) field of the FIT. All records in the file are of equal size. Maximum record length is 1310710 characters; minimum length is 10.

The FIT field that must be specified for F type records is:

FL Fixed length

Any value in the record length (RL) field of the FIT is ignored, and the number of characters specified by the value of the FL field of the FIT are moved when a GET or PUT macro is issued. A value must be supplied for the FL field for the file to be successfully opened. No padding is supplied on a read.

RECORD MARK TYPE R

The size of an R type record is specified indirectly by a special delimiting character that terminates each record. The user specifies the delimiting character in the record mark character (RMK) field of the FIT. The same delimiting character is used for each record in the file. This character can be any character of the character set. The record length must be less than or equal to the number of characters specified by the maximum record length (MRL) field of the FIT. Maximum record length is 131071 characters. Minimum record length should be at least 10 characters to ensure a correct detection of end-of-data conditions.

The FIT fields that must be specified for R type records are:

MRL	Maximum record length
RMK	Record mark character (default is right bracket,])

For a file read, if the delimiting character is not found in the first number of characters specified by the value of the MRL field, that number of characters are moved to the working storage area and an excess data error is given. For writing, if the delimiting character is not found in the first number of characters specified by the value of the MRL field, no data is written to the file and an excess data error is given.

In the example in figure 2-9, MRL=120 and RMK=62_8. The characters are read up to the record mark character.

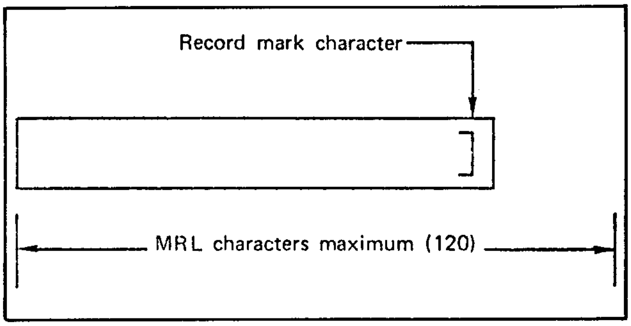


Figure 2-9. R Type Record Example

SYSTEM RECORD TYPE S

On PRU devices, each record is a system-logical-record occupying an integral number of central memory words. On S/L tapes, each record is a tape physical record. The differences in processing of S type records for S/L tapes and PRU devices are shown in table 2-8.

S type records are word-oriented. When physical blocks are being read from S/L tapes, however, the record length (RL) field of the FIT represents the actual number of characters in the block. For all other cases, the value of the RL field represents the record length rounded upward to a multiple of 10.

The FIT fields that must be specified for S type records are:

RL	Record length (write only)
MRL	Maximum record length (read only)

An S type record can be created by executing one PUT macro, a series of PUTP macros with a terminating WEOR macro, or a PUTP macro with a TERM parameter. When the WEOR macro is used after a PUTP, level numbers 0 through 16_8 can be written on NOS/BE to terminate the record. Use of levels other than 0 is discouraged, however. If a series of PUTP macros is followed by a PUT, the record written through the PUTP macro is terminated and a new record to satisfy the PUT macro is begun. A user-specified value for the RL field causes the current record to be terminated when the number of characters specified have been written.

S type records can be specified for a magnetic tape file with unknown format so that the record length will be returned to the RL field after each read. An S or L tape must be specified on a LABEL or REQUEST statement unless the format is known to be SI or I.

TRAILER COUNT TYPE T

T type records consist of a fixed-length header and a variable number of fixed-length trailer items. A count field in the fixed-length header specifies the number of fixed-length trailer items appended to each record. The value recorded in the count field can be display code, right-justified, and zero or blank filled.

The FIT fields that must be specified for T type records are:

HL	Header length
TL	Trailer length
CP	Trailer count beginning character position (counting from 0)
CL	Count field length (one to six characters)
MRL	Maximum record length

The value of CL located within the record is interpreted as a binary number of the C1 field of the FIT is set to YES, or as a sign-overpunch if the sign-over punch (SB) field of the FIT is set to YES.

The value of the CP field plus the value of the CL field must be less than or equal to the value of the HL field. The value of the HL field must be less than or equal to the value of the MRL field. Maximum record length that can be specified is 1310710 characters. Minimum record length (MNR) must be large enough to contain the count field and should be at least 10 characters to ensure a correct detection of end-of-data conditions. If HL is less than or equal to nine characters, padding from the last word of a block can be read as valid data for a PRU device. The logical structure of a T type record is shown in figure 2-10.

TABLE 2-8. PROCESSING FOR S TYPE RECORDS

Specification	PRU Device	S/L Tape
Block type	Block type is ignored.	Block type is ignored. Every logical record is one physical record.
Maximum block length	MBL is forced to multiples of PRU size of the device.	If the user specifies an MBL, the value must be greater than or equal to MRl and less than or equal to 5120 characters for S tapes and BFS in characters minus 20 for L tapes; otherwise, MBL defaults to the S tape block size. An error results if an MBL greater than block size is specified.
Record length	RL is rounded up to an integral number of central memory words.	RL specifies the number of characters read or written.
PUT	One system-logical-record of length RL is written, terminated by a level 0.	One physical record of length RL is written (no level number).
PUTP	PTL characters are moved into the buffer (maximum of RL if specified).	PTL characters are moved into the buffer (maximum of RL if specified).
WEOR	Terminate the current record and system-logical-record and, on NOS/BE, write a level 0 through 16_8.	Terminate the current physical record.
Maximum record length	MRL=0 allows any length record; if MRL≠0 and the record exceeds MRL, an error is given.	If the user specifies MRL, the value must be less than or equal to MBL; otherwise, MRL defaults to the S tape block size. MRL is reduced to S tape block size if an MRL larger than S tape block size is specified.
GET	MRL must be large enough to contain the entire system-logical-record. If the record exceeds MRL, excess data error is given.	MRL must be large enough to contain the physical record. If the record exceeds MRL, an excess data error is given.
GETP	PTL characters, or the number of characters remaining in the record, are moved from the buffer to WSA.	PTL characters, or the number of characters remaining in the record, are moved from the buffer to WSA.
ENDFILE	Terminate current system-logical-record and write level 0. Write a zero-length PRU with level 17_8.	Terminate the current physical record. Write a tapemark.

UNDEFINED TYPE U

Specifying U type records permits processing of any record type not provided by BAM.

The FIT fields that must be specified for U type records are:

RL	Record length (read and write)
MRL	Maximum record length (read only)

The user must supply a value for the RL field of the FIT for each GET and PUT. The value of the RL field must be less than or equal to the value specified by the MRL field. Maximum record length that can be specified is 1310710 characters.

The RL field of the FIT is altered at the completion of a GET only if an end-of-data has been detected before the number of characters specified by the user in the RL field have been read. The value of the RL field indicates the number of characters transferred.

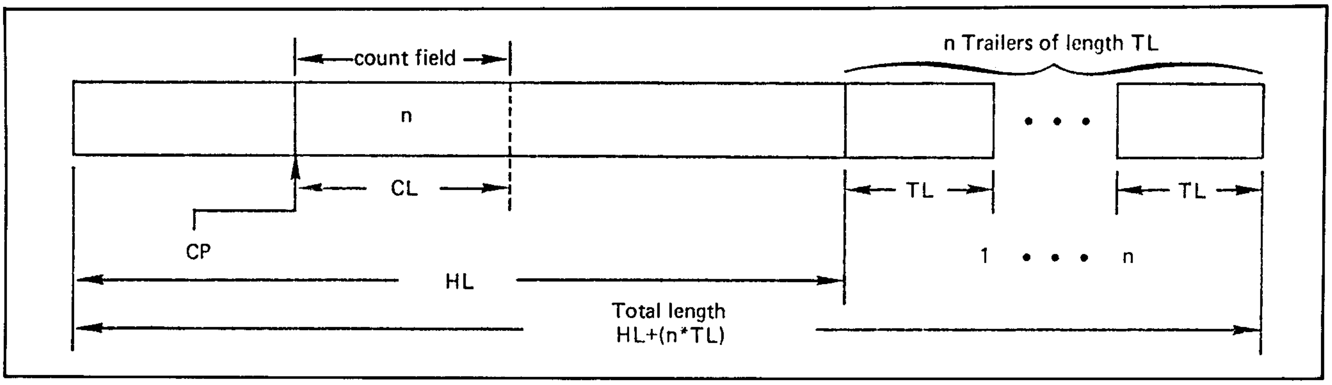


Figure 2-10. T Type Record Format

S type records can be specified for a magnetic tape file with unknown format so that the record length will be returned to the RL field after each read. An S or L tape must be specified on a LABEL or REQUEST statement unless the format is known to be SI or I.

CONTROL WORD TYPE W

A W type record is any length less than or equal to the number of characters specified by the value of the maximum record length (MRL) field of the FIT beginning at a word boundary (bit 59 of the word). A record is represented in the file as an integral number of central memory words, prefixed with a record control word supplied by BAM. The format for the record control word is shown in figure 2-11. The control word is also written at all block boundaries.

The FIT fields that must be specified for W type records are:

RL	Record length (write only)
MRL	Maximum record length (read only for WA files, read and write for SQ files)
CM	Conversion mode set to NO for SQ files

The RL field of the FIT (or the PTL field if a record is written in pieces) must be specified for writing; when reading, the value of the RL field is determined by the control word. Only the characters specified by the value of the RL field are returned to the working storage area on a read. The contents of any unused bits in the last word returned are undefined.

To ensure that a tape file with W type records can always be closed, the length of a noise record should be less than 10 characters. W type records cannot be used with E or K type blocks.

ZERO BYTE TYPE Z

A Z type record is terminated by a 12-bit byte of zeros in the low-order position of the last word in the record. The full length (FL) field of the FIT must be specified for Z type records. The value of the FL field can be between 1 and 1310710.

The FIT field that must be specified for Z type records is:

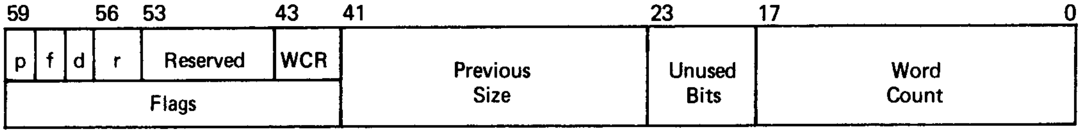
FL	Full length

When a record is read, the zero byte is stripped from the record and blank padding is added to fill the working storage area to (FL+9)/10 words. When Z type records are being read and a zero byte terminator is not found within FL/10+1 words, an excess data error is returned. However, the examination of subsequent characters continues until the first terminator is encountered or a file boundary is reached. If the end-of-information is encountered before the zero byte is found, it is possible the file did not contain Z type records. At the conclusion of a read operation, the RL field of the FIT is set to the number of user characters read, not including blank padding.

When a record is written and the value of the RL field is not zero, the end of the record is determined by searching backwards from the character position specified by the value of the RL field for the first nonblank character. The zero byte is added in the nearest appropriate position. Binary zero-fill is done from the last significant character to the zero byte.

When a record is being written and the value of the RL field is zero, the end of the record is determined by searching backwards from the character position specified by the FL field. When a nonblank character is found, the zero byte is added in the nearest appropriate position. Binary zero-fill is done from the last significant character to the zero byte. If a nonblank character appears in the low-order position of the last word, the record written to the device is one word larger than the physical size of the record in the working storage area, because the nearest appropriate position for the zero byte is in the low-order 12 bits of the word past the character position specified by the FL field. The record on the output device is larger than the value specified in the FL field, but memory is not altered beyond the number of characters specified in the FL field.

If the last character of the record being written is : or %, one blank is appended. If the last word of the record contains nine characters, a zero is added to fill out the word and an additional zero word is appended (a total of 66 bits of binary zero).



59 Parity bit. Used to maintain odd parity within the control word.

58 Flag bit. Used in combination with the delete bit to signal the end-of-section or partition. Word count must be zero. The values are:

f	d	
1	1	End-of-section
1	0	End-of-partition
0	1	Deleted record
0	0	Normal record

57 Delete bit. A record in whose control word this bit is set is considered to be logically deleted from the file; the record is not passed to the user when the file is read. See the flag bit explanation.

56 thru 54 Reserved for users.

53 thru 44 Reserved for CDC.

43 thru 42 W-continuation record flags. The values are:

43	42	
0	0	Complete record
0	1	First piece of W-continuation record
1	0	Middle piece of W-continuation record
1	1	Last piece of W-continuation record

41 thru 24 The size of the previous record in central memory words including the control word for the record(needed for backspacing). This field is zero if there are no previous records.

23 thru 18 Number of rightmost unused bits in the last word ($0 \leq bits \leq 59$).

17 thru 0 Number of central memory words necessary to contain the record, not including the control word (word count decimal $\leq 2^{17}-2$). If the W-continuation record (WCR) flag is nonzero, the word count refers to only a part of the record.

Figure 2-11. W Type Record Control Word Format

A blank line written with the PUT macro appears on the external device as a blank character (55_8) followed by 54 bits of binary zero. On a coded 7-track S/L tape, a colon (00_8) written on the file is read as a zero (33_8) because of the BCD to display code conversion performed by the tape controller. Z type records give indeterminate results and should not be used on coded 7-track S/L tapes.

FILE INFORMATION TABLE 3

A file information table (FIT) is required for all files. Information in this table defines the file and how it is accessed. The FILE macro and FILE control statement are used to create and update the FIT. The FILE macro assembles a FIT in the COMPASS program where the macro is encountered. Pertinent information from the FILE control statement is saved on local file ZZZZZDG until OPENM time. When the file is opened, the saved information is stored into the FIT and takes precedence over any corresponding preexisting information. A blank FIT, except for addressing information and logical file name, could be set up in the user program with definition of file characteristics deferred until the file is opened.

Fields in the FIT can be changed using the STORE macro or the FILE control statement. The user identifies the fields by the keywords of the FILE macro. Fields in the FIT can be retrieved by using the FETCH macro and the keywords of the FILE macro.

Macro requests for file operations can result in amendment of the FIT fields. Certain macro operands are stored in FIT fields prior to performance of the request, and values in FIT fields can be stored as a result of processing the request. Also, certain fields in the FIT are maintained to reflect the current state of the file.

FILE MACRO

The FILE macro constructs the file information table at the address where the macro is encountered during assembly; the FIT must be built before the file is opened. The macro conforms to COMPASS coding conventions. The format of the FILE macro is shown in figure 3-1. The interaction between lfn and LFN=axxxxxxx is shown in table 3-1.

The FILE macro does not check fields for validity or consistency. Fields exceeding the maximum specified sizes are truncated; assembler warning messages are produced.

Misspelled or unrecognizable parameters generate null parameters, and the fields they reference are set to zero. Null parameters are ignored. Warning messages are generated when overlapping fields are specified.

The FILE macro should specify the file organization mnemonic. Any parameter not applicable to that file organization is ignored and an error type 4 message is generated during assembly.

The values specified for the other FILE macro parameters are assembled into the FIT; they can be specified in any order. Table 3-2 shows which FILE macro parameters are applicable to each file organization. An X indicates appropriate file organizations. (Note that the numbers appearing in parentheses are explained at the end of table 3-2.)

[lfn] FILE [LFN=axxxxxx] [,keyword=option, . . .]

lfn — Symbolic address where the FIT is assembled in the COMPASS program, and logical file name by which the file can be referenced if the LFN=axxxxxx is absent or the same name.

LFN — FIT field mnemonic for logical file name; it must be specified with axxxxxx if lfn is absent.

axxxxxx — Logical file name by which the file can be referenced, and symbolic address where the FIT is assembled in the COMPASS program if lfn is absent.

keyword — Symbolic name of the FIT field.

option — Selected option of the FIT field.

Figure 3-1. FILE Macro Format

TABLE 3-1. LFN AND lfn INTERACTION

Statement	COMPASS Location Value	Contents of First Word of FIT (lfn)
A FILE	A	A
FILE LFN=A	A	A
A FILE LFN=A	A	A
A FILE LFN=B	A	B

A detailed explanation of each FIT field that can be specified by the FILE macro parameter follows. The default value is indicated for each field.

The option aexp represents an absolute expression; exp is any COMPASS expression.

- ASCII

  ASCII character set bits for INTERCOM terminals. Not used under NOS.

Absent or ASCII=0	64-character display code.
ASCII=1	95-character ASCII subset.
ASCII=2	128-character ASCII.

TABLE 3-2. PARAMETERS FOR FILE MACRO BY FILE ORGANIZATION

FILE Macro Parameters	File Organization		FILE Macro Parameters	File Organization	
	Sequential (SQ)	Word Addressable (WA)		Sequential (SQ)	Word Addressable (WA)
ASCII	x		LFN	x	x
BBH	x	x	LL	$x^{(4)}$	
BFS	$x^{(1)}$	$x^{(1)}$	LP	$x^{(4)}$	
BT	x		LT	x	
B8F	x		LX	x	
CF	x	x	MBL	x	
CL	$x^{(2)}$		MNB	x	
CM	x		MNR	x	
CNF	x		MRL	x	x
CP	$x^{(2)}$		MUL	x	
C1	$x^{(2)(4)}$		NOFCP	x	x
DFC	x	x	OF	x	x
DX	x	x	PC	x	
EFC	x	x	PD	x	x
EO	x	x	RB	$x^{(5)}$	
ERL	x	x	REL		
EX	x	x	RMK	$x^{(6)}$	
FF	x		RT	x	x
FL	$x^{(3)}$	$x^{(3)}$	SB	$x^{(2)(4)}$	
FO	x	x	SBF	x	x
FWB	x	x	SPR	x	
HL	$x^{(2)}$		TL	$x^{(2)}$	
LA	x		ULP	x	
LBL	x		VF	x	x
LCR	$x^{(7)}$		WSA	x	x

Notes:

1. Length in words
2. T type records only
3. F and Z type records only
4. D type records only
5. K type blocks only
6. R type records only
7. I-O tape file only

- BBH

Buffer below highest high address. Refer to appendix E for a discussion of the BBH field and loading BAM.

Absent or BBH=NO Buffer is not below the highest high address.

BBH=YES Buffer can be below the highest high address if space is available.

- BFS

Buffer size in words.

Absent or BFS=0 BAM provides the buffer space if necessary; the first word address of the buffer (FWB) field is set to point to the first word address of the space obtained.

BFS=aexp	Buffer size; maximum is 2^{17}-1, or 131000 words. User specifies in words. A practical limit for BFS is (2^{18}/10)-1, or 262000, because this is the largest single move that can be processed without impacting performance.

● BT

Block type for sequential files; tapes are always blocked.

Absent or BT=I	Internal, block recovery control word; I type.
BT=C	Character count in characters per block; C type.
BT=K	Record count, m records per block; K type.
BT=E	Exact record count; E type.

● B8F

Flag for eight bit data. Round RL in PUTs for S type record down to exact multiple of 8 bits; used in FORM and 8-bit subroutines.

Absent or B8F=NO	S-record is written as RL number of 6-bit characters which the tape driver may round to the next higher multiple of 8 bits.
B8F=YES	S-record is written as (RLx6)/8 (rounded down) number of 8-bit characters; the tape driver will not add any extraneous characters.

● CF

Close flag. File positioning at CLOSEM time.

Absent or CF=R	Rewind.
CF=N	No rewind.
CF=U	Unload.
CF=RET	Return; rewind and unload.
CF=DET	Detach; no rewind.
CF=DIS	Disconnect terminal file.

● CL

Count field length of a T type record.

Absent or CL=0	No trailer count field defined.
CL=aexp	Length in characters of the trailer; maximum is 6.

● CM

Conversion mode.

Absent or CM=NO	No conversion.
CM=YES	Conversion between external and internal code for sequential tape files.

● CNF

Connect file flag.

Absent or CNF=NO	Normal mass storage file input/output.
CNF=YES	Terminal file.

● CP

Trailer count beginning character position of T type record.

Absent or CP=0	Beginning character position is zero.
CP=aexp	Beginning character position, numbered from zero on the left; maximum is 1310710.

● C1

COMP-1; format for the length field for D and T type records.

Absent or C1=NO	Field is display code.
C1=YES	Field is binary (COBOL COMP-1).

● DFC

Dayfile control.

Absent or DFC=0	Except for fatal errors, no dayfile messages are written.
DFC=1	Error messages are written on the dayfile.
DFC=2	Notes are written on the dayfile.
DFC=3	Errors and notes are written on the dayfile.

● DX

End-of-data exit routine address. The system stores a jump at the first address of the routine and control passes to the first executable statement, which is routine+1. If SBF is set to YES, the data exit is not taken.

Absent or DX=0	No routine is specified.
DX=exp	Address of the routine to be entered when an end-of-data condition occurs.

- EFC

Error file control (file ZZZZZEG).

Absent or EFC=0	No error file messages are written.
EFC=1	Error messages are written on the error file.
EFC=2	Notes are written on the error file.
EFC=3	Errors and notes are written on the error file.

The FITDMP macro forces EFC=0 to 2 and EFC=1 to 3.

- EO

Error option for parity error processing.

Absent or EO=T	Terminate the file; issue a fatal error.
EO=D	Drop bad data.
EO=A	Accept bad data.
EO=TD	Terminate the file, issue a fatal error, and display the block containing the parity error on error file ZZZZZEG.
EO=DD	Drop bad data and display the block containing the parity error on error file ZZZZZEG.
EO=AD	Accept bad data and display the block containing the parity error on error file ZZZZZEG.

- ERL

Trivial error limit.

Absent or ERL=0	No trivial error limit; an indefinite number of trivial errors is permitted.
ERL=aexp	Maximum number of trivial errors allowed before a fatal error occurs; maximum is 551.

- EX

Error exit routine address. The system stores a jump at the first address of the routine and control passes to the first executable statement, which is routine+1. If SBF is set to YES, the error exit is not taken by GET and PUT.

Absent or EX=0	No routine is entered if an error occurs, control is returned to the user's in-line code.
EX=exp	Address of the error exit routine to be entered when an error occurs.

- FF

File flush by operating system on abnormal termination; sequential files only.

Absent or FF=NO	Buffer not flushed.
FF=YES	Buffer flushed for an output file having scratch disposition.

- FL

Fixed length for F type records; full length for Z type records.

Absent or FL=0	Must be defined for open.
FL=aexp	Record length in characters for F type records, 10 through 1310710. For Z type records, 1 through 1310710; establishes the upper limit of characters or blank padding moved to the working storage area.

- FO

File organization.

Absent or FO=SQ	Sequential file.
FO=WA	Word addressable file.

- FWB

First word address of the buffer. If FWB is not provided by the user, the minimum buffer needed or the amount specified by the BFS field is provided.

Absent or FWB=0	No user-supplied buffer.
FWB=exp	Address of the buffer.

- HL

Header length; length of the fixed length portion of a T type record.

Absent or HL=0	Must be defined for open.
HL=aexp	Header length in characters cannot be less than CP+CL; maximum is 1310710.

- LA

Label area address.

Absent or LA=0	No area specified.
LA=exp	First word address of the label area.

● LBL

Label area length.

Absent or LBL=0	No label area length specified.
LBL=aexp	Length in characters; maximum is 900.

● LCR

Label creation flag

Absent or LCR=CRT	Create new label.
LCR=CHK	Check existing labels.

● LFN

Logical file name.

LFN=axxxxxx	axxxxxx is a one- to seven-character name beginning with a letter.

● LL

Length field length of a D type record.

Absent or LL=0	Must be defined for open.
LL=aexp	Length in characters; maximum is 6.

● LP

Beginning character position of the length field for a D type record.

Absent or LP=0	Beginning character position is zero.
LP=aexp	Beginning character position numbered from zero on the left; maximum is $10x(2^{17}-LL-1)$.

● LT

Label type.

Absent or LT=UL	Unlabeled.
LT=S	ANSI standard.
LT=NS	Nonstandard.
LT=ANY	Any label type. User label processing is not allowed.

● LX

Label routine exit.

Absent or LX=0	No user label processing routine supplied.
LX=exp	Address of the user-supplied label processing routine.

● MBL

Maximum block length in characters; should not be changed after OPENM.

Absent or MBL=0	The default depends on block type:
	BT=K error.
	BT=E error.
	BT=I MBL forced to 5120.
	BT=C MBL forced to 5120 characters for S tapes and BFS in characters minus 20 for L tapes; PRU devices considered unblocked.

MBL=aexp	Length of data block in characters. For K and E type blocks with Z type records, MBL must not be less than FL+10. For I type blocks, any MBL is overridden.

● MNB

Minimum block length for sequential file K and E type blocks.

Absent or MNB=0	No minimum block length specified.
MNB=aexp	Minimum block length in characters; maximum is MBL.

● MNR

Minimum record length of sequential file records.

Absent or MNR=0	Minimum length is zero. For BT=K or E and RT=F, MNR is set to MRL by default to prevent processing of block padding as valid data.
MNR=aexp	Minimum record length in characters; maximum is MRL.

● MRL

Maximum record length of D, R, T, U, and W type records.

Absent or MRL=0	No maximum record length; any record length is acceptable for PUT. No data is moved for GET.
MRL=aexp	Maximum length in characters; maximum is 1310710. Establishes the upper limit of characters moved to the working storage area.

- MUL

Multiple of characters per block in which sequential file K and E type blocks are written.

Absent or MUL=0	Characters per block is a multiple of 2.
MUL=aexp	Characters per block is a multiple of aexp; maximum is 62.

- NOFCP

No FILE control statement processing.

Absent or NOFCP=NO	FILE control statement processed at SETFIT or OPENM.
NOFCP=YES	FILE control statement not processed.

- OF

Open flag. File positioning at OPENM time.

Absent or OF=R	Rewind.
OF=N	No rewind.
OF=E	Position at end-of-information for extend.

- PC

Padding character for sequential file K and E type blocks. Specified in display code. PC must not be the same as the record mark character.

Absent	Padding character is 76_8 (ASCII circumflex or CDC logical NOT).
PC=ccB	Padding character is octal value cc; maximum is 77_8.

- PD

Processing direction.

Absent or PD=INPUT	Input (read).
PD=OUTPUT	Output (write).
PD=IO	Input-output (read and/or write).

- RB

Records per block in a sequential file K type block.

Absent or RB=0	RB set to 1.
RB=aexp	Blocking factor limit is 4095.

- REL

Key relation; relation of record key to key value at location KA.

REL=EQ	Specifies an equal to relation.
REL=LE	Specifies a less than or equal to relation. (Not applicable to extended indexed sequential files).
REL=GE	Specifies a greater than or equal to relation.
REL=LT	Specifies a less than relation. (Not applicable to extended indexed sequential files).
REL=GT	Specifies a greater than relation.

- RMK

Record mark character in display code. Used as the delimiting character with R type records. RMK must not be the same as the padding character.

Absent or RMK=0	Record mark is 62_8 (]).
RMK=ccB	Record mark is octal value cc; maximum is 77_8.
RMK=1Rx	Record mark is x; any character.
RMK=cc	Record mark is decimal value cc; maximum is 63.

- RT

Record type.

Absent or RT=W	Control word.
RT=F	Fixed length.
RT=R	Record mark.
RT=Z	Zero byte type.
RT=D	Decimal character count.
RT=T	Trailer count.
RT=U	Undefined.
RT=S	System-logical-records.

- SB

Sign overpunch; COBOL sign overpunch option for the length field for D and T type records.

Absent or SB=NO	Unsigned display code.
SB=YES	Sign-overpunch scheme used.

- SBF

Suppress buffer flag. Suppresses allocation of buffers and circular buffering. The GETWR and PUTWR functions do not require circular buffers for sequential files with S type records or files with K type blocks and the RB field set to 1. If all the records of a word addressable file are multiples of PRU size and start on PRU boundaries, the circular buffer is not used.

Absent or SBF=NO	Allocates buffers from the information given in the FWB and BFS fields.
SBF=YES	No buffer space is allocated.

Must be set to NO and a central memory buffer must be provided for an S type record PUTWR/GETWR to and from extended memory on the CDC CYBER 176 Computer System.

- SPR

Suppress read ahead. Used by CRM during processing of blocks with read parity error and processing of EOFs on S and L tapes; unconditionally reset to NO at the end of processing.

Absent or SPR=NO	Read ahead.
SPR=YES	Read only one block at a time.

- TL

Trailer length of a T type record.

Absent or TL=0	Must be defined for open.
TL=aexp	Specified in characters; maximum is 131071.

- ULP

User label processing. (See section 6.) Specifies conditions that transfer control to the user label processing routine.

Absent or ULP=NO	None.
ULP=V	VOL/EOV.
ULP=F	HDR/EOF.
ULP=VF	VOL/HDR/EOV/EOF.
ULP=U	UVL/UHL/UTL.
ULP=VU	VOL/EOV/UVL/UHL/UTL.
ULP=FU	HDR/UHL/EOF/UTL.
ULP=VFU	All.

- VF

Volume close flag. Volume positioning at CLOSEM time.

Absent or VF=U	Unload volume (reel).
VF=R	Rewind volume (reel).
VF=N	No rewind.

- WSA

Working storage area address. Must be set before any file processing command uses the working storage area. It can be set by macros GET, PUT, and REPLACE.

Absent or WSA=0	Address of the working storage area starts at word zero of the user field length.
WSA=exp	Address of the working storage area.

FILE CONTROL STATEMENT

With the FILE control statement, the user specifies file information to update the FIT when the SETFIT macro is issued, or the first time the file is opened in the job step. FILE control statements are not processed if NOFCP is set to YES by the FILE or STORE macro. This run-time control over file specification allows a single program to process files with different record or block types. Corresponding FIT fields have the value specified on the last control statement encountered.

FILE control statements must be placed before any program call in which the information on them is to be used. Data on this control statement is stored on local file ZZZZZDG by the FILE control statement processor. Because processing of the FILE control statement involves calling a central processor program, it should not be placed within a load set sequence, for example, between a LOAD and an EXECUTE. If more than one FILE control statement appears for a given file, the data on the first control statement can be overwritten by the data on a subsequent statement when overlapping fields occur on those statements. The FILE control statement conforms to operating system coding conventions.

A FILE control statement cannot be continued to a second card or card image, but the same logical file name can appear on more than one FILE control statement.

If an error diagnostic is produced by FILE control statement processing, the entire statement is ignored. FILE control statement diagnostics are written on the dayfile as soon as the error is encountered; they name the faulty parameter and are self-explanatory. Control is passed to the next EXIT if an error occurs in FILE control statement processing. No EXIT is taken for advisory diagnostics.

The FILE control statement format is shown in figure 3-2. FILE control statement keyword options can be specified in any order. Keywords have the same meaning as described for the FILE macro.

If only the lfn parameter appears in the FILE control statement, the FIT fields for that file revert back to those specified in the program for all succeeding job steps, unless another FILE control statement references that file. If the FILE control statement appears without any parameters, FIT fields for all files revert back to those specified in the program for all succeeding job steps until another FILE control statement is encountered.

FILE (lfn[=axxxxxx] [,keyword=option] . . .)

lfn — Name of a FIT; required.

=axxxxxx — Optional new name for the FIT; allows a file to be requested by a new name without reassembly.

keyword=option — Symbolic name of the FIT field and the option selected.

Figure 3-2. FILE Control Statement Format

The FILE control statement parameters are listed in table 3-3. The various options for a keyword are separated by the | symbol. If the keyword is selected, one of the options must be selected and the others must be omitted.

Parameter values are absolute; generally they refer to number of characters. Value formats are denoted as:

n...n	Value is decimal
n...nB	Value is octal
n...nW	Value is decimal, specified in words

Parameter values for the FIT fields that can be set by the FILE control statement are the same as the parameter values for the FILE macro. The parameter values for the FIT fields that can be set by the FILE control statement but not by the FILE macro are as follows:

- MFN

  Multifile set name.

  MFN=axxxxxx — axxxxxx is the one- to seven-character name beginning with a letter.

- PNO

  Multifile position number. Specifies the position number of the member file on the multifile set.

  PNO=aexp — aexp is the position number in display code.

TABLE 3-3. FILE CONTROL STATEMENT PARAMETERS

Keyword	Options	Keyword	Options
ASCII	0|1|2	LT	S|NS|UL|ANY
BBH	NO|YES	MBL	0|n . . . n|n . . . nB|n . . . nW
BFS	0|n . . . n|n . . . nB	MFN	file name
BT	I|C|K|E	MNB	0|n . . . n|n . . . nB|n . . . nW
B8F	NO|YES	MNR	0|n . . . n|n . . . nB|n . . . nW
CF	R|N|U|RET|DET|DIS	MRL	0|n . . . n|n . . . nB|n . . . nW
CL	0|n . . . n|n . . . nB|n . . . nW	MUL	0|n . . . n|n . . . nB
CM	NO|YES	OF	R|N|E
CNF	NO|YES	OMIT	macro name/macro name/. . .
CP	0|n . . . n|n . . . nB|n . . . nW	PC	0|n . . . n
C1	NO|YES	PD	INPUT|OUTPUT|IO
DFC	0|1|2|3	PNO	0|n . . . n|n . . . nB
EFC	0|1|2|3	RB	0|n . . . n|n . . . nB
EO	T|D|A|TD|DD|AD	REL	EQ|LE|GE
ERL	0|n . . . n|n . . . nB	RMK	0|ccB|1Rx|cc
FF	NO|YES	RT	W|F|R|Z|D|T|U|S
FL	0|n . . . n|n . . . nB|n . . . nW	SB	NO|YES
FO	SQ|WA	SBF	NO|YES
HL	0|n . . . n|n . . . nB|n . . . nW	SPR	NO|YES
LBL	0|n . . . n|n . . . nB|n . . . nW	TL	0|n . . . n|n . . . nB|n . . . nW
LCR	CHK|CRT	ULP	NO|V|F|VF|U|VU|FU|VFU
LFN	lfn	USE	macro name/macro name/. . .
LL	0|n . . . n|n . . . nB	VF	U|R|N
LP	0|n . . . n|n . . . nB|n . . . nW		

RUN-TIME MANIPULATION

The user can communicate with BAM through the FIT without knowing the exact format of the FIT. This is done with the FETCH, STORE, and SETFIT macros.

FETCH

The FETCH macro retrieves the contents of a specified FIT field by a reference to its mnemonics. The macro format is shown in figure 3-3.

FIT field mnemonics can be any of the keywords used with the FILE macro, or any of the fields listed in figure 3-3. The macro generates code to extract the requested value from the FIT. The code expansion destroys values in user registers Xf, Xm, Af, and Xi (which can be Xf or Xm).

FETCH fit,keyword,Xi,f,m

fit — Logical file name address of the FIT, or any COMPASS expression giving the FIT address. If fit is Xf or Xm, its contents are changed upon return.

keyword — Any of the keywords in the FILE macro, FILE control statement, or any of the following (when the keyword represents a length, the length is returned as characters):

BN	Block number
ECT	Error count
ES	Error status (equivalent to IRS)
FNF	Fatal error flag
FP	File position field
IRS	Error code
LOP	Last operation
NOFCP	No FILE control statement processing
OC	Open/close status
PEF	Parity error flag
PTL	Partial transfer length
RC	Record count
RL	Record length
SES	System error severity
VNO	Volume number
WA	Current word address
WPN	Write bit

f — Number of the X register used to fetch the FIT word. Must be 1 through 5 (default is 5).

m — Number of the X register used as a mask (default is 7).

Xi — X register to receive the value of the requested field. If keyword represents a 1-bit field, it is returned in the sign bit. Keywords that are file names are returned left-justified with zero fill; otherwise, the keyword is returned right-justified with zero fill.

Figure 3-3. FETCH Macro Format

STORE

This macro places a user-determined value in a FIT field at execution time. The format of the STORE macro is shown in figure 3-4. The STORE macro generates code to store the requested value in the FIT. This code expansion destroys the values in user registers Xf, Xs, Xm, Af, As, and Xi (which can be Xf, Xs, or Xm).

STORE fit,keyword=$\left\{\begin{matrix}\text{option}\\ \text{Xi}\end{matrix}\right\}$,f,s,m

fit — Logical file name address of the FIT, or any COMPASS expression giving the address or a tag.

keyword — Any keyword described in connection with the FILE macro.

option — Options associated with the keyword.

Xi — X register containing the proper code for the keyword. When the keyword represents a length, it is specified in characters.

f — Number of the X register used to fetch the FIT word. Must be 1 through 5 (default is 5).

s — Number of the X register used to store the FIT word. Must be 6 or 7 (default is 6).

m — Number of the X register used as a mask (default is 7).

Figure 3-4. STORE Macro Format

Most FIT fields listed in appendix D can be set symbolically by STORE. Some fields are protected against a STORE; others, such as the structure of a sequential file, are not protected but should not be changed after the file has been opened.

A parameter can be set by using the option with the keyword, or by using a register to hold the option as shown in figure 3-5. Examples a and b have an identical effect, just as c and d have an identical effect.

```
a.  STORE fit,RL=10

b.  SX1    10
    STORE fit,RL=X1

c.  STORE fit,FO=SQ

d.  SX1    0
    STORE fit,FO=X1
```

Figure 3-5. STORE Macro Examples

SETFIT

The SETFIT macro sets fields in the FIT. The macro format is shown in figure 3-6. The SETFIT macro makes FILE control statement information available without the need for complete OPENM processing. This makes it possible for system routines to obtain information, such as run-time buffer requirements, needed by other system routines.

SETFIT fit

fit Address of the FIT, or an X register containing the address of the FIT.

Figure 3-6. SETFIT Macro Format

Also, SETFIT allows the user to cause FILE control statement processing when it would not otherwise occur. Values in all user registers are destroyed.

SETFIT is valid only for a closed file. Once FILE control statement values are placed in the FIT, the macro sets the FILE control statement processed flag (PDF) field of the FIT to inhibit further FILE control statement processing during OPENM. The flag is cleared during subsequent OPENM processing.

If SETFIT is issued and the user setting for the buffer size (BFS) field is zero, the BFS field is set to the buffer size normally allocated, based on other FIT values.

FILE PROCESSING 4

This section explains the logical operations of processing a sequential or word addressable file and explains macros as applicable to each file organization. For a general explanation of all macros and a detailed listing of their parameters, refer to section 5.

Before a file can be processed, the user must establish a file information table (FIT). Establishing the FIT sets a name by which the file can be referenced and defines the file structure and processing limitations. This table contains fields that are referenced whenever BAM processes the file. FIT fields can be set prior to file processing by the FILE control statement, FILE macro, SETFIT macro, and STORE macro.

SEQUENTIAL FILES

In addition to the file manipulation macros, the following macros can be used to process a sequential file:

CHECK,CHECKR

CLOSEM

ENDFILE

GET,GETP,GETWR

OPENM

PUT,PUTP,PUTWR

REPLACE

REWINDM

SKIPdu

WEOR

WTMK

All record types are applicable for sequential files. Except for S type records, records in a sequential file are physically grouped into blocks. Once the user has defined the record and block type, BAM performs all the manipulations required for block construction. Sequential files can reside on mass storage devices or magnetic tape; files with K or E type blocks can reside only on S/L tapes.

OPEN PROCESSING

All files must be initialized using the OPENM macro. Before opening a file, however, the user must call for construction of the FIT by specifying the logical file name. The file organization can also be specified, but the default is sequential.

The record type (RT) and block type (BT) fields, and any other fields needed to describe record and block type, must also be specified before a new file can be opened. For certain system special-named files, BAM forces the values of the RT, BT, and FL fields of the FIT, as shown in table 4-1. The user can override these values with the FILE control statment.

TABLE 4-1. SYSTEM FILES FORCED VALUES

System File	Forced Values
lfn=INPUT	RT=Z, BT=C, FL=80
lfn=OUTPUT	RT=Z, BT=C, FL=140
lfn=PUNCH	RT=Z, BT=C, FL=80
lfn=PUNCHB	No forced value

Consistency checks are performed on certain FIT fields when the file is opened the first time in a job step. Table 5-1 in section 5 lists the fields that are checked for consistency. If a file is closed and then reopened and the close flag (CF) field of the FIT is set to R or N, consistency checks and complete FILE control statement processing are not repeated.

The following fields can be specified prior to opening a file, but need not be set in the FIT until they are required by file processing commands; they can change at any time during a subsequent file processing run:

DX	End-of-data exit; default is no end-of-data routine
EX	Error exit; default is no error routine
ERL	Trivial error limit; default is an indefinite number of trivial errors permitted
DFC	Dayfile control; default is only fatal errors listed
EFC	Error file control; default is no error messages

If label processing is specified, it is initiated during OPENM processing. A conflict between labels specified on the REQUEST or LABEL statement and the label type (LT) field causes an informative dayfile message and inhibits user label processing. When a labeled file is opened, label checking and creation are based on the label creation flag (LCR) field of the FIT. Refer to section 6 for further information about label processing.

INPUT/OUTPUT PROCESSING

The GET and PUT macros and variations of these macros read and write files. A working storage area must be established to pass data to and from the program and a file storage device. The user defines the working storage area (WSA) by supplying an address for the WSA field of the FIT. This is normally done when the GET or PUT macro is issued. A GET macro transfers data from the buffer area to the working storage area. A PUT macro transfers data from the working storage area to the buffer area.

If only the GETWR, PUTWR, REWINDM, and SKIP macros are to be used for files with logical and physical records equivalent, the suppress buffer flag (SBF) field of the FIT can be set to YES. The file must have S type records, or K type blocks with one F or U type record per block. If these restrictions are observed, field length requirements are reduced and central processor time required for each input/output operation is reduced. The elapsed time required to obtain input/output overlap with processing is dependent on the use of the CHECK or CHECKR macro. If the restrictions are not observed, processing advantages do not apply and the use of CHECK or CHECKR is redundant.

Input Processing

The maximum record length (MRL) field of the FIT must be set by the user for reading a file. If the MRL field is zero, no data is transferred from the buffer to the working storage area. If the MRL field is not zero, that value becomes the upper limit for the number of characters transferred even if the record exceeds that length.

Records in a sequential file are read in the order that they occur in the file. They can be read as whole or partial records.

The GET macro reads whole records. The record length (RL), record count (RC), and block number (BN) fields of the FIT are updated during processing. Data transfer always starts at the next record available. If a GET macro is issued when the file is positioned at midrecord because of a prior GETP macro, a skip is made to the record boundary before beginning the GET operation. When the GET macro encounters any end-of-data condition, control is passed to the end-of-data routine.

If the amount of data indicated by the W control word or by the contents of a length or record mark character field is greater than the value specified by the MRL field, the record is truncated to the number of characters specified by the MRL field and an excess data error is returned. If the amount of data is less than the value specified by the fixed length (FL) field on F type records or less than the indicated record length on other types, an insufficient data error is returned.

If the number of characters is not a multiple of ten, the remaining characters from RL to the word boundary are defined. This is applicable to all record types except Z.

At the conclusion of a successful read operation, the value of RL field is the same as the value specified for the RL field for the operation requested. At the conclusion of a read with an insufficient data error, the RL field reflects the number of characters transferred to the working storage area.

The GETP macro transfers part of a record to the user working storage area. The partial transfer length (PTL) field specifies the number of characters to be transferred. At the end of the GETP operation, the PTL field indicates the number of characters actually transferred. The value of the PTL field at transfer completion is the same as the transfer requested unless a record boundary or error condition is encountered.

If the GETP operation initiates record transfer, the EOR flag in the file position (FP) field of the FIT is cleared. When the last data of the record is transferred, the EOR flag is reset. A GETP operation does not cross record boundaries.

The GETP macro transfers characters from the beginning of a record or from the next character available in the record. If the SKIP parameter is specified, however, transfer begins at the start of the next record if current position is within a record. The SKIP parameter is ignored if current position is at the beginning of a record. When the first GETP macro for a record is issued, the RL field is cleared. At the completion of each GETP operation, the RL field is updated to indicate the number of characters read so far.

For U type records, the RL field must be used to specify total record length prior to issuing the first GETP macro for the record. If the length of an S type record is unknown, the user must make a series of GETP requests for PTL characters, where PTL is the length of the working storage area. When the first GETP macro is executed, the FP field of the FIT is set to zero to indicate position in the midst of a logical record. When a subsequent GETP macro completes record retrieval, the EOR flag of the FP field is set, and the length of an S type record becomes known. Consequently, the user must check the FP field for EOR to determine when the record boundary has been reached.

For D and T type records, the first GETP macro for a record must initiate transfer of at least the number of characters specified by the value of the minimum record length (MNR) field. For R type records, the GETP macro is not valid.

S type records can be larger than 2^{23}-1 characters. In this case, RL is mod 2^{22}.

The GETWR macro initiates the transfer of data in units of words and transfers control to the user. The GETWR macro is intended for use in conjunction with the suppress buffer option. Refer to GETWR processing in section 5 for a complete description of the macro.

Output Processing

The PUT, PUTP, and PUTWR macros write data to a sequential file. An existing file can have records added to it after the previous EOI.

The MRL field need not be set to execute a PUT or variation of a PUT. When a record is transferred from the working storage area to the buffer and the MRL field is set to zero, any number of characters can be written. If the MRL field is not zero, that value becomes the upper limit on the number of characters that can be transferred.

The PUT macro writes an entire record. Data transferred by the PUT macro is written immediately following the last data written to the file. Each PUT operation creates a new record. On R type records, the user must place the record mark character in the record. On D and T type records, the user must set the control fields. The record count (RC) and block number (BN) fields are updated by BAM when record and block boundaries are crossed.

The PUTP macro transfers part of a record from the working storage area. The user must set the PTL field to specify the number of characters to be written. The execution of the first PUTP macro begins a new record. The second PUTP macro writes characters immediately after the last character written. For R type records, the PUTP macro is not valid. The RL field can be specified for the first PUTP macro for S, U, Z, and W type records. If the RL field is zero for Z type records, the value of the FL field is used. For all other record types, the value of the RL type field is determined by BAM. When the number of characters equal to the RL value have been transferred, the record is terminated. S, U, and W type records can use the TERM parameter on the last PUTP to terminate the record. An indefinitely long S type record can be written by using a series of PUTP macros followed by a WEOR of any level, or a PUTP macro with the TERM parameter specified.

The PUTWR macro initiates the transfer of data in units of words and transfers control to the user. The PUTWR macro is intended for use in conjunction with the suppress buffer option. Refer to GETWR processing under the GET macro discussion in section 5 for a complete description of the macro.

A file can be updated using the REPLACE macro. The REPLACE macro replaces the last record read with a record from the working storage area. The replacement record has the same record length as the record being replaced, and it must be a mass storage file. The record type can be W or F only; the block type must be C.

Processing 9-Track Binary S/L Tapes

Nine-track tapes must record multiples of eight bits; however, BAM deals exclusively in 6-bit characters. If the data being written is not a multiple of eight bits, the tape driver rounds it up to the next multiple of eight bits. If the data being read from the 9-track device is not a multiple of six bits, BAM rounds it up to the next multiple of six bits. If the file is repeatedly copied, a block can contain up to three extraneous undefined 6-bit characters before it is a multiple of six and eight.

To compensate for this, the user of S type records can either set the maximum record length (MRL) field to three characters larger than the actual data size or ignore the excess data errors. For record types other than S, the user can specify a value of greater than three for the minimum record length (MNR) field; BAM then ignores three or less extraneous characters at the end of the block.

To avoid the extraneous characters when the user is processing 8-bit data in S type records, the record length (RL) field should specify a value rounded up to the next multiple of six, and the B8F field should be set to YES. This causes BAM to write the next lower multiple of eight bits to the device.

FILE POSITIONING

The REWINDM macro repositions a mass storage file to the BOI. REWINDM positions labeled tapes to a point after the labels at the beginning of the first file volume. REWINDM positions unlabeled tapes to the load point of the volume currently mounted.

The SKIPdu macro repositions an existing sequential file forward or backward. The user must specify the direction of the skip, the type of units to be skipped, and the number of units to be skipped.

Backward Skipping

A file positioned at unit number m with a skip count of n is positioned to unit m minus n upon completion of the skip backward. Positioned at a unit means ready to read beginning at that unit. The position of a file after a SKIPBu of two units is shown in figure 4-1.

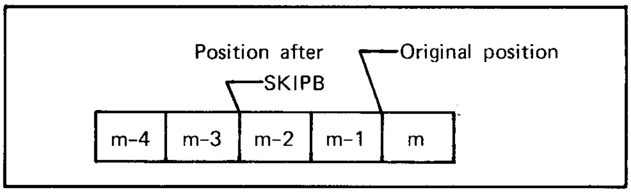


Figure 4-1. SKIPBu Positioning

If an input file is positioned at midrecord when a SKIPBu macro is issued, operation is as if the file were positioned at the end of that unit. If a file is positioned at midrecord when a SKIPBu macro with a zero count is issued, the file is positioned to the start of that unit. A SKIPBL macro after the execution of a PUTP macro that did not terminate a record is an error.

No automatic volume switching occurs when a SKIPBu macro is issued for a multivolume tape file. An error results if the load-point is reached. If a boundary condition is detected before the skip count is exhausted, control is transferred to the end-of-data routine with the appropriate file position set. The file is left positioned immediately before the delimiter. The boundary conditions are:

SKIPBL	Section, partition, beginning-of-volume
SKIPBP	Partition
SKIPBu	Beginning-of-information, load point on a tape file

The restrictions on SKIPBu, with respect to record and block type, are as follows:

- SKIPBL is not supported for T, R, U, and D record formats, or for K and E type blocks.

- If SKIPBL is attempted when a file residing on a PRU device with C type blocks and F type records is positioned at EOS, EOP, or EOI, it is not possible to determine the exact record boundary. If the fixed length (FL) field is not a multiple of 10, positioning can be unpredictable (not a record boundary).

Forward Skipping

A file positioned at unit number m with a skip count of n is positioned to unit m plus n upon completion of the skip forward. Positioned at a unit means ready to read that unit. If a file is positioned at midrecord whan a SKIPFu macro with a zero count is issued, the file is positioned forward to the unit boundary. If a file is positioned in the middle of a record when a SKIPFu macro with a nonzero count is issued, the file is positioned forward to the unit boundary, and then positioned forward the number of units specified. A SKIPFL macro is not allowed with U type records. An output file cannot be positioned forward.

If a boundary condition is detected before the skip count is exhausted, control is transferred to the end-of-data routine with the appropriate file position set. The file is left positioned immediately after the terminator. The boundary conditions are:

SKIPFL	Section, partition
SKIPFP	Partition
SKIPFu	End-of-information

CLOSE PROCESSING

At completion of processing, a file must be closed by the CLOSEM macro. Any remaining records of an output file are written from the buffer to the file storage device; the open/close flag (OC) field of the FIT is set to closed; the action designated by the close flag (CF) field of the FIT is performed; and control is returned to the user.

It is important that all files be closed. During normal termination, the error file buffer is flushed when the last file in a job step is closed. Therefore, error information can be lost if all files are not closed.

End-of-Data Processing

End-of-data occurs when an input/output data transfer or positioning operation is attempted and there is no more data or space on the file, partition, section, or volume because one of the following end conditions was encountered:

- End-of-information

- End-of-partition

- End-of-section

The end-of-data exit (DX) field specifies the address of a user routine for processing an end-of-data condition. When an end-of-data condition exists, control is passed to the address (DX)+1. A jump back to the user in-line return code is stored at the DX address. The file position (FP) field specifies the end condition that caused the transfer of control to the end-of-data exit. No end-of-data exit is taken if SBF is set to YES.

The only requests permitted for sequential files opened for input, after file position EOI has been set, are CLOSEM, REWINDM, and SKIPBu. The only requests permitted for I-O sequential files, after file position EOI has been set, are CLOSEM, SKIPBu, REWINDM, ENDFILE, or PUT.

A GET operation that transfers control to the end-of-data exit does not transfer data to the working storage area. Transfer of control to a user's data exit is an empty GET in that no more data remains; therefore, an end-of-data condition exists. The FP field is not set until a file is logically at the position specified.

Caution must be taken with short records, since PRU devices always contain blocks which are a multiple of 10 characters. EOS, EOP, and EOI are not always correctly detected on a file on a PRU device with F, R, U, D, or T type records and C type blocks when the value of the RL field is less than 10 characters. The padding that has been added to the final block of the file can be greater than or equal to the length of the record. The EOI is not recognized and the padding is processed as valid data.

File Boundary Processing

The CLOSEM macro must be issued to ensure proper EOI processing. The buffer is flushed and, except for unlabeled S/L tapes, an EOI is written to the file. A CLOSEM request for an OUTPUT or I-O sequential file can cause trailer records to be written for W type record files. A deleted zero-length record is written on OUTPUT or I-O sequential files.

Label processing is performed, if appropriate. Label processing performed on I-O sequential files is controlled by the last operation on the file. If the operation was output, labels are created. If the operation was input, labels are checked. On any input labeled file, label checking is performed only if the end-of-information or end-of-volume has been reached. Control is transferred to the user-supplied label routine, if one has been specified.

The CLOSEM VOLUME request forces volume switching to the next reel of a multivolume file. If a value is not supplied for the CF field with the CLOSEM macro, the value in the volume close flag (VF) field of the FIT is used. The current volume number (VNO) field of the FIT is incremented when volumes are switched.

The following actions occur when N is used to specify no rewind on a multivolume file opened for OUTPUT:

- Unlabeled S/L Tapes

  Two tapemarks are written.

  Volume is rewound or unloaded (N parameter is overridden).

  New volume is requested by the system and checked to verify that it contains no labels that would prevent writing on it.

  Data transfer continues on the new volume.

- Unlabeled SI or I Tapes

  Default tapemark and EOV1 label is written.

  Volume is rewound or unloaded (N parameter is overridden).

  New volume is requested by the system and checked to verify that it contains no labels that would prevent writing on it.

  Data transfer continues on the new volume.

- Standard Labeled S/L and SI or I Tapes

  If the user has issued a CLOSEM/VOLUME causing the buffer to be flushed, or if the system has detected an end-of-tape, the following occurs:

  Control is passed to label routine exit (LX) if defined.

  EOV labels are written.

  Volume is rewound or unloaded.

  Control passes to the LX address if defined.

  New volume is requested and checked.

  BOV labels are written.

  Data transfer continues.

Divisions larger than a record can be specified by issuing a macro to write an end-of-section, end-of-partition, or end-of-information. A partition can be terminated with the ENDFILE macro. Before the EOP is written, the buffer is flushed. The results of ENDFILE depend on the format of the file as described under a description of the macro in section 5.

A section can be terminated by using the WEOR macro. Before the EOS is written, the buffer is flushed. The results of WEOR depend on the format of the file, as shown under a description of the macro in section 5.

The purpose of the WTMK macro is to write tapemarks in nonstandard label processing. It should not be used elsewhere.

TERMINAL FILE PROCESSING

BAM uses a specialized capsule for processing files on terminal devices. It processes Z and S type records. W and U type records can also be specified, but they are processed as S type records. D, T, R, and F type records cannot be specified.

If the file device type is terminal, BAM sets the connect file (CNF) field of the FIT to YES during open processing. If the user sets the CNF field to YES, BAM connects the file. CNF must be set only on a closed (or never opened) file so that the connect request can be processed at open time.

The user need not reserve buffer space for terminal files. BAM uses file ZZZZZOU to write data to the terminal. The user defined FIT is used for reading; data is read directly to the working storage area.

Under the NOS/BE operating system, the type of character set used can be specified by setting the ASCII field of the FIT. If this field is nonzero, the record length (RL) field is still treated as the number of 6-bit characters to be read or written, but blank stripping and padding is done using a 12-bit ASCII blank (0040B) instead of a 6-bit display code blank (55B). This ensures no extraneous display code blank (55B) being added or removed from ASCII files.

If the last character of the record being written is a colon (:), BAM appends one blank character (55B).

Under the NOS/BE operating system, an input file can be terminated with a %EOF to ensure an end-of-data exit. A %EOR sets FP to EOS and takes the end-of-data exit.

A terminal file can be closed and disconnected by setting the cf parameter to DIS with the CLOSEM macro. For the file that is to be used subsequently as a disk file, the CNF field can be set to NO and the file reopened. For an opened disk file that is to be used as a terminal file, the file can be closed with a CLOSE/DET, CLOSE/RET, or CLOSE/U; CNF set to YES; and the file reopened.

Programs doing terminal I/O and using static loading must use the special names TGET and TPUT with the USE parameter on the FILE control statement to load the special terminal I/O capsule. Refer to appendix E for a discussion of static and dynamic loading.

WORD ADDRESSABLE FILES

In addition to the FIT manipulation macros, only the following macros can be used to process word addressable files:

CLOSEM

GET

OPENM

PUT

Word addressable files must reside on mass storage. Only record types F, U, and W can exist in word addressable files.

OPEN PROCESSING

All files must be initialized using the OPENM macro. Default values are inserted into FIT fields for certain values not supplied prior to open processing.

When a file is opened as a new or existing file, the user must have previously set the record type (RT) field of the FIT, or the default of W type records is set. For F type records, the fixed length (FL) field must also be set.

The following FIT fields can be set before the file is opened and should not be changed until another open is executed:

PD	Processing direction; default is INPUT
FWB	First word address of the buffer; default address is supplied by BAM
BFS	Buffer size; default of minimum space is provided except when the suppress buffer flag (SBF) field is set to YES; a BFS supplied by the user is rounded down to a multiple of PRU size plus 1

The following FIT fields need not be set until they are required by file processing commands and can be changed at any time:

EX	Error exit; default is no error routine
DX	End-of-data exit; default is no end-of-data routine
MRL	Maximum record length; default is 0

Certain consistency checks are performed on FIT fields when the file is opened. Table 5-1 in section 5 lists the fields that are checked for consistency.

INPUT/OUTPUT PROCESSING

The GET and PUT macros read and write files. A working storage area must be established to pass data to and from the program and a file storage device. The user defines the working storage area (WSA) by supplying an address for the WSA field of the FIT. This is normally done when the GET or PUT macro is issued. A GET macro transfers data from the buffer area to the working storage area. A PUT macro transfers data from the working storage area to the buffer area.

If all the records of a word addressable file are multiples of PRU size and start on PRU boundaries, the circular buffer is not used to process the records. The suppress buffer flag (SBF) field can be set to YES, and no buffer is allocated. If a record is encountered that is not a multiple of PRU size and does not start on a PRU boundary, an error is issued.

BAM uses the word address (WA) field of the FIT to determine where to read or write data. When a file is opened as a new file, the WA field is set to 1. It is updated after every read or write. If a sequential read or write is desired, the WA field need not be reset by the user.

Any mass storage file can be processed as a word addressable file. Allowances must be made for short PRUs and level numbers; these can be present as a result of previous system processing. An attempt to retrieve word addresses between the end of the short PRU and the start of the next PRU returns an invalid word address error. A read that continues past the short PRU returns an insufficient data error. A read of a level 0 to 16_8 indicator returns an end-of-section; a read of a zero-length level 17_8 returns an end-of-partition.

Writing a record into any part of a short PRU causes that PRU to be rewritten as a full PRU without comment. End-of-section or end-of-partition status no longer exists. These files cannot have been written as word addressable files but must have originally been written as sequential files.

The end-of-data exit (DX) field specifies the address of a user routine for processing an end-of-data condition. An end-of-data exit is taken on an end-of-section or end-of-partition in a W control word. Control is passed to the address (DX)+1. A jump back to the user in-line return code is stored at the DX address. The file position (FP) field specifies to the end-of-data exit. A read at the end-of-information takes an end-of-data exit with the file position (FP) field of the FIT set to EOI.

Input Processing

The maximum record length (MRL) field of the FIT must be set by the user for reading a file with U or W type records; the fixed length (FL) field must be set by the user for reading a file with F type records. When a record is transferred from the buffer to the working storage area and the MRL field is zero, no data is transferred. If the MRL field is not zero, that value becomes the upper limit for the number of characters transferred even if the record exceeds that length.

A file is read by the GET macro. The RL field is used with U type records only; it must be set. After the GET macro is executed, the RL field contains the number of characters read. W type records are actually one word longer than the RL value returned to allow for the control word. The user must allow for this when calculating the value for the WA field for random access. When a W type record is read, only RL characters are returned to the working storage area. The control word is not returned. If the amount of data indicated by the W type record control word or by the contents of a length field is greater than the value of the MRL field, the record is truncated to the number of characters specified by the value of the MRL field and an excess data error is returned.

If the number of characters is not a multiple of ten, the remaining characters from RL to the word boundary are undefined.

Output Processing

The MRL field need not be set to execute a PUT macro. Any number of characters can be written when a record is transferred from the working storage area to the buffer, if the MRL field is set to zero. If the MRL field is not zero, that value becomes the upper limit on the number of characters that can be transferred.

A file is written by the PUT macro. The RL field must be set for U and W type records. The length specified need not be a multiple of 10; however, writing always begins at the left on a word boundary. If the previous write was not a full word, the rightmost character positions are undefined and the next write begins on a new word.

If the value of the WA field is beyond the EOI of the current file, the file is automatically extended and all indications of the previous EOI are gone. Word addressable files are extended in multiples of PRUs. BAM maintains a pointer to the physical EOI but not to the user EOI. If the contents of the file do not require a complete multiple of a PRU, the physical EOI and the user EOI are different.

CLOSE PROCESSING

At completion of processing, a file must be closed by the user with the CLOSEM macro. Any remaining records of an output file are written from the buffer to the file storage device; the open/close flag (OC) field of the FIT is set to closed; and control is returned to the user.

MACROS 5

Macros are used for processing the files established with the FILE macro and control statement. All macros reside on COMPASS system text IOTEXT, which must be specified by the S=IOTEXT parameter on the COMPASS control statement at assembly time. An alphabetical listing of all macros with their parameters in COMPASS format is included in this section.

DESCRIPTIVE CONVENTIONS

The macros conform to COMPASS syntax. The location, operation, and variable fields are separated by one or more blanks. In the macro parameter strings, the fit parameter is required. All others are optional and positional. When optional parameters are omitted, their positions must be marked by commas; trailing commas can be omitted.

For example, the format of the OPENM macro is:

```
    OPENM fit,pd,of
```

If the pd parameter is not used when the OPENM macro is issued, the format is:

```
    OPENM fit,,of
```

The first parameter of every macro identifies the file information table for the referenced file. If the address specified by the fit parameter is invalid, the results are indeterminate. It can be specified by any of the following:

lfn	Location field name of the first word of the FIT, one through seven alphabetic or numeric characters
Rn	Any A, B, or X register containing the FIT address
exp	Any COMPASS expression giving the address of the FIT

When elements are stacked in braces { }, one must be chosen; the others must be omitted. Only parameters applicable to the file organization set in the FIT should be specified. Supplying parameters applicable to the other file organization could cause erroneous results.

MACRO EXECUTION

The current contents of the FIT are used for macro execution. Because the last value set in the FIT is used for execution, default values identified in the macro parameter lists are valid only if the FIT fields have not been changed previously. FIT fields can be set by any of the following:

- FILE macro parameters

- FILE control statement parameters, which can override defaults during open

- A SETFIT macro, which can call for FILE control statement processing without full open processing

- Individual fields, which can be set by the STORE macro before or after open

- Defaults, which can be set during open

- Parameters specified in processing macros that are moved to the FIT before file processing occurs (a zero value in a parameter list moves a zero to the FIT; a null value does not affect the FIT)

The user should presume all registers are destroyed during macro execution. Registers are not saved or restored.

The user macros, with the exception of FETCH, FILE, CLOSEL, STLD.RM, and STORE, generate code as follows:

- When checking for syntax errors is completed, all nonnull parameters after the FIT address are placed in registers.

- Register B6 is set to the end of the macro expansion as the return address.

- A jump to the proper BAM entry point is generated in the top of a word; bits indicating which parameters were specified with the macro are set in the bottom of the word.

- The FIT address is placed in register A0; if it is already in A0, no code is generated.

- Register B1 is set to 1; if B1=1 pseudo-op is in effect no code is generated.

CHECK

The primary use of the CHECK macro is to check the completion status of input/output operations initiated by GETWR or PUTWR. It can also be used to check input/output completion status after any macro is issued. This macro is applicable to sequential and word addressable files. The file is checked for input/output activity. If active, the job is placed in recall until activity ceases; control is returned to the user. If the file has no input/output activity, control is returned to the user. Data and error exits are suppressed, so the user should examine the file position (FP) and error status (ES) fields of the FIT before continuing.

When the CHECK macro is used to ensure completion of a GETWR request, the RL field contains the record length when CHECK is complete. If an S or L tape is being read, the value of the RL field is the actual number of characters in the record. For S type records on other devices, however, the value of the RL field is the record length rounded upward to a multiple of 10.

When the CHECKR macro is used, the status of the input/output activity is checked and control is returned immediately to the user. The job is not put in recall. If input/output activity is complete, control is returned to a location tag; otherwise, control is returned to the user following the CHECKR.

The formats of the CHECK and CHECKR macros are shown in figure 5-1.

CHECK fit

CHECKR tag_1,fit

fit Address of the FIT.

tag_1 Designates the location to receive control when input/output activity is complete.

Parameters can be specified as registers.

Figure 5-1. CHECK and CHECKR Macro Formats

CLOSEM

The CLOSEM macro terminates file processing and positions the file as specified. It should be the last macro issued for a file. The CLOSEM macro is applicable to both file organizations. Format of the CLOSEM macro is shown in figure 5-2.

CLOSEM fit,cf,typ

fit Address of the FIT.

cf Positions the file after close processing:

R	Rewind (default if a FILE close)
N	No rewind
U	Unload (default if a VOLUME close); if a FILE close, release buffer space and remove name from active file list
RET	Return; rewind and unload tape; release buffer space and remove name from active file list
DET	Detach; no rewind; release buffer space and remove name from active file list
DIS	Disconnect; disconnect terminal file and remove name from active file list

typ Type of close to be performed:

FILE	Closes the file; file processing is terminated (default).
VOLUME	Processing on the current volume is terminated, and volumes are switched; the volume number is incremented, and file processing can continue on the new volume without OPENM.

Only the fit parameter can be specified as a register.

Figure 5-2. CLOSEM Macro Format

When the CLOSEM macro is executed for a file open for output, any information in the file buffer is written to the file device as part of file termination. For sequential files on tape, appropriate label processing occurs during close. Refer to section 6, Label Processing, for a complete description of file and volume label processing.

Close processing for a file varies according to the value specified for the cf parameter of the CLOSEM macro, as follows:

- Rewind

  The file is rewound. In a multivolume file, the current volume is rewound.

- No rewind

  The file is not rewound. Physical file positioning is preserved; however, logical file positioning may not be. Logical and physical position coincide when record access stops on a file boundary or RT=S.

- Unload

  The file is rewound. The open/close flag (OC) field of the FIT is cleared. If it is a permanent file, it is disassociated from the job and returned to the permanent file manager. Any unit record file (OUTPUT, PUNCH, or a file that has had ROUTE for NOS or DISPOSE for NOS/BE performed) is disassociated from the job. The disposition of the unit record file is the same as that determined by the operating system UNLOAD control statement. A magnetic tape is unloaded, but the device is not returned to the system. Any scratch mass storage space assigned to the file is released.

- Return

  The processing is the same as for unload, except that for a tape file, the device is returned to the system.

- Detach

  The file is not rewound. The OC field of the FIT is cleared. The file is no longer logically available; however, it has not been physically released.

- Disconnect

  The OC field of the FIT is cleared. The file is disconnected from the terminal. A non-terminal file is not rewound and is no longer logically available.

A CLOSEM request for a file that has never been opened, or a file that has been closed but not unloaded or reopened, has the following effects:

- The FIT error status redundant close is set.

- File positioning is the same as for an open file.

- Control is returned to the error exit.

- No label processing is performed.

If a file is closed and then reopened, FIT verification and FILE control statement processing are not repeated if the CF field is set to R or N. Therefore, FIT fields such as BT, RT, and FO should not be changed when the file is reopened. To have FIT verification and FILE control statement processing repeated, the file must have been closed with the CF field set to U, RET, DET, or DIS.

If a file is closed and then reopened, the second OPENM uses the old values in the FWB and BFS fields; therefore, the buffer allocated could be too small. To ensure allocation of a new buffer whenever a file is reopened in a program, the file must be closed with the CF field set to DET, and the BFS field must be reset.

ENDFILE

The ENDFILE macro writes an end-of-partition on a file opened for output or input/output. It is applicable to sequential file organization only. Format of the ENDFILE macro is shown in figure 5-3.

ENDFILE

fit Address of the FIT or register containing the address.

Figure 5-3. ENDFILE Macro Format

For W type records, the ENDFILE macro writes a control word with an end-of-partition flag, and the current PRU or block is terminated. For S/L devices when record type is not W, the ENDFILE macro terminates the current block and writes a tapemark. For PRU devices when record type is not W, the ENDFILE macro terminates the current system-logical-record with a short PRU level 0, and writes a zero-length PRU level 17_8.

Multiple ENDFILE macros execute as encountered. ENDFILE calls in midrecord are only allowed for files with S type records; for other record types, the end-of-partition is not written and a nonfatal error is issued.

GET

The GET macro retrieves data from a file and delivers it to the working storage area. It is allowed with files opened for input or input/output only. This macro has several forms, which are shown in figure 5-4.

The GET macro transfers a record from a file to the specified working storage area. Lengths are specified and returned in characters. If the number of characters is not a multiple of ten, the remaining characters from RL to the word boundary are undefined. This is applicable to all record types except Z. GET is applicable to both file organizations.

Applicable parameters by type of file organization for GET are:

Sequential	fit,wsa,rl,dx,
Word addressable	fit,wsa,rl,ex,wa

GET fit,wsa,rl, $\left\{\begin{matrix}\text{dx}\\ \text{ex}\end{matrix}\right\}$, wa

GETWR fit,wsa,rl

GETP fit,wsa,ptl,dx,,SKIP

fit Address of the FIT.

wsa Address of the working storage area to which the user record is delivered.

ptl Partial transfer length; number of characters to be transferred.

rl Record length in characters. Required for U type records only.

dx Address of the end-of-data routine.

ex Address of the error routine.

SKIP If the file is positioned at mid-record, advances to the beginning of the next record before transferring data; cannot be used with U type records.

wa Word address on word addressable files where reading is to start. Word addresses begin with 1.

Parameters (except SKIP) can be specified as registers; if parameters are not specified, values in appropriate FIT fields are used (except GETWR where all parameters are required).

Figure 5-4. GET, GETWR, and GETP Macro Formats

The following FIT fields are updated during GET processing:

RL Actual length of the record read is returned. Length is specified in characters. For Z type records, the number of significant characters is returned.

RC Record count is updated each time GET reads a record.

For record types other than U, control information in the record or FIT fields is used to determine record length. If the GET request encounters a record longer than the length specified in the maximum record length (MRL) field in the FIT, an excess data error occurs. The number of characters specified by the MRL field are transferred, the remaining characters are skipped, and control passes to the error exit. A record greater than the value specified by the MRL field is prevented from overwriting a portion of the calling program or other preserved information. Control is passed to the user end-of-data exit by a GET request that detects a section or partition boundary, or the end of the file.

The GETWR macro initiates the transfer of data in units of words, and transfers control to the user. GETWR is intended for use in conjunction with the suppress buffer option. The suppress buffer flag (SBF) field of the FIT can be set by a FILE control statement. If the SBF field of the FIT is set to

YES, the data is transferred directly to the working storage area, not to the buffer. If the SBF field is set to NO, the data is transferred through the buffer to the working storage area. To check for completion of the operation, the CHECK or CHECKR macro must follow.

The GETWR macro is applicable to sequential files only. The working storage area and the record length must be specified. When reading or writing small S type records to or from an S or L tape, it is sometimes advantageous to set the SBF field to NO, thus gaining nonstop input/output at the expense of buffer space.

When the SBF field is set to NO, all applicable FIT parameters must still be supplied for GET or PUT operations. Also, any data or error exits specified for GET or PUT operations are taken if the SBF field is set to NO. If the SBF field is set to YES, no data or error exits are taken.

For an S type record PUTWR/GETWR to and from extended memory on the CDC CYBER 176 Computer System, SBF must be set to NO and a central memory buffer must be provided.

The GETP macro transfers partial records in lengths specified by the ptl parameter; it can be used to transfer an arbitrary amount of data from a record. GETP is applicable to sequential files only. For U type records, a value must be stored in the RL field before the first GETP on the file.

OPENM

Before a file can be read or written, it must be made available by an OPENM macro. Macros that affect the FIT (FILE, STORE, FETCH, and SETFIT) can be used before the OPENM macro. Any file manipulation macro, however, is valid only after the file has been opened. Error procedures are initiated if attempts are made to access an unopened file.

OPENM is applicable to both file organizations. Format of the macro is shown in figure 5-5.

OPENM prepares a file for processing by creating and linking all required system tables for a file, by translating user-supplied parameters into appropriate values in the relevant tables, and by interfacing with label processing. When OPENM is executed, the following events occur:

- FILE control statement processing occurs if it has not been suppressed by SETFIT execution. FILE control statement processing can be initiated by SETFIT prior to OPENM. If so, SETFIT sets the PDF field in the FIT to inhibit reprocessing of the FILE control statement. OPENM execution clears the PDF field.

- The FIT is checked for logical consistency. Conditions investigated are listed in table 5-1. Depending on the file organization, additional checks can be made for required fields and other defaults supplied.

- Buffer parameters are processed.

- A read ahead is performed on sequential files opened for input.

OPENM fit,pd,of

fit Address of the FIT.

pd Specifies type of processing:

INPUT File is opened for read only (default, which applies when file is first opened and if no previous pd value has been set in the FIT)

OUTPUT File is opened for write only

I-O File is opened for read and write

of Open flag; specifies file positioning at open time:

R File is rewound before any other open procedures are performed (default).

N No file positioning is done before other open procedures.

E For sequential files, the file is positioned immediately before the EOI to allow extensions to a mass sotrage file; for permanent sequential files, the user must include an EXTEND control statement if the file is opened with an E position and operations are under NOS/BE.

Only the fit parameter can be specified as a register.

Figure 5-5. OPENM Macro Format

TABLE 5-1. FIT CONSISTENCY CHECKS

Condition	Action
RT=D, LL=0	Error
RT=T, and CL, HL, or TL=0	Error
RT=Z, FL=0	Error
RT=F, FL=0	Error
RT=T, HL not greater than CL+CP	Error
OF=E, file is not mass storage	Error
FO=LB	Error
Invalid BT field	Error
BT=I, RT ≠ W	Error
BT=K, RB=0	Default, RB=1
BT=K, MBL=0	Error
MRL, MBL=0, BT=K, E	Error
BT=K, E, file is not S/L device	Error
BT=K, E, RT=W	Error

- Label processing is initiated if appropriate for a sequential file.

- If no error has been detected, the open/close flag (OC) field in the FIT is set to open and control transfers to the user.

Complete open processing occurs when the first OPENM macro in a job step is issued. If a file is closed and then reopened, FIT verification and FILE control statement processing are not repeated if the close flag (CF) field of the FIT is set to R or N.

Any error detected during open processing sets the error status (ES) field of the FIT. If a user error routine has been specified by the EX field, control passes to that routine. If the user routine corrects the condition that caused the error and executes another open, processing can continue; otherwise, the OC flag reflects 0 (not open) and further file access is prohibited.

Buffer fields are investigated when a file is opened. If the FWB field is zero (no buffer address supplied), an address is allocated. If the BFS field is zero (no buffer size supplied), the minimum space required is calculated and the value is stored in the BFS field. Although BAM sets the buffer pointers in the FIT during OPENM processing, buffer allocation does not actually take place until the first macro requiring a buffer is issued. If the SBF field has been set to YES to suppress buffering, no buffer is allocated.

The timing in relation to specifying file processing parameters and open processing is important. These parameters differ for each file organization. Section 4 lists the requirements for the specific parameters by file organization. The following shows the possible relationships between the OPENM macro and the parameters:

- Certain parameters must be set in the FIT with the FILE macro, FILE control statement, or the STORE macro prior to open time; otherwise, a default value is assumed without comment. These parameters are effective only until another open is executed; attempted changes are ignored without comment or error until another open is executed. At that time, the current values in the FIT are used to accomplish the open.

- Certain parameters need not be set in the FIT until they are required by file processing commands. Once set, they remain in effect until changed.

- Certain parameters have no default and must be set in the FIT to avoid a fatal error prior to use by a file processing command.

An OPENM can follow a FATAL OPENM attempt; FNF and PDF are cleared, permitting FILE control statement processing unless NOFCP is set to YES.

PUT

The PUT macro transfers data from working storage to a file; it is allowed for files opened for output or input/output only. This macro has three forms, which are shown in figure 5-6.

The PUT macro transfers a record from working storage to a file. It is applicable to both file organizations.

PUT fit,wsa,rl,ex,wa

PUTWR fit,wsa,rl

PUTP fit,wsa,ptl,ex,,rl,TERM

fit Address of the FIT.

wsa Address of the working storage area.

rl Number of characters to be written, or for PUTWR the number of words.

ptl Partial transfer length; number of characters to be transferred.

ex Address of the error routine.

wa Word address.

TERM Signals a record is to terminate with this PUTP; used only with W, S, or U type records.

Parameters can be specified as registers; if parameters are not specified, values in appropriate FIT fields are used (except PUTWR where all parameters are required).

Figure 5-6. PUT, PUTWR, and PUTP Macro Formats

Applicable parameters by type of file organization for PUT are:

Sequential	fit,wsa,rl,ex
Word addressable	fit,wsa,rl,ex,wa

The rl parameter need not be specified for files with record types F, Z, T, D, and R. Instead, record length for these formats is determined by BAM using fields in the FIT and the content of the record in the working storage area. The value of the RL field for F, Z, T, D, and R type records is determined as follows:

F Record length is taken from the FL field of the FIT.

Z If the rl parameter is nonzero in the PUT macro or the RL field of the FIT is nonzero, the end of the record is determined by searching backwards from the character position specified by the value of the RL field. If the rl parameter is not supplied and the RL field is zero, the end of the record is determined by searching backwards from the character position specified by the value of the FL field. A zero byte terminator is appended in BAM's buffer from that point. Intervening characters are binary zero filled.

R Record length is determined by scanning the record in the working storage area for the terminating record mark character (RMK) which was specified in the FIT. An error occurs if the record mark is not found within the maximum record length.

T Decimal count is extracted from the record and used to calculate the record length. Count field length (CL), trailer count beginning character position (CP), header length (HL), and trailer length (TL) are obtained from fields in the FIT.

D Decimal character record length is extracted from the record. Length field length (LL) and length field beginning character position (LP) are obtained from fields in the FIT.

In all preceding cases, the transferred record length is stored in the RL field of the FIT at the end of the PUT operation.

The RL field must be specified for U, S, and W type records with PUT requests. Lengths specified by the user for W and S format records exclude the record control word and level number appendage. They are supplied by BAM. S type records on a PRU device are always an integral number of words (multiple of ten 6-bit characters) in storage. The value specified by the RL field is rounded upward, if necessary. A level 0 appendage is recorded for each completed PUT operation for S type records. For S/L tapes, the number of characters specified by the RL field are written as one tape block.

For any word addressable files, the word address (WA) field in the FIT is updated to reflect the next available word address; therefore, such files can be written sequentially.

Any errors during PUT or PUTP processing cause transfer to the error routine if one has been specified. In the case of excess or insufficient data errors, no data has been transferred. In the case of other errors, data is unreliable.

The PUTP macro is used to create a single record from a series of write requests. It transfers partial records in lengths specified by the ptl parameter. It can be used to transfer an arbitrary amount of data to a record. By changing the wsa parameter from call to call, portions of the same record can be transferred from different parts of central memory. The PUTP macro is applicable to sequential files only. It is not allowed for R type records.

The PTL field indicates the number of characters to be transferred from the working storage area to the record under construction. The PTL field of the FIT is used for any PUTP operation not containing a ptl parameter value in the macro.

The PTL field must be set for the PUTP macro that initiates a new record. If the record length is specified, it becomes the maximum number of characters possible in the record and is used to determine an excess data error condition. If a PUTP request supplies data that would exceed the record length, or if any other macro requests file action prior to completion of the record, a fatal error condition occurs.

The termination of a record being constructed by a series of PUTP operations is recognized by the total record length (RL) set by the first PUTP macro specified, or by the presence of the TERM parameter to signify the last partial write for this record. For S, U, and W type records using the PUTP macro, the RL field can be set to zero and the TERM parameter used.

The user can make a WEOR request to signal the end of an S type record created by a sequence of PUTP requests. The level number specified by the WEOR macro can be 0 through 16_8; only level 0 should be written. Levels 1 through 16_8 exist to support downward compatability in certain system programs. The ENDFILE, REWINDM, CLOSEM, SKIPB, WTMK, and PUT macros also cause termination of a record (block) by adding a level 0. If Z type records are written by the PUTP macro, trailing blanks are suppressed only with the record portion of the last partial transfer.

The PUTWR macro initiates the transfer of data in units of words, and then transfers control to the user. Because the operation might not be complete, the CHECK or CHECKR macro must follow. The PUTWR macro is valid only for sequential files and is intended for use in conjuction with the suppress buffering option. Refer to the GETWR discussion in this section. The working storage area and the record length must be specified with the PUTWR macro.

For an S type record PUTWR/GETWR to and from extended memory on the CDC CYBER 176 Computer System, SBF must be set to NO and a central memory buffer must be provided.

REPLACE

The REPLACE macro replaces the last record read with a record from the working storage area. It is applicable to sequential mass storage files with C type blocks and F or W type records. Format of the REPLACE macro is shown in figure 5-7.

REPLACE fit,wsa,,ex

fit	Address of the FIT.
wsa	Address of the working storage area with the new record.
ex	Address of the error routine.

All parameters can be specified as registers.

Figure 5-7. REPLACE Macro Format

Replacement records must be the same size as the record replaced. If the last operation was not a GET or the file is not positioned at EOR, a trivial error results and the request is ignored.

REWINDM

The REWINDM macro positions an unlabeled or nonstandard labeled tape file to the beginning of the current volume. A mass storage file or labeled tape is rewound to beginning-of-information. It is applicable to both file organizations. Format of the REWINDM macro is shown in figure 5-8.

REWINDM fit

fit	Address of the FIT or register containing the address.

Figure 5-8. REWINDM Macro Format

The file need not be open when the REWINDM macro is issued. If the last operation was a write, buffers are cleared and end-of-information written before a file is rewound.

SKIPdu

The SKIPdu macro repositions a file in a forward or backward direction. It is applicable to sequential files only. Format of the SKIPdu macro is shown in figure 5-9. SKIPBL is not supported for T, R, U, and D type records or K and E type blocks; SKIPFL is not supported for U type records.

SKIPdu fit,count

d	Direction of skip:

F	Forward

B	Backward

u	Units to be skipped:

L	Logical records

P	Physical records or system-logical-records of level 0

F	Tapemark or level 17_8 on PRU devices

fit	Address of the FIT.

count	Number of units to be skipped. A null parameter results in a zero count.

The count and fit parameters can be specified as registers.

Figure 5-9. SKIP Macro Format

The SKIPdu macro checks user parameters, reads from the assigned device, positions according to the specified unit to be skipped, and returns control to the user. The SKIPdu macro does not return a record to the working storage area. If a boundary condition is detected before the skip count is exhausted, control is transferred to the end-of-data routine with the appropriate file position set.

A SKIPdu macro call transfers control to the end-of-data routine under the following conditions:

- SKIPFL encounters end-of-information
- SKIPFL or SKIPBL encounters end-of-partition
- SKIPFP or SKIPBP encounters level 17_8 or a tapemark
- SKIPFL or SKIPBL encounters end-of-section
- SKIPBL encounters beginning-of-volume
- SKIPBu detects the load point on a tape file
- SKIPFF or SKIPFP encounters end-of-information

SKIPdP, SKIPdF, and SKIPBL do not detect parity errors. SKIPFL does detect parity errors. A negative skip count is not allowed; the request is ignored, and an error is issued.

If a file is positioned at midrecord when a SKIPdu macro is issued, processing is as follows:

SKIPFu,fit,0	The file is positioned forward to the unit boundary.
SKIPBu,fit,0	The file is positioned backward to the unit boundary.
SKIPFu,fit,n	The file is positioned forward to the unit boundary and then forward n units.
SKIPBu,fit,n	The file is positioned forward to the unit boundary and then backward n units.

An output file can be positioned backward only. If the previous operation was a PUT, the file is terminated before reverse motion is initiated.

WEOR

The WEOR macro is used to terminate a section. The macro format is shown in figure 5-10. WEOR writes an end-of-section for sequential files, if applicable, as shown in table 5-2.

WEOR fit,lvl

fit	Address of the FIT.

lvl	Level number to be appended. The lvl parameter is set to 00 by default; lvl can be any octal value from 00 to 16_8. See explanation below.

All parameters can be specified as registers.

Figure 5-10. WEOR Macro Format

For S type records, a read of EOS returns an EOR value to the file position (FP) field; the EOS value is never returned. For K, E, or C type blocks on an S/L device, an EOS cannot be detected by the GET macro.

For S type records, the WEOR macro can be used to terminate the system-logical-record being constructed by a series of PUTP macros. The WEOR macro terminates the current record and appends a level number. The level number specified by the WEOR macro can be 0 through 16_8; only level 0 should be written. Levels 1 through 16_8 exist to support downward compatability in certain system programs.

TABLE 5-2. WEOR PROCESSING

Device	End-of-Section		Boundary Written
	Block Type	Record Type	
PRU device	I	W	One-word record pointing back to the last I block boundary. Control word with EOS flags; terminate the block with level 0.
	C	W	Control word with EOS flags; terminate the block with level 0.
	C	All but W	Terminate the block with level not greater than 16_8.
S/L tape	I	W	Zero-length deleted records to exceed noise record size; one-word record pointing back to the I block boundary; control word with EOS flags. Terminate the block.
	C	W	Zero-length deleted records to exceed noise record size. Control word with EOS flags. Terminate the block.
	C,K,E	All but W	Terminate the block.

For W type records, the file must be on a record boundary to write an end-of-section control word. The record count is updated. The WEOR macro writes a deleted, zero-length record with the flag bit set.

WTMK

The WTMK macro is provided to record a tapemark, or level 17_8, in nonstandard label processing. It is applicable to sequential file organizations only. Format of the WTMK macro is shown in figure 5-11.

```
WTMK fit

fit      Address of the FIT or register containing the
         address.
```

Figure 5-11. WTMK Macro Format

The WTMK macro does not flush the buffer. It checks user parameters, terminates the current block, and records the tapemark on S/L tapes or a level 17_8 for files residing on PRU devices. Control is then returned to the user. The block number (BN) field of the FIT is not cleared to zero.

LABEL PROCESSING 6

Tape label processing takes place when a sequential file on magnetic tape is opened or closed. File labeling conventions facilitate the exchange of magnetic tapes between installations. Recording a file using any labeling convention has meaning only for sequential files. The tape formats supported under the NOS operating system are: SI binary, I, and S/L. The tape formats supported under the NOS/BE operating system are: SI coded and binary and S/L.

The conversion mode (CM) FIT field must be set to YES for coded tape formats. In coded mode, the operating system converts 6-bit display code to and from coded characters when reading and writing. In binary mode (CM=NO), the operating system reads and writes 6-bit display code. Refer to the appropriate operating system reference manual for a complete discussion of tape processing.

LABEL DEFINITIONS

The three basic classes of labeling conventions are standard labeled files, nonstandard labeled files, and unlabeled files.

STANDARD LABEL

A standard labeled file is recorded with label groups appended to the data. The content of the labels and the format of the file so recorded conform to the American National Standards X3.27-1969, Magnetic Tape Labels for Information Interchange. Standard label processing applies only to sequential files on magnetic tape.

A label group is composed of a number of 80-character blocks separated by an interrecord gap. Labels appear in memory as display code characters; they are recorded on the file in the character set specified by the coversion mode parameter on the LABEL control statement. The label group is separated from the data records in the file by a hardware tapemark. The three types of label groups are volume/header group, end-of-file group, and end-of-volume group. The position of these groups in relation to file data is shown in figure 6-1. Labels are represented by the four characters of their identifiers and numbers. Table 6-1 shows the contents of each label defined by ANSI.

NONSTANDARD LABEL

A nonstandard label is a descriptive record appended to data according to a set of rules other than the ANSI standard convention. BAM allows nonstandard labels to be written for processing by the user for sequential files on all devices.

UNLABELED

An unlabeled file has no system descriptive records at the beginning of the file. The first block of the file is treated as a data block. An unlabeled file on an SI or I tape has a system-processed trailer label. The presence of this label allows end-of-information to be defined. Multivolume processing is done automatically by the operating system for SI or I tapes.

NOTE

> On an S/L tape, no system trailer label exists and end-of-information is undefined.

On input, a tapemark encountered after the end-of-tape reflective spot signals end-of-volume and the operating system switches volumes. The formats of unlabeled magnetic tape files are shown in figure 6-2.

LABEL PROCESSING FIT FIELDS

The following FIT fields are used during label processing:

LT Label type. LT is determined when the file is opened, based on parameters on the FILE macro or control statement. If label type is unspecified, user label processing is not allowed.

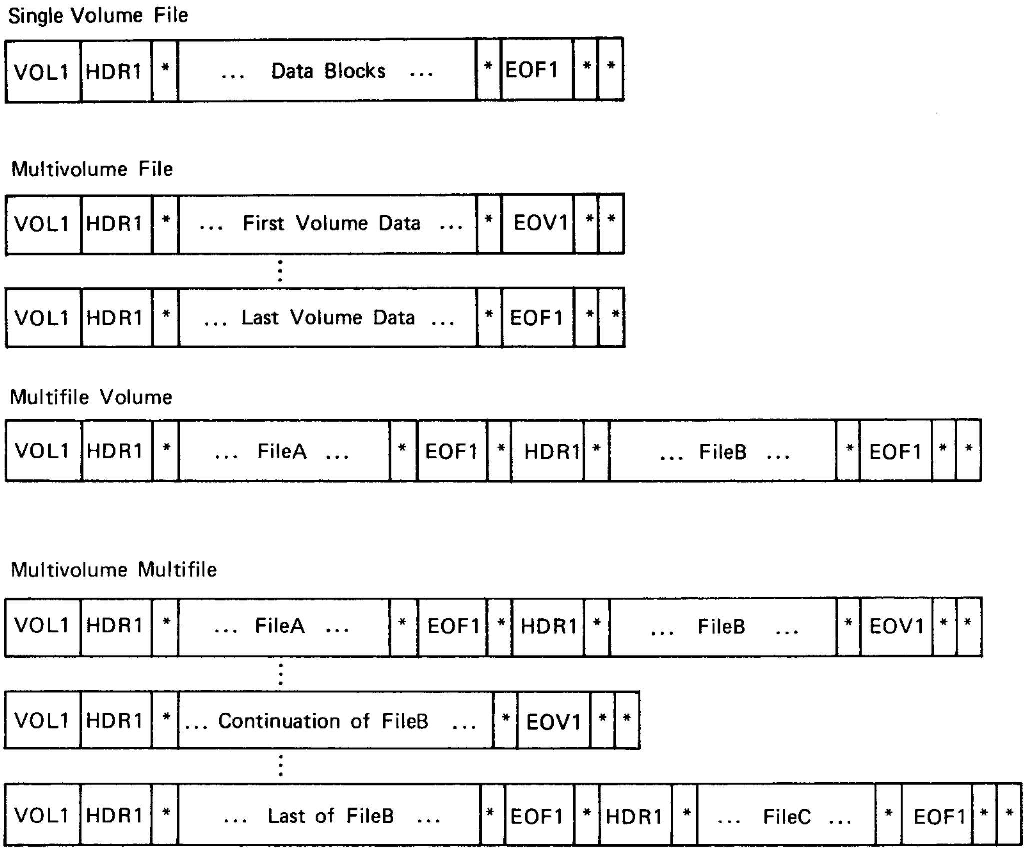
LT=S	Standard
LT=UL	Unlabeled
LT=NS	Nonstandard
LT=ANY	Unspecified

LCR Label creation flag. LCR is determined when the file is opened, based on the parameters of the FILE macro or control statement.

LCR=CHK	Existing label is read and checked
LCR=CRT	New label is written

LA Label area address. Labels are delivered into the area as a result of the GETL macro. Labels are fetched and submitted for processing as a result of the PUTL macro. If LA is zero, no user label processing can be done.

LBL Label area length in characters.


Single Volume File
VOL1 | HDR1 | * | ... Data Blocks ... | * | EOF1 | * | *
Multivolume File
VOL1 | HDR1 | * | ... First Volume Data ... | * | EOV1 | * | *
VOL1 | HDR1 | * | ... Last Volume Data ... | * | EOF1 | * | *
Multifile Volume
VOL1 | HDR1 | * | ... FileA ... | * | EOF1 | * | HDR1 | * | ... FileB ... | * | EOF1 | * | *
Multivolume Multifile
VOL1 | HDR1 | * | ... FileA ... | * | EOF1 | * | HDR1 | * | ... FileB ... | * | EOV1 | * | *
VOL1 | HDR1 | * | ... Continuation of FileB ... | * | EOV1 | * | *
VOL1 | HDR1 | * | ... Last of FileB ... | * | EOF1 | * | HDR1 | * | ... FileC ... | * | EOF1 | * | *

* Tapemarks written by system hardware; user does not control.

Figure 6-1. Standard Label Tape Formats

TABLE 6-1. ANSI STANDARD LABELS

Label	Character Position	Field	ANSI Name (System Name)	Length	Contents in Memory	Default Written	Checked On Input
Volume Header	1-3	1	Label Identifier	3	VOL	VOL	Yes
	4	2	Label Number	1	1	1	Yes
	5-10	3	Volume Serial Number	6	Any 6-bit display code ASCII characters	As typed from console	Yes if file assigned by volume serial number
	11	4	Accessibility	1	Space	Space	No
	12-31	5	Reserved	20	Spaces	Spaces	No
	32-37	6	Reserved	6	Spaces	Spaces	No
	38-51	7	Owner ID	14	Any 6-bit display code ASCII characters	Spaces	No
	52-79	8	Reserved	28	Spaces	Spaces	No
	80	9	Label Standard Level	1	1	1	No

TABLE 6-1. ANSI STANDARD LABELS (Cont)

Label	Character Position	Field	ANSI Name (System Name)	Length	Contents in Memory	Default Written	Checked On Input
First File Header	1-3	1	Label Identifier	3	HDR	HDR	Yes
	4	2	Label Number	1	1	1	Yes
	5-21	3	File Identifier (File Label Name)	17	Any 6-bit display code ASCII characters	Spaces	Yes
	22-27	4	Set Identification (Multifile Set Name)	6	Any 6-bit display code ASCII characters	Volume serial number of first reel of the set	No
	28-31	5	File Section Number (Reel Number)	4	4 digits indicating number of volume in the file	0001	Yes
	32-35	6	File Sequence Number (Position Number)	4	4 digits indicating number of file in multifile set	0001	Yes
	36-39	7	Generation Number (Used only by NOS)	4	4 digits indicating the generation number of the file	0001	Yes
	40-41	8	Generation Version Number (Edition Number)	2	2 digits indicating the edition of the file	00	Yes
	42-47	9	Creation Date	6	Space followed by 2 digits for year, 3 digits for day	Current date is used	Yes
	48-53	10	Expiration Date	6	Same as field 9	Same as field 9	Yes
	54	11	Accessibility	1	Any 6-bit display code ASCII characters	Space	No
	55-60	12	Block Count	6	Zeros	Zeros	Yes
	61-73	13	System Code	13	Any 6-bit display code ASCII characters	Spaces	No
	74-80	14	Reserved	7	Spaces	Spaces	No
Additional File Header	1-3	1	Label Identifier	3	HDR	HDR	Yes
	4	2	Label Number	1	2-9	2-9	Yes
(Reserved for operating system use)		All other fields are not checked on input.					
First End-of-File	1-3	1	Label Identifier	3	EOF	EOF	Yes
	4	2	Label Number	1	1	1	Yes
	5-54	3-11	Same as corresponding HDR1 label fields				
	55-60	12	Block Count	6	6 digits indicating number of data blocks since the last HDR label group		Yes

TABLE 6-1. ANSI STANDARD LABELS (Cont)

Label	Character Position	Field	ANSI Name (System Name)	Length	Contents in Memory	Default Written	Checked On Input
First End-of-File (Cont)	61-80	13-14	Same as corresponding HDR1 label fields				
Additional End-of-File	1-3	1	Label Identifier	3	EOF	EOF	Yes
	4	2	Label Number	1	2-9	2-9	Yes
(Reserved for operating system use)		All other fields are not checked on input.					
First End-of-Volume	1-3	1	Label Identifier	3	EOV	EOV	Yes
	4	2	Label Number	1	1	1	Yes
		All other fields are identical to EOF1 label.					
Additional End-of-Volume	1-3	1	Label Identifier	3	EOV	EOV	Yes
	4	2	Label Number	1	2-9	2-9	Yes
(Reserved for operating system use)		All other fields are not checked on input.					
USER	1-3	1	Label Identifier	3	3-letter code: UVL, UHL, or UTL		Yes
	4-80	Any characters. Content of these fields is not checked on input; it is written as received from the user.					

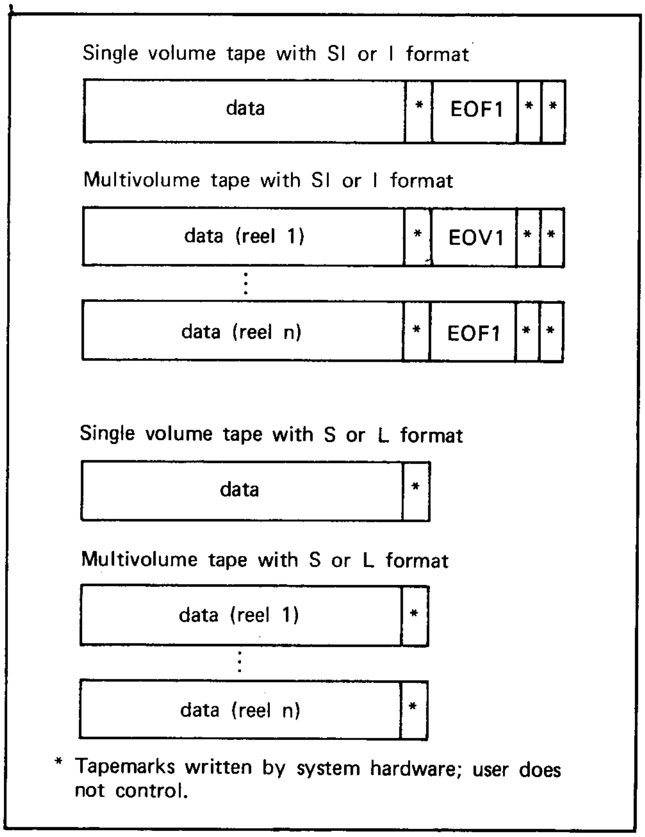

Single volume tape with SI or I format
data
*
EOF1
*
*
Multivolume tape with SI or I format
data (reel 1)
*
EOV1
*
*
data (reel n)
*
EOF1
*
*
Single volume tape with S or L format
data
*
Multivolume tape with S or L format
data (reel 1)
*
data (reel n)
*
* Tapemarks written by system hardware; user does not control.

Figure 6-2. Unlabeled Tape Format

LX Label routine exit address. Control is passed to the LX routine for user label processing at certain file positions, depending on the contents of the ULP field.

ULP User label processing. Types of label processing that are available for standard labels. Any specification for ULP except NO is acceptable. For nonstandard labels when ULP is not set to NO, the user must process all labels.

ULP=NO	No user label processing
ULP=V	User volume label processing
ULP=F	User file label processing
ULP=U	User label processing (UHL, UTL, UVL)
ULP=VF	Combination of V and F
ULP=VU	Combination of V and U
ULP=FU	Combination of F and U
ULP=VFU	Combination of V, F, and U

DECLARING LABEL TYPE

Before a file is opened, the user must set the label type (LT) and label creation flag (LCR) fields of the FIT with a FILE macro or control statement. The equivalent information must be specified on a LABEL or REQUEST control statement. Refer to the appropriate operating system reference manual for a complete description of these control statements. Under the NOS operating system, a file with nonstandard label type must be declared unlabeled on the REQUEST or LABEL statement.

A standard labeled multifile set is specified with the MF parameter on the REQUEST control statement for NOS/BE and with the SI parameter on the LABEL control statement for NOS. In addition, the multifile set name (MFN) field in the FIT must be set with the FILE control statement or STORE macro. The format of a multifile set is explicitly specified by the ANSI standard. User label processing of any of the label groups is supported as described in the User Label Processing Macros subsection. A multifile set can be created and read only if the user supplies all the labels.

Tapes with earlier standard labels, Z labels, can be read but not written. They must be identified by the Z parameter on the REQUEST or LABEL control statement for the operating system to read these files. Under the current ANSI standard, density of label data is the same as that of subsequent data. Earlier standards allowed data recording density to be specified by character 12 of the VOL1 label.

STANDARD LABEL PROCESSING

The VOL1, HDR1, EOV1, and EOF1 labels are always processed to ensure adherence to ANSI standards and LABEL statement parameters. VOL1, EOF1, and EOV1 are written by the system with default values. Any user values are ignored without comment. A tapemark is written to terminate label groups on standard labeled output tapes.

User processing is allowed on ANSI-defined labels when label type (LT) is declared standard and the user label processing (ULP) field is set to identify a label group. Working knowledge of standard magnetic tape label structures is necessary for the following label processing discussions. The notations UTL(f) and UTL(v) indicate all the user trailer (UTL) labels that can follow EOF and EOV labels. UHL(a) and UVL(a) indicate all the user header (UHL) or user volume (UVL) labels that can follow the HDR and VOL1 labels.

A file is initialized by the operating system when the LABEL control statement is encountered. Consequently, user label processing should not be attempted when a LABEL control statement is used. If user label processing is desired, the REQUEST control statement must be used instead of LABEL.

User label processing is controlled by the FIT fields LX, LA, and ULP. Control passes to a user label processing exit if the user has specified user label processing (ULP on the FILE control statement or macro), and the position of the standard labeled file is such that there are labels to be processed. User label processing (ULP) can be specified for some label groups and not others. If LX is zero, the operating system supplies the requisite label.

Label processing capabilities are provided by the GETL, PUTL, and CLOSEL macros. GETL retrieves the next label of a label string and delivers it to the label area. Labels are written by PUTL. CLOSEL terminates label processing. More detail is given under User Label Processing Macros.

INPUT TAPE USER PROCESSING

Existing standard labels can be checked when the processing direction (PD) field of the FIT is set to INPUT, or I-O with the LCR field set to CHK. Labels can be retrieved with GETL; each GETL returns an 80-character label to LA. If the value of the LBL field is less than 80, LBL characters are retrieved and an error flag is set. If LBL exceeds 80, an error flag is set and 80 characters are delivered to LA.

OPENM of Input Tape

If the value of the LX field is zero, the header label group is processed automatically by the operating system. If the LX field is nonzero, control is given to the user's routine twice during OPENM processing. At the first exit, the open/close flag (OC) field of the FIT is set to not open and the PUTL macro can be issued to have the system perform a label check for header labels that are checked on input as indicated in table 6-1. At the second exit, the OC field is set to open and the GETL macro can be issued.

The labels that can be retrieved during OPENM processing are shown in table 6-2 in the order in which successive GETL macros would retrieve them. Label processing is not allowed for an OPENM with no rewind.

TABLE 6-2. INPUT FILE LABELS ACCESSED AT OPENM

ULP	Labels Retrieved by GETL
V	VOL1
F	HDR1-9
U	UVL(a), UHL(a)
VF	VOL1, HDR1-9
VU	VOL1, UVL(a), UHL(a)
FU	HDR1-9, UHL(a), UVL(a)
VFU	VOL1, UVL(a), HDR1-9, UHL(a)

CLOSEM of Input Tape File

If the LX field is zero, EOF1 is processed automatically by the operating system. If LX is nonzero, control is passed to the user label processing routine twice. At the first exit, the OC field of the FIT is set to open and the PUTL macro can be issued to have the system perform a label check for trailer labels that are checked on input as indicated in table 6-1. At the second exit, the OC field is set to closed, and trailer labels can be retrieved with the GETL macro. The labels available depend on the contents of the ULP field as shown in table 6-3.

TABLE 6-3. INPUT FILE LABELS ACCESSED AT CLOSEM

ULP	Labels Retrieved by GETL
V	None
F	EOF1-9
U	UTL(f)
VF	EOF1-9
VU	UTL(f)
FU	EOF1-9,UTL(f)
VFU	EOF1-9, UTL(f)

CLOSEM of Input Tape Volume

If the LX field is zero, EOV1 is processed automatically by the operating system. Volumes are switched and header labels are processed automatically. If the LX field is nonzero, control is passed to the user at address LX when the file position (FP) is beginning-of-volume (BOV) because CLOSEM VOLUME could have been issued midreel.

When an EOV occurs because the label group indicating end-of-tape has been reached by a GET or SKIPFL macro, control is transferred to the LX address at EOV and BOV. The user must differentiate between these file positions in the label routine of the program.

When the file position is EOV, the labels that can be retrieved by the GETL macro are those listed in table 6-4. When the file position is BOV, the labels that can be retrieved by the GETL macro are those listed in table 6-5.

TABLE 6-4. INPUT FILE LABELS ACCESSED AT CLOSEM VOLUME (EOV)

ULP	Labels Retrieved by GETL
V	EOV1-9
F	None
U	UTL(v)
VF	EOV1-9
VU	EOV1-9, UTL(v)
FU	UTL(v)
VFU	EOV1-9, UTL(v)

TABLE 6-5. INPUT FILE LABELS ACCESSED AT CLOSEM VOLUME (BOV)

ULP	Labels Retrieved by GETL
V	None
F	HDR1-9
U	UVL(a), UHL(a)
VF	HDR1-9
VU	VOL1, UVL(a), UHL(a)
FU	HDR1-9, UHL(a), UVL(a)
VFU	VOL1, UVL(a), HDR1-9, UHL(a)

OUTPUT TAPE USER PROCESSING

A new standard label can be written when the processing direction (PD) field is set to OUTPUT or when the PD field is set to I-O and the LCR field is set to CRT.

In a user label routine, labels can be written with the PUTL macro. Each PUTL takes one 80-character label from address LA and writes it to the file. The 80 characters must be correctly formatted or an error results. If the length of the label area or the label address is zero, no user labels are written, but default VOL1, HDR1, EOF1, or EOV1 label is supplied and an error is returned.

VOL1, HDR1, EOF1, and EOV1 are ANSI-required labels and the operating system ensures that these labels are written to the file. All header labels must be supplied at open time; all trailer labels must be supplied at close time. The user can supply the header or trailer labels in any order because the operating system reorders them. User label processing is allowed on output only if the OPENM macro with the rewind option is used.

OPENM of Output Tape

If the LX field is zero, default VOL1 and HDR1 labels are supplied automatically. If the LX field is nonzero, control is passed to the user label processing routine. Labels that can be written are indicated in table 6-6. Each PUTL macro writes one label.

TABLE 6-6. OUTPUT FILE LABELS WRITTEN AT OPENM

ULP	Labels Written by PUTL
V	None
F	HDR2-9[†]
U	UVL(v), UHL(f)
VF	HDR2-9[†]
VU	UVL(v), UHL(f)
FU	HDR2-9[†], UHL(f), UVL(v)
VFU	UVL(v), HDR2-9[†], UHL(f)

[†] These labels are reserved for operating system use.

CLOSEM of Output Tape File

If the LX field is zero, a default EOF1 is supplied automatically, and any user EOF1 is ignored. If the LX field is nonzero, control is passed to the user label processing routine. Labels that can be written are shown in table 6-7.

TABLE 6-7. OUTPUT FILE LABELS WRITTEN AT CLOSEM

ULP	Labels Written by PUTL
V	None
F	EOF2-9†
U	UTL(f)
VF	EOF2-9†
VU	UTL(f)
VFU	EOF2-9†, UTL(f)

†These labels are reserved for operating system use.

CLOSEM of Output Tape Volume

If the LX field is zero, default EOV1, VOL1, and HDR1 labels are supplied. If the LX field is nonzero, control is passed to address LX when the file position is EOV and BOV. In either case volume switching is automatic.

When the file position is EOV, the labels that can be written with the PUTL macro are those listed in table 6-8. EOV1 is always supplied by the operating system because its content must be an image of HDR1. If a user issues a PUTL macro for an EOV1, it is ignored. When the file position is BOV, the labels that can be written with the PUTL macro are those listed in table 6-9.

TABLE 6-8. OUTPUT FILE LABELS WRITTEN AT CLOSEM VOLUME (EOV)

ULP	Labels Written by PUTL
V	EOV2-9†
F	None
U	UTL(v)
VF	EOV2-9†
VU	UTL(v), EOV2-9†
VFU	EOV2-9†, UTL(v)

†These labels are reserved for operating system use.

TABLE 6-9. OUTPUT FILE LABELS WRITTEN AT CLOSEM VOLUME (BOV)

ULP	Labels Written by PUTL
V	None
F	HDR2-9†
U	UVL(a), UHL(a)
VF	HDR2-9†
VU	UVL(a), UHL(a)
FU	HDR2-9†, UHL(a), UVL(a)
VFU	UVL(a), HDR2-9†, UHL(a), EOV2-9†

† These labels are reserved for operating system use.

NONSTANDARD LABEL PROCESSING

Nonstandard label processing is entirely the responsibility of the user. This type of label processing is available for sequential files on all devices.

The nonstandard labels can be header and/or trailer labels. Header labels appear between the beginning-of-information and a user-defined point and end-of-volume or end-of-information. The delimiting and processing of nonstandard labels is the user's responsibility.

INPUT FILE USER PROCESSING

Each GETL macro retrieves the number of characters of data specified by the label area length (LBL) field, or fewer characters, from a physical record and delivers them to address LA. If a tapemark or level 17_8 is reached, the GETL macro returns with an end-of-labels file position and no data is transferred.

For an input file, control is passed to the label processing routine during OPENM processing or CLOSEM processing when the file is positioned at end-of-partition. The nonstandard label can then be retrieved with the GETL macro.

During CLOSEM processing of an input volume, control is passed to the user at address LX. The CLOSEM macro should be called when the user has determined that end-of-volume processing is required. File position should be end-of-section or end-of-partition, and the labels should be separated from data. If an end-of-data is encountered during forward reading, control is passed to the end-of-data exit (DX) routine if present. End-of-volume labels must be processed in the end-of-data routine before CLOSEM is called. The user has the option of issuing a CLOSEM VOLUME/FILE at this time. If CLOSEM VOLUME is issued, volumes are switched automatically and control is passed to the user at address LX at load-point.

If the system closes an input volume, automatic volume switching takes place only at the first tapemark after the reflective spot. Control is passed to address LX at BOV for label processing.

OUTPUT FILE USER PROCESSING

Each PUTL macro delivers the number of characters specified by the LBL field from the label area (LA) to the input/output device. The device is formatted as one physical record. The user can use the WTMK macro for writing record delimiters. Delimiters are not required; processing is entirely up to the user. When the system closes the volume because the reflective spot has been encountered, the output buffer is not flushed. If the user closes the volume, the buffer is flushed before any label processing.

For an output file, control is passed to address LX during OPENM and CLOSEM processing. The user can then write labels with the PUTL macro.

For an output volume, control is passed to address LX twice during CLOSEM processing. The first time is for creation of trailer labels, and the second time is for creation of header labels.

Automatic volume swapping occurs after the tape reflective spot is encountered. In this case, label processing is available only at BOV.

USER LABEL PROCESSING MACROS

Macros provide label processing capabilities. The macros provided retrieve labels (GETL), submit labels for writing or checking (PUTL), and terminate user label processing (CLOSEL). They are applicable only to sequential files.

GETL

The GETL macro retrieves the next label of a label group and delivers it to the label area. Format of the macro is shown in figure 6-3.

```
GETL fit,la,lbl
```

fit Address of the FIT.

la Address of the label area; holds the label fetched by the GETL macro.

lbl Length (in characters) of the label area.

All parameters can be specified as registers.

Figure 6-3. GETL Macro Format

During OPENM and CLOSEM processing, entry is made into the label routine and labels appropriate to the current file position are made available to the user via the GETL macro. The GETL macro validates the contents of certain FIT fields and ensures the legality of the call. The file organization (FO) field must be set to sequential (SQ). The label type (LT) field must be set to standard (S) or to nonstandard (NS) with the ULP field set to other than NO. The processing direction (PD) field must be set to INPUT or to I-O with the LCR field set to CHK. A check is made that the LBL field is nonzero, and that the label area is specified. If the labels are standard, the file must be a tape file and the user label processing flags must be set (the ULP field not set to NO).

If labels are standard, the number of characters specified by the LBL field are moved to the user label area at LA. If the LA field has not been set either previously or by the GETL macro, an error exit is taken. If the number of characters specified by the LBL field is greater than 80, only 80 characters are retrieved. If the LX field is zero, no label processing routine exists. The ULP field is used in conjunction with the file position (FP) field to determine what type of label is to be retrieved.

When the GETL macro is issued for standard labels and no errors are detected, the user label processing flags are checked to determine what types of labels are appropriate. The next appropriate label is moved to the label area at LA. If none exists, the end-of-labels flag is set and control is returned to the user label routine. If the value of the LBL field is greater than 80, only 80 characters are moved to the label area at LA. If the LBL field is less than 80, only the number of characters specified by the LBL field are moved to the label area. Labels are retrieved in sequential order. For example, at beginning-of-information with the ULP field set to F, the labels on a file containing HDR1, HDR2, and HDR3 labels would be available in the order HDR1, HDR2, HDR3. Each call to GETL would retrieve only one label.

When the GETL macro is issued for nonstandard labels and no errors are detected, a physical record is read and the number of characters specified by the LBL field are moved to the label area at LA. If the physical record is larger than the LBL field, only the number of characters specified by the LBL field are moved. If the physical record is smaller than the LBL field, as many characters as possible are moved and the number of characters moved are returned in the LBL field.

PUTL

The PUTL macro writes a label. Format of the macro is shown in figure 6-4.

```
PUTL fit,la,lbl
```

fit Address of the FIT.

la Address of the label area; contains the label to be written on the file.

lbl Length (in characters) of the label area.

All parameters can be specified as registers.

Figure 6-4. PUTL Macro Format

During OPENM and CLOSEM processing, entry is made into the label routine. At this time, labels appropriate to the current file position are submitted to be written on an output file. The PUTL macro validates the contents of certain FIT fields to ensure the legality of the call. The file organization (FO) field must be set to sequential (SQ). The label type (LT) field must be set to standard (S) or to nonstandard (NS) with the ULP field set to other than NO. The processing direction (PD) field must be set to OUTPUT or to I-O with the LCR field set to CRT. Additional checks are made that the LBL field is nonzero, and that a label area is specified. If the labels are standard, the file must be a tape file and the user label processing flags must be set to other than NO.

The ULP field is used in conjunction with the file position (FP) field to determine if the label being submitted is legal at the present file position. The first three characters of the label at LA are used to determine the type of label: VOL, HDR, EOV, or EOF. If the LA field has not been set either previously or by the PUTL macro, an error exit is taken.

If labels are standard and no errors have been detected, each call to the PUTL macro examines the label at LA, keying on the first four characters. The ULP field is checked to see if the submission of the label at the current file position is allowed. At beginning-of-information with the ULP flag set to F, submission of a VOL 1 label would not be allowed.

If labels are nonstandard and no errors have been detected, the number of characters specified by the LBL field are taken from the label at LA and written to the file as a physical record.

The PUTL macro can be used on an input type file with standard labels to have the system perform a label check. The LT field must be set to S and the PD field set to INPUT or to I-O with the LCR field set to CHK. At the first label exit taken during OPENM and CLOSEM processing, a PUTL macro can be issued. This causes the labels to be moved from the label area LA to the label buffer. The system then compares this label to the input file label. If they are unlike, the file cannot be opened and a fatal error occurs on NOS; on NOS/BE, the operating system issues an error diagnostic and asks the operator for a GO or DROP decision.

CLOSEL

The CLOSEL macro terminates label processing and returns control to OPENM or CLOSEM processing. CLOSEL must be called to terminate user label processing because it is the only way for the user to return control to BAM. Format of the CLOSEL macro is shown in figure 6-5.

CLOSEL fit

fit Address of the FIT or register containing the address.

Figure 6-5. CLOSEL Macro Format

The CLOSEL macro is used to exit a label processing routine and to return to the calling routine for continued processing. Entry into the label processing routine is made at various times during OPENM and CLOSEM processing. When entry is made, the address of the active FIT is available in register A0. Generally, on input type files (the PD field is set to INPUT or to I-O with the LCR field set to CHK), entry is made when the labels are made available for checking. On output type files, entry is made to the label processing routine to allow the user to submit labels to be written on the file.

On nonstandard end-of-volume and end-of-file labels, label processing must be performed by the user at the end-of-data exit (DX) address. This exit is taken at the tapemark before the nonstandard label. In this case, the CLOSEL macro returns control in-line after the CLOSEL macro.

STANDARD CHARACTER SETS A

CONTROL DATA operating systems offer the following variations of a basic character set:

- CDC 64-character set
- CDC 63-character set
- ASCII 64-character set
- ASCII 63-character set

Table A-1 shows these character sets. The set in use at a particular installation was specified when the operating system was installed.

Depending on another installation option, the system assumes an input deck has been punched either in 026 or in 029 mode (regardless of the character set in use).

Under NOS/BE, the alternate mode can be specified by a 26 or 29 punched in columns 79 and 80 of the job statement or any 7/8/9 card. The specified mode remains in effect through the end of the job unless it is reset by specification of the alternate mode on a subsequent 7/8/9 card.

Under NOS, the alternate mode can be specified by a 26 or 29 punched in columns 79 and 80 of any 6/7/9 card, as described for a 7/8/9 card. In addition, 026 mode can be specified by a card with 5/7/9 multipunched in column 1, and 029 mode can be specified by a card with 5/7/9 multipunched in column 1 and a 9 punched in column 2.

Graphic character representation appearing at a terminal or printer depends on the installation character set and the terminal type. Characters shown in the CDC Graphic column of table A-1 are applicable to BCD terminals; ASCII graphic characters are applicable to ASCII-CRT and ASCII-TTY terminals.

Several graphics are not common for all codes. Where these differences in graphics appear, assignment of collation positions and translation between codes must be made. Tables A-2 and A-3 show the CDC and ASCII character set collating sequences.

TABLE A-1. STANDARD CHARACTER SETS

Display Code (octal)	CDC			ASCII		
	Graphic	Hollerith Punch (026)	External BCD Code	Graphic Subset	Punch (029)	Code (octal)
00[†]	: (colon)[††]	8-2	00	: (colon)[††]	8-2	072
01	A	12-1	61	A	12-1	101
02	B	12-2	62	B	12-2	102
03	C	12-3	63	C	12-3	103
04	D	12-4	64	D	12-4	104
05	E	12-5	65	E	12-5	105
06	F	12-6	66	F	12-6	106
07	G	12-7	67	G	12-7	107
10	H	12-8	70	H	12-8	110
11	I	12-9	71	I	12-9	111
12	J	11-1	41	J	11-1	112
13	K	11-2	42	K	11-2	113
14	L	11-3	43	L	11-3	114
15	M	11-4	44	M	11-4	115
16	N	11-5	45	N	11-5	116
17	O	11-6	46	O	11-6	117
20	P	11-7	47	P	11-7	120
21	Q	11-8	50	Q	11-8	121
22	R	11-9	51	R	11-9	122
23	S	0-2	22	S	0-2	123
24	T	0-3	23	T	0-3	124
25	U	0-4	24	U	0-4	125
26	V	0-5	25	V	0-5	126
27	W	0-6	26	W	0-6	127
30	X	0-7	27	X	0-7	130
31	Y	0-8	30	Y	0-8	131
32	Z	0-9	31	Z	0-9	132
33	0	0	12	0	0	060
34	1	1	01	1	1	061
35	2	2	02	2	2	062
36	3	3	03	3	3	063
37	4	4	04	4	4	064
40	5	5	05	5	5	065
41	6	6	06	6	6	066
42	7	7	07	7	7	067
43	8	8	10	8	8	070
44	9	9	11	9	9	071
45	+	12	60	+	12-8-6	053
46	-	11	40	-	11	055
47	*	11-8-4	54	*	11-8-4	052
50	/	0-1	21	/	0-1	057
51	(	0-8-4	34	(	12-8-5	050
52	)	12-8-4	74	)	11-8-5	051
53	$	11-8-3	53	$	11-8-3	044
54	=	8-3	13	=	8-6	075
55	blank	no punch	20	blank	no punch	040
56	, (comma)	0-8-3	33	, (comma)	0-8-3	054
57	. (period)	12-8-3	73	. (period)	12-8-3	056
60	≡	0-8-6	36	#	8-3	043
61	[	8-7	17	[	12-8-2	133
62	]	0-8-2	32	]	11-8-2	135
63	%[††]	8-6	16	%[††]	0-8-4	045
64	≠	8-4	14	" (quote)	8-7	042
65	↱	0-8-5	35	_ (underline)	0-8-5	137
66	∨	11-0	52	!	12-8-7	041
67	∧	0-8-7	37	&	12	046
70	↑	11-8-5	55	' (apostrophe)	8-5	047
71	↓	11-8-6	56	?	0-8-7	077
72	<	12-0	72	<	12-8-4	074
73	>	11-8-7	57	>	0-8-6	076
74	≤	8-5	15	@	8-4	100
75	≥	12-8-5	75	\	0-8-2	134
76	¬	12-8-6	76	¯ (circumflex)	11-8-7	136
77	; (semicolon)	12-8-7	77	; (semicolon)	11-8-6	073

[†] Twelve zero bits at the end of a 60-bit word in a zero byte record are an end of record mark rather than two colons.

[††] In installations using a 63-graphic set, display code 00 has no associated graphic or card code; display code 63 is the colon (8-2 punch). The % graphic and related card codes do not exist and translations yield a blank (55_8).

TABLE A-2. CDC CHARACTER SET COLLATING SEQUENCE

Collating Sequence Decimal/Octal		CDC Graphic	Display Code	External BCD	Collating Sequence Decimal/Octal		CDC Graphic	Display Code	External BCD
00	00	blank	55	20	32	40	H	10	70
01	01	≤	74	15	33	41	I	11	71
02	02	%	63 †	16 †	34	42	∨	66	52
03	03	[	61	17	35	43	J	12	41
04	04	→	65	35	36	44	K	13	42
05	05	≡	60	36	37	45	L	14	43
06	06	∧	67	37	38	46	M	15	44
07	07	↑	70	55	39	47	N	16	45
08	10	↓	71	56	40	50	O	17	46
09	11	>	73	57	41	51	P	20	47
10	12	≥	75	75	42	52	Q	21	50
11	13	¬	76	76	43	53	R	22	51
12	14	.	57	73	44	54	]	62	32
13	15	)	52	74	45	55	S	23	22
14	16	;	77	77	46	56	T	24	23
15	17	+	45	60	47	57	U	25	24
16	20	$	53	53	48	60	V	26	25
17	21	*	47	54	49	61	W	27	26
18	22	−	46	40	50	62	X	30	27
19	23	/	50	21	51	63	Y	31	30
20	24	,	56	33	52	64	Z	32	31
21	25	(	51	34	53	65	:	00 †	none†
22	26	=	54	13	54	66	0	33	12
23	27	≠	64	14	55	67	1	34	01
24	30	<	72	72	56	70	2	35	02
25	31	A	01	61	57	71	3	36	03
26	32	B	02	62	58	72	4	37	04
27	33	C	03	63	59	73	5	40	05
28	34	D	04	64	60	74	6	41	06
29	35	E	05	65	61	75	7	42	07
30	36	F	06	66	62	76	8	43	10
31	37	G	07	67	63	77	9	44	11

†In installations using the 63-graphic set, the % graphic does not exist. The : graphic is display code 63, External BCD code 16.

TABLE A-3. ASCII CHARACTER SET COLLATING SEQUENCE

Collating Sequence Decimal/Octal		ASCII Graphic Subset	Display Code	ASCII Code	Collating Sequence Decimal/Octal		ASCII Graphic Subset	Display Code	ASCII Code
00	00	blank	55	20	32	40	@	74	40
01	01	!	66	21	33	41	A	01	41
02	02	"	64	22	34	42	B	02	42
03	03	#	60	23	35	43	C	03	43
04	04	$	53	24	36	44	D	04	44
05	05	%	63†	25	37	45	E	05	45
06	06	&	67	26	38	46	F	06	46
07	07	'	70	27	39	47	G	07	47
08	10	(	51	28	40	50	H	10	48
09	11	)	52	29	41	51	I	11	49
10	12	*	47	2A	42	52	J	12	4A
11	13	+	45	2B	43	53	K	13	4B
12	14	,	56	2C	44	54	L	14	4C
13	15	–	46	2D	45	55	M	15	4D
14	16	.	57	2E	46	56	N	16	4E
15	17	/	50	2F	47	57	O	17	4F
16	20	0	33	30	48	60	P	20	50
17	21	1	34	31	49	61	Q	21	51
18	22	2	35	32	50	62	R	22	52
19	23	3	36	33	51	63	S	23	53
20	24	4	37	34	52	64	T	24	54
21	25	5	40	35	53	65	U	25	55
22	26	6	41	36	54	66	V	26	56
23	27	7	42	37	55	67	W	27	57
24	30	8	43	38	56	70	X	30	58
25	31	9	44	39	57	71	Y	31	59
26	32	:	00†	3A	58	72	Z	32	5A
27	33	;	77	3B	59	73	[	61	5B
28	34	<	72	3C	60	74	\	75	5C
29	35	=	54	3D	61	75	]	62	5D
30	36	>	73	3E	62	76	^	76	5E
31	37	?	71	3F	63	77	_	65	5F

†In installations using a 63-graphic set, the % graphic does not exist. The : graphic is display code 63.

ERROR PROCESSING AND DIAGNOSTICS

All user requests are checked to ensure proper processing. If results are not satisfactory, an error condition exists and the following occurs:

- A three-digit octal error code is returned in the error status (ES) field of the FIT.
- For a parity error, a severity level is set in the system parity error severity (SES) field.
- For a fatal error, the fatal/nonfatal flag (FNF) field is set in the FIT.
- Action indicated by the user setting of the error option (EO) field takes place, as discussed elsewhere in this section.
- An error exit is taken if the user has set the error exit (EX) field of the FIT and SBF=NO. If SBF=YES, no error exit is taken for GET or PUT.
- Error messages and notes are written to the dayfile and/or the ZZZZZEG error file depending on the values of the dayfile control (DFC) and error file control (EFC) fields.

ERROR COMMUNICATION

Regarding errors, the user and the error processor communicate through FIT fields ES, EX, EO, ERL, ECT, and PEF. The error status (ES) field is a 9-bit field set to an octal value after an attempt at error resolution is made and control is ready to be returned to the user. When an attempt is made to execute an input or output request after an error, the ES field is not cleared. If the request is not legal, the trivial error count (ECT) is incremented, and execution proceeds. If a subsequent error is detected, the ES field reflects the most recent error. The user is responsible for clearing the ES field if an error exit (EX) is not supplied; the ES field is checked after every macro call.

FIT fields and their meaning relevant to error processing are:

- FNF

  Fatal/nonfatal flag; set to 1 for fatal errors.

- PEF

  Parity error flag; set to 1 for parity errors.

- SES

  System parity error severity; set to the severity level of the parity error. The levels have the meanings shown in table B-1.

TABLE B-1. TYPES OF PARITY ERRORS

Value	Severity	Explanation
1	Read parity error level 1	Recovery to record boundary is possible. The number of bad records and blocks is known. BAM can recover.
2	Read parity error level 2	Recovery to record boundary is possible. The number of bad blocks is known but not the number of lost records. BAM can recover.
3	Read parity error level 3	Recovery to record boundary is possible. The number of bad records and blocks is unknown. BAM can recover.
4	Read parity error level 4	Recovery to record boundary is not possible. Fatal, BAM cannot recover.
5	Write parity error level 1	Irrecoverable tape write parity error. CLOSEM VOLUME recommended.
6	Write parity error level 2	Irrecoverable tape write parity error. CLOSEM VOLUME cannot be executed.

- EX

  Error exit; interpreted as follows:

EX=0	No user error routine; control is returned as a normal exit; the ES field is set with an error code. If the value of EX is zero and a fatal (F) error is encountered, the message is put on the dayfile.
EX≠0	If a fatal or trivial error occurs, control is transferred to EX+1; a jump to the user in-line return address is stored in the EX field, and the ES field is set.

- ERL

Trival error limit which can be specified by the user; interpreted as follows:

ERL=0 Limit not specified; no error count is accumulated. The number of trivial error permitted is indefinite.

ERL≠0 The job is terminated when the value of the ECT field reaches the value of the ERL field.

- EO

Error option; the EO field is used in conjunction with parity errors. If the TD, AD, or DD option is used and the EFC field is set to 3, the block containing the parity error is dumped to the error file for display by the error processor. The EO field is interpreted as follows:

EO=T or TD All parity errors are fatal.

EO=A or AD All parity errors should be disregarded (the bad data read as if it were good), but the ES field is set to 137 and control is passed to the error exit (EX) routine at the end of the record. If another error occurs when trying to read bad data, error 137 is overwritten by the next error; however, the parity error flag (PEF) remains set.

EO=D or DD The block in which the parity error occurs is dropped and BAM attempts to find the start of the next good record. If successful, the error exit is taken with the ES field set to 137, the SES field set to 3, and the FNF field set to 0. The content of the working storage area is undefined, and the file is positioned in front of the next good record. If unsuccessful, the error exit is taken with the ES field set to 137, the SES field set to 3, and the FNF field set to 1 (fatal).

- DFC

Dayfile control; set by the user to control the listing of error messages on the dayfile. The DFC field is interpreted as follows:

DFC=0 No dayfile messages except fatal errors (default).

DFC=1 Error messages to the dayfile.

DFC=2 Notes to the dayfile.

DFC=3 Error messages and notes to the dayfile.

- EFC

Error file control; set by the user to control the listing of error messages on the error file. The EFC field is interpreted as follows:

EFC=0 No error file entries (default).

EFC=1 Error messages to the error file.

EFC=2 Notes to the error file.

EFC=3 Error messages and notes to the error file.

The system message disposition (SDS) and extended diagnostic (EXD) fields of the FIT, which were part of a previous version of the error processor, are replaced by the DFC and EFC fields. If the SDS or EXD fields are used with the FILE macro, a warning assembly diagnostic is issued and no comparable values are placed in the DFC and EFC fields. If they are used with the FETCH or STORE macro, they are translated into compatible values for the DFC and EFC fields. The SDS field set to YES is equivalent to the DFC field set to 2. The EXD field set to YES is equivalent to the EFC field set to 1.

ERROR PROCESSING

If the EFC field is set to nonzero, the CRMEP control statement can be used to process the ZZZZZEG error file and control the listing of error messages on the output file. The error file is always flushed when abnormal termination occurs. At the completion of a job step, the error file buffer is flushed if all files are closed. The format of the CRMEP control statement is shown in figure B-1. The parameters, options, and defaults for the CRMEP control statement are listed in table B-2. The first default is set if neither the parameter nor the option is specified. The second default is set if the parameter is specified without an option. More than one option can be specified with a parameter, and more than one parameter can be specified on one CRMEP control statement. If a parameter is incorrectly specified, the CRMEP control statement ignores the incorrect parameter and those following it.

CRMEP (parameter=option$_1$/option$_2$/ ... /option$_n$, ...)

parameter Mnemonic specifying type of error file processing and listing.

option Selected setting of the specified parameter.

Figure B-1. CRMEP Control Statement Format

The FITDMP macro can be used to capture the contents of FIT fields for display by the post error processor (CRMEP). When the FITDMP macro is executed, the FIT, and the FIT display identifier if the id parameter is specified, are written to the ZZZZZEG error file. The CRMEP control statement can then be used to display the FIT on the output file. The format of the macro is shown

TABLE B-2. CRMEP CONTROL STATEMENT PARAMETERS

Parameter	Option	First Default	Second Default	Description
LO	N		X	Select notes.
	-N	X		Omit notes.
	F	X	X	Select fatal error messages.
	-F			Omit fatal error messages.
	D	X	X	Select data manager messages.
	-D			Omit data manager messages.
	T		X	Select trivial error messages.
	-T	X		Omit trivial error messages.
SF	$lfn_1/lfn_2/.../lfn_n$	All	All	Select messages associated with specified files.
OF	$lfn_1/lfn_2/.../lfn_n$	None	None	Omit messages associated with specified files.
SN	$mno_1/mno_2/.../mno_n$	All	Hardware and parity errors	Select only specified message numbers.
ON	$mno_1/mno_2/.../mno_n$	None	Error messages 142 and 143 only	Omit only specified message numbers.
L	lfn	OUTPUT	LIST	Specify output file name.
RU	blank			Return unload of error file performed at end of processing.
	0	X		Error file position at EOI at end of processing.
PW	pw	72 (connected file) 132 (unconnected file)	72 (connected file) 132 (unconnected file)	Specify page width for CRMEP output file (range can be 40-160 characters).

in figure B-2. The FIT display identifier, which can be up to ten characters, identifies the particular fit dump. The id parameter specifies the location of the display identifier.

```
FITDMP  fit,id

fit     Address of the FIT.

id      Address of the FIT display identifier.
```

Figure B-2. FITDMP Macro Format

To ensure that notes are written to the error file, the EFC field of the FIT must be set to 2 or 3. When the FITDMP macro is used, the EFC field is forced to 2 when set to 0 and forced to 3 when set to 1. Note number 1000 is reserved for user FIT dumps.

Upon encountering an error condition, the error status (ES) field is set to the appropriate error number, the trivial error count (ECT) field is incremented, and it is compared with the trivial error limit (ERL) field. If the ERL field is zero, unlimited errors are allowed and the ECT field is not incremented in the FIT. If the value of the

ERL field is nonzero and the ECT field is less than the ERL field, control passes to the error exit (EX) routine if defined, or back to the user's in-line code if the EX field is zero. In the latter case, it is the user's responsibility to check the error status. If the ERL field is nonzero and the value of the ECT field is equal to the value of the ERL field, the ES field is set to 356 (trivial error limit reached). The fatal/nonfatal (FNF) flag is set and another message is written. Control is returned as described above. If the FNF flag is set and any other function is attempted on the file, a 115 error is generated and the job is aborted.

CLASSES OF ERRORS

Syntax errors are diagnosed. The messages are self-explanatory. System errors are detected by the operating system. Execution errors, occurring during execution of input and output requests, are subdivided into call errors and invalid input/output requests.

CALL ERRORS

Call errors are undetectable parameter errors, such as:

```
GET         X1
```

If register X1 does not contain the valid FIT address, an unpredictable BAM error or hardware mode error can result.

INVALID INPUT/OUTPUT REQUESTS

Requests for illegal input/output operations produce the following general types of errors:

- FIT

  Content of address given as the FIT address does not pass a test for plausibility. It does not contain a legal logical file name in bits 59 through 18, or the FIT has inconsistencies.

- File organization

  Attempts to issue input/output requests or specifications are illegal on the type of file specified in the FO field of the FIT.

- Block type

  Attempts to issue input/output requests are illegal for the block type specified in the BT field of the FIT.

- Record type

  Attempts to issue input/output requests are illegal for the record type specified in the RT field of the FIT.

- OPENM/CLOSEM

  Input/output requests are illegal for files opened or closed as specified in the OC and/or ON fields of the FIT.

- Processing direction

  Input/output requests that would violate the processing direction limitations specified in the PD field of the FIT.

- File position

  Input/output requests are illegal for the file position given by the FP field of the FIT.

- Last operation

  Input/output requests are illegal in the context of the last operation; for example, a read after a write on tapes.

- Key

  Attempts to access or write records whose keys are not within the range of keys defined for a file. This includes attempts to access sequential files by keys.

- Data

  Errors in data specification, such as inconsistency between the amount of data requested and the amount actually present, illegal field present in the data, required field absent, or parity errors.

- Device

  Attempts to execute an input/output request are illegal on the device upon which the file resides.

- Label

  Label information submitted by the user does not correspond with the existing label, or the label is incorrectly formatted.

All errors are fatal or nonfatal. Some nonfatal errors are trivial in that no user action is required. Fatal errors usually indicate incorrect parameter specification and incomplete or contradictory information which is a user program error. A fatal diagnostic is always printed on the dayfile.

If an EX field has been specified in the FIT, any error causes a transfer of control to the address in EX+1 for a recovery routine after the error has been resolved. Fatal errors inhibit any further attempts at input/output on the file. Such attempts cause the job to terminate. In the absence of a value in the EX field, errors set the ES field and return control to the calling program. The ES field is not cleared after an error.

BAM is in the user's field length and is subject to destruction by the user.

DIAGNOSTICS

Table B-3 is a list of notes or informative messages.

TABLE B-3. NOTES OR INFORMATIVE MESSAGES

Code	Message
1000	USER FIT DUMP .....
1137	THE FOLLOWING BLOCK CONTAINS A PARITY ERROR .....

Table B-4 contains the following:

- Code

  Octal value corresponding to the error condition.

- Message

  Diagnostic output which varies depending on the values of the DFC and EFC fields, and the parameters specified with the CRMEP control statement.

- Significance

  Meaning of the message.

- Action

  Suggestion for the user to correct the error condition.

- Severity

  Type of error; can be any of the following:

F	Fatal
T	Trivial
T/F	Trivial under some conditions, fatal under others.

TABLE B-4. DIAGNOSTICS

Code	Message	Significance	Action	Severity
001	INVALID FO	File organization must be sequential (SQ) or word addressable (WA).	Correct the file organization field.	F
002	FIT/FILE ORGANIZATION MISMATCH	The file organization specified does not match any opened files.	Check to see that the correct file is being processed, or that the FO field is specified correctly.	F
020	INVALID BT	Block type must be I, C, K, or E.	Correct the block type field.	T
022	W RECORDS DISALLOWED ON BT=E/K	W type records cannot be written for E or K type blocks.	Correct the record type or block type field.	T
025	BT=I, RT NE W	I type blocks require W type records.	Correct the block or record type field.	T
026	SQ BTS REQUIRE MBL	Maximum block length must be specified for SQ files with K or E type blocks.	Specify the maximum block length field.	T
030	INVALID RT	Record type must be W, S, Z, F, R, T, D, or U; it must conform to other file specifications, such as block type or file organization.	Correct the record type field.	T
031	RT=F/Z AND FL=0	For fixed length F or zero-byte terminated Z type records, a maximum record length must be specified in the FL field of the FIT.	Specify the maximum record length field.	T

TABLE B-4. DIAGNOSTICS (Cont)

Code	Message	Significance	Action	Severity
032	RT=T AND HL OR TL=0	For T type records, the header length (HL) must be large enough to hold the CL that defines the length of the trailer count field. The length of the trailer count field must be given in TL and must be at least one character long.	Correct the header length or the trailer length field.	T
033	RT=D AND LL=0/RT=T AND CL=0	For D type records, the LL field of the FIT must provide the length of the record field that specifies record length.	Specify the length of the D type record length field.	T
		For T type records, the CL field of the FIT must provide the length of the field that specifies the number of trailer items.	Specify the length of the trailer count field of the T type record.	
034	RT=T/D, PTL ON FIRST PUTP EXCLUDES CONTROL FIELD	D and T records must have control field within PTL characters transferred during first PUTP issued. PTL must be greater than or equal to LP+LL+1 for D type records and CP+CL+1 for T type records for the first PUTP issued per record.	Correct the program.	F
035	RT=T/D, MRL EXCLUDES CONTROL FIELD	For T and D type records, the record must contain a field identifying record length.	Check that for D type records LP+LL is less than MRL. For T type records, CP+CL must be less than MRL. The position count for LP and CP begins with 0.	T
036	RL INCONSISTENT WITH RECORD DESCRIPTION	For T type records, the fixed header length (HL) must include a count field CL characters long, beginning at CP, to identify trailer item count. The length field specified by LL and LP for D type records must be within MNR.	For T type records, check that the count field is within HL. For D type records, check that the length field is within MNR. The current record is ignored. Positions CP and LP are counted from 0.	T
037	RT=D/T AND CL/LL>6	For D and T type records, the length of the count field must be one to six character positions.	Correct the length of the count field.	T
040	REDUNDANT OPEN	A file must be closed before open processing, such as user label processing for sequential files or buffer allocation, takes place. A redundant open call is ignored.	Correct the program to close the file before open processing.	T
047	OPEN EXTEND ON TAPE FILE	The E option for OPENM is valid only for a sequential file on mass storage.	Change the E option for the OPENM macro.	T

TABLE B-4. DIAGNOSTICS (Cont)

Code	Message	Significance	Action	Severity
051	SETFIT DISALLOWED ON OPEN FILE	Open processing would have already processed the FILE control statement. The SETFIT function processes FILE control statements without full open processing.	Change the placement of the SETFIT macro.	T
060	REDUNDANT CLOSE	A second call to close the file was issued. The operations requested by the CF field are performed before the error is issued.	Correct the program to eliminate the redundant close operation.	T
070	OUTPUT REQUEST, PD=INPUT OR READ ONLY PERMISSION	A file opened with read only permission or with pd set to INPUT cannot be written. The write statement is ignored.	If the file is to be written, store OUTPUT or IO in the PD field of the FIT prior to opening the file and check the file permissions.	T
071	INPUT REQUEST, PD=OUTPUT	A file opened with pd set to OUTPUT cannot be read. The read statement is ignored.	If the file is to be read, store INPUT or IO in the PD field of the FIT before opening the file.	T
100	CANNOT SEQUENTIALLY POSITION BEYOND FILE BOUNDS	A sequential read or SKIPFL is not possible with the file at EOI. A SKIPBL is not possible with the file at BOI.	The file must be repositioned if further access is desired. Repeated access attempts with file at the end cause the fatal error flag to be set.	F
104	UNABLE TO FLUSH BUFFER	A parity or system error might exist in an output sequential file just prior to a close request that requires the buffer to be flushed.	Rerun the program.	T
110	FILE NOT OPEN	A file must be opened before it can be read or written. Omission of required FIT field parameters or inconsistencies in parameters specified inhibit open.	Correct the program to open the file before reading or writing; or, correct omissions or inconsistencies of FIT fields.	T
111	NO CHECK ON LAST REQUEST	The CHECK or CHECKR macro must be issued after each GETWR or PUTWR macro.	Correct the program to issue the CHECK or CHECKR macro.	T
113	GET/PUT CANNOT BE USED IF SBF=YES	If file organization is sequential, only the GETWR or PUTWR macros can be used if the SBF field is set to YES.	Correct the program to use the GETWR or PUTWR macro or set the SBF field to NO.	T
115	OUTSTANDING FATAL ERROR ON THE FILE	A fatal error prevents future access to the file with the error, but it does not cause job termination unless the user attempts further operations on the file.	Correct and rerun.	F
116	GET FOLLOWS AN OUTPUT OPERATION. FO=SQ	A sequential file cannot be read immediately after a write.	Continue writing, or reposition the file before a read. The current read statement is ignored.	T
120	INVALID KEY/WORD ADDRESS/RECORD NUMBER	Word address for a word addressable file must be less than EOI for GET macros.	Correct the word address field.	T

TABLE B-4. DIAGNOSTICS (Cont)

Code	Message	Significance	Action	Severity
123	PARTIALS NOT SUPPORTED FOR RT=R	The GETP and PUTP macros cannot be issued for a sequential file with R records.	Correct the program to use the GET or PUT macros.	T
130	RT=W BAD CONTROL WORD, FILE DEFECTIVE OR MISPOSITIONED	Record type was specified as W. This message indicates the records being read are not, in fact, W type records.	Check that the existing file is correctly described, formatted, and positioned.	T/F
135	RMS READ PARITY ERROR	The operating system returned parity error status after reading a word addressable file.	Recreate the file on a good device. If the error persists, follow site-defined procedures for reporting software errors or operational problems.	T/F
136	RMS WRITE PARITY ERROR	The operating system returned parity error status after writing a word addressable file.	Recreate the file on a good device. If the error persists, follow site-defined procedures for reporting software errors or operational problems.	F
137	SQ READ PARITY ERROR	A parity error occurred while reading a sequential file.	Check the SES field of the FIT for severity and retry. (See the beginning of this appendix.)	T/F
140	SQ WRITE PARITY ERROR	A parity error occurred while writing a sequential file.	Check the SES field of the FIT for severity and retry. (See the beginning of this appendix.)	T/F
141	EXCESS DATA IS FATAL TO PUTP	The value of the RL field is greater than the value of the MRL field during a series of PUTP macros. The error is fatal because part of the bad record is already in the file.	Correct the program.	F
142	EXCESS DATA	In a write, no information is written to the file. For a read, MRL characters are transferred to the working storage area and remaining record characters skipped. On a read, the record length exceeds FL/MRL defined. For GET processing, the following conditions cause an error. Record types: W RL in control word>MRL Z No zero byte found before FL characters R No record mark found before MRL T,D Control field RL>MRL S MRL reached before level number encountered	Correct the inconsistency between the RL and FL or MRL fields.	T

TABLE B-4. DIAGNOSTICS (Cont)

Code	Message	Significance	Action	Severity
		U RL>MRL F Excess data cannot occur On PUT processing, the record mark character for an R type record was not found before MRL characters, or the user has supplied RL>MRL/FL.		
143	INSUFFICIENT DATA	Control information in the record being read (record length in a W type control word, or length calculated by fields such as CP and CL) specifies a length for each record. The record existing in the file is smaller than the specified length. All characters available are returned. For I and C type blocks, an end-of-section was encountered before the record terminated. For K and E type blocks, the block end occurred before the record ended. The data transferred through PUTPs is less than FL for an F type record.	No action is required.	T
144	INCOMPLETE PARTIAL PUT SEQUENCE	The previous record was not complete.	Correct the program.	F
150	FILE NOT ON RMS	Word addressable files must be created on a disk, drum, or family pack.	Correct the control statement to ensure a valid device assignment.	T/F
152	LT=S, DT=RMS	Standard labels, which conform to ANSI standards, can exist only on tape files. Label processing statements are ignored because the file is assigned to rotating mass storage.	Correct the inconsistency between label type and device type fields.	T
154	BT=K/E ON PRU TYPE DEVICE	K or E type blocking is possible only for files on S or L tapes.	Change block type, or add an S or L parameter to the REQUEST or LABEL control statement.	T
157	S-TAPE BUT MBL>5120 CHARACTERS	Maximum block length for S tapes is 5120 characters.	Change MBL to an allowable value or use an L tape.	T
162	INVALID CONVERSION	The CM field of the FIT must not be YES for W type records.	Change the conversion mode field.	T
165	ILLEGAL FILE NAME	The LFN does not consist of one to seven letters and digits, the first being a letter.	Correct the LFN or the FIT address.	F

TABLE B-4. DIAGNOSTICS (Cont)

Code	Message	Significance	Action	Severity
167	RECORD LENGTH OUTSIDE MIN-MAX RANGE - REQUEST IGNORED	For D or T type records, the control field specified is outside the value specified by the RL field, or not within the values specified by the MNR and MRL fields.	Check to see that the CL/CP fields or the LL/LP fields are specified correctly.	T
170	RECORD SIZE EXCEEDS BLOCK SIZE OR IS NEGATIVE	For K and E type blocking, records cannot be split between blocks. Individual records must be smaller than the block defined by MBL or the maximum block allowed on the device.	Correct the RL or MBL field.	T/F
173	INVALID RL/PTL/MBL/MNB	The record length, partial transfer length, block size, or minimum block length is specified incorrectly.	Correct the RL, PTL, MBL, or MNB field. Make sure MNB is less than MBL.	T
207	MINIMUM RECORD SIZE EXCEEDS MAXIMUM	Required parameter MRL must be equal to or larger than MNR.	Correct the inconsistency between the MRL and MNR fields.	F
245	FUNCTION DISALLOWED ON THIS FO	The macro issued is not valid for the file organization specified in the FIT.	Correct the program.	T
254	PARTIALS NOT SUPPORTED FOR FO=WA	The GETP and PUTP macros cannot be issued for a word addressable file.	Correct the program to use the GET or PUT macro.	T
255	RECORD SPECIFICATION NOT COMPATIBLE WITH SBF=YES	For word addressable files with the SBF field set to YES, the RL field must be a multiple of PRU size and the WA field must be a multiple of PRU size plus one.	Correct the program to specify correct values for the RL and WA fields, or set the SBF field to NO to allow a buffer to be allocated.	T
256	PARTIALS NOT SUPPORTED FOR RT=R	PUTPs cannot be issued for R type records.	Correct the program.	F
300	NO READ PERMISSION	To be read, a permanent file must be attached with read permission.	Attach the file with the required read permission.	F
301	NO WRITE OR MODIFY PERMISSION	A permanent file requires proper access permissions. Modify permission is required for any updating operation.	Attach the file with the required write permission.	F
302	NO EXTEND OR ALLOCATE PERMISSION	A permanent file requires extend permission before new records can be inserted.	Attach the file with the required extend permission.	F
312	INVALID LABEL GROUP	Labels that can be accessed are affected by the current file position. Header labels, for example, cannot be accessed at end-of-information.	Check that file position is consistent with label action requested.	F
315	FILE ORGANIZATION IS NOT SEQUENTIAL	Standard labels can be used only with sequential files on tape.	Check that file organization is consistent with label type.	F

TABLE B-4. DIAGNOSTICS (Cont)

Code	Message	Significance	Action	Severity
316	TOO MANY LABELS	The number of labels that can be written is limited by ANSI standards.	Correct the program.	F
320	INVALID LABEL SEQUENCE	The ULP option controls the type of labels that can be accessed.	Remove conflicts between ULP and the type of label.	F
321	DATA STRUCTURES MUST BE WORD ALIGNED	If WSA or KA is provided as a CHARACTER type data structure, it must be word aligned within that structure.	Word align WSA and/or KA.	F
325	STANDARD LABELS NOT ALLOWED ON MASS STORAGE	LT=S is valid only for tape files.	Correct the inconsistency between label type and device.	F
326	GETL/PUTL ILLEGAL ON UNLABELED FILE	A tape file must have a label declared on a REQUEST or LABEL control statement before user label access is possible.	Change the label type field.	T
327	GETL ATTEMPTED BEYOND END OF LABELS	Tapemarks separating data and labels stop label processing.	Correct the program.	F
330	INVALID PARAMETER VALUE (LA, LBL, ULP)	LA must be zero or an address in a user program. LBL must indicate the length of the label area, 0 to 900 characters. ULP options are V, F, U, VU, VF, FU, VFU, and NO.	Check GETL or PUTL parameters of FIT fields.	F
332	FILE REQUEST LABEL TYPE DISAGREES WITH LT FIELD OF FIT	When a REQUEST control statement specifies a labeled tape, the user must set LT to S.	Correct the inconsistency between the REQUEST or LABEL control statement and the label type field.	T
346	CMM NOT AVAILABLE AND THERE IS NO LIST OF FILES ADDRESS	A new block for the list-of-files cannot be allocated, and the LOF$RM entry point has been cleared.	Correct the program to not destroy the pointer. A default list with sixty-five entries is supplied.	F
347	FDL ERROR CODE n ON CAPSULE axxxxxx	Either CMM is not loaded when FDL is called to load a capsule, or the BAMLIB file is not valid.	Check the load sequence or map to see if CMM is loaded. Fix the static load calls to load the proper routines. If using local libraries, check for a valid BAMLIB file. If not using local libraries, follow site-defined procedures for reporting software errors or operational problems.	T
352	FILE TO BE CLOSED IS NOT KNOWN	The logical file name specified does not match any existing file.	Check that the logical file name is correctly specified.	T
354	BUFFER SPACE SUPPLIED IS INSUFFICIENT FOR I/O	A buffer specified by BFS (size in words) is not large enough to hold at least the larger of one block specified by MBL+2 (block length in characters) or one physical record unit for the file's resident device. A record written on a connected file on NOS/BE is larger than the current buffer.	Increase the BFS value.	T/F

TABLE B-4. DIAGNOSTICS (Cont)

Code	Message	Significance	Action	Severity
355	CODE MODULES REQUIRED FOR I/O NOT LOADED	Routines necessary for processing have not been loaded.	Refer to appendix E for correct static loading procedures.	T
356	TRIVIAL ERROR LIMIT REACHED	Error count ECT equals the user-defined error limit ERL, resulting in a fatal error.	Correct the errors.	F
357	UNABLE TO OBTAIN SPACE FOR BUFFER	Required space has not been allocated. CMM is not available, and the FWB field is zero.	Supply a value for the FWB field or remove the OMIT=CMM parameter.	F
403	SKIPBL DISALLOWED	A backward skip is not possible for D, U, R, and T type records or K and E type blocks.	Correct the program.	T
404	SKIPFL DISALLOWED FOR RT=U	No forward record skip is possible for U type records.	Correct the program.	T
406	REPLACE ATTEMPTED ON TAPE FILE	For sequential files, the REPLACE macro can be used only on disk files.	Copy the file to mass storage.	T
407	FO=SQ REPLACE ATTEMPTED WHEN FP≠EOR	The REPLACE macro must be preceded by a GET macro or a GETP macro of a full record.	Correct the program to read a full record before the REPLACE macro is issued.	T
410	FO=SQ REPLACE ATTEMPTED WHEN LOP≠GET	For sequential files, the record to be replaced must be read before the REPLACE macro is issued.	Correct the program.	T
411	FO=SQ REPLACE ENCOUNTERED EOS/EOP/BOI	The GET or REPLACE macro did not work properly.	Follow site-defined procedures for reporting software errors or operational problems.	T
412	FO=SQ REPLACE ILLEGAL FOR THIS RT - USE RT=F/W	For sequential files, the REPLACE macro can only be used with W or F type records.	Correct the program.	T
413	FO=SQ REPLACE ILLEGAL FOR THIS BT - USE BT=C	For sequential files, the REPLACE macro can only be used with C type blocks.	Correct the program.	T
452	FILE POSITIONING ERROR	An attempt was made to position the file beyond EOI.	Correct the program to check the FP field or specify the DX field.	F
556	OPEN FAILURE	System CIO OPEN request failed.	Check the CODE and STATUS field in the FIT.	F
712	NEGATIVE OR OVERSIZED ARGUMENT - WSA, SKP, OR LA	One of the parameters indicated was erroneously specified when a macro was issued.	Correct the program.	F
713	NEGATIVE OR OVERSIZED ARGUMENT - RL, ST, OR LBL	One of the parameters indicated was erroneously specified when a macro was issued.	Correct the program.	F
714	NEGATIVE EX OR DX PARAMETER	A negative value was specified for the DX or EX field.	Correct the program.	F

TABLE B-4. DIAGNOSTICS (Cont)

Code	Message	Significance	Action	Severity
715	NEGATIVE OR OVERSIZED ARGUMENT - WA OR KA	Either the WA or KA field was erroneously specified.	Correct the program.	F
716	NEGATIVE OR OVERSIZED ARGUMENT - PTL OR KP	Either the PTL or KP field was erroneously specified.	Correct the program.	F
717	NEGATIVE OR OVERSIZED ARGUMENT - MKL, POS, GPS, OR TRM	One of the parameters indicated was erroneously specified when a macro was issued.	Correct the program.	F
720	DEVICE CAPACITY EXCEEDED	The CIO read driver has encountered an error.	Check the job dayfile for the specific read driver error.	T
721	ERROR DETECTED BY OPERATING SYSTEM	A system and/or hardware error has been encountered that cannot be handled.	Check the job dayfile for a system and/or hardware error message.	F

GLOSSARY

Advanced Access Methods (AAM) -
A file manager that processes indexed sequential, direct access, and actual key file organizations and supports the Multiple-Index Processor.

Basic Access Methods (BAM) -
A file manager that processes sequential and word addressable file organizations.

Beginning-of-Information (BOI) -
The start of the first user record in a file. System information, for example tape labels of sequential files, can appear before the beginning-of-information.

Block -
A logical or physical grouping of records to make more efficient use of hardware. Only sequential files are blocked. One of the following block types must be specified by the programmer: C, I, K, or E.

Boundary -
A file boundary is a physical indication that marks a logical division within a sequential file. BOI and individual user records are always recognized; other boundaries are affected by the record and blocking type and the file storage device. A word boundary is the first character position in a central memory word.

Character -
A letter, digit, punctuation mark, or mathematical symbol forming part of one or more of the standard character sets. Also, a unit of measure used to specify block length, record length, and so forth. A character is represented in six bits as internal display code.

Circular Buffer -
A temporary central memory storage area that contains data during input/output operations. Routines that process I/O treat the first word of the buffer area as contiguous to the last word of the buffer area.

Close -
A set of terminating operations performed on a file when input and output operations are complete. All files processed by BAM must be closed.

Combined Input/Output (CIO) -
An operating system routine that performs input and output.

CRMEP Control Statement -
A control statement that processes the BAM error file.

CYBER Record Manager (CRM) -
A generic term relating to the common products BAM and AAM.

Default -
A value assumed in the absence of a user-specified value declaration for the parameter involved. Values for many defaults are defined by the installation.

End-of-Information (EOI) -
The end of the last user record in a file. Trailer labels are considered to be past the end-of-information. End-of-information is undefined for unlabeled S or L tapes.

Error File -
A special file created with the logical file name ZZZZZEG to hold BAM error messages; the file is processed by the CRMEP control statement.

Extended Memory -
Any extension to central memory.

Field -
A portion of a word or record; a subdivision of information within a record; also, a generic entry in a file information table identified by a mnemonic.

Field Length -
The area in central memory allocated to a particular job; the only part of central memory that a job can directly access. Contrasts with mass storage space or tapes allocated for a job and on which user's files reside.

File -
A logically related set of information; the largest collection of information that can be addressed by a file name. It starts at beginning-of-information and ends at end-of-information. Every file in use by a job must have a logical file name.

FILE Control Statement -
A control statement that supplies file information table values after a source language program is compiled or assembled but before the program is executed. Basic file characteristics such as organization, record type, and description can be specified in the FILE control statement.

File Information Table (FIT) -
A table through which a user program communicates with BAM. For direct processing through BAM, a user must initiate establishment of this table. All file processing executes on the basis of information in this table. The user can set FIT fields directly or use parameters in a file access call that sets the fields indirectly. Some product set members set the fields automatically for the user.

Installation Option -
One of several alternate means of processing that is selected when BAM is installed at a computer installation. Once an option is selected, all subsequent use of BAM is governed by the selection. For all options or limits defined as installation options, the user should consult with a system analyst to determine the valid limits.

I Tape (Internal) -
A magnetic tape with recording format of physical records containing the contents of 0 to 512 central memory words of binary information. I tapes are only supported under the NOS operating system.

Job Step -
The execution of a control statement.

Key -
Information used to identify a record.

LDSET -
The loader control statement. Various parameters include:

LIB	Make available the named library
USE	Load the routines named
STAT	Static loading requested
OMIT	Inhibit loading of routines named

Level Number -
An octal number 0 through 17_8 that is recorded in a short physical record unit or zero-length physical record unit marker; the number is used to form system-logical-record groups within files. Level number 17_8 indicates a logical end-of-partition. Level number 16_8 is used by checkpoint/restart and should not otherwise be specified by the user. The system creates system-logical-records with a level number of 0 for mass storage files and SI tapes when the user does not specify otherwise.

Load Set -
A group of control statements beginning with a call that causes information to be loaded into central memory and ending with a call for execution of a loaded program. Nonloader statements must not appear in a load set.

Logical File Name -
The name given to a file being used by a job. The name must be unique for the job and must consist of one to seven letters or digits, the first of which must be a letter.

L Tape (Long Stranger) -
A 7-track or 9-track, labeled or unlabeled magnetic tape with blocks containing more than 5120 characters. Normally written by other than CYBER 170-compatible systems.

Macro -
A single instruction which when compiled into machine code generates several machine code instructions.

Maintenance Run -
A program or job to update an existing file; technically refers to that part of the job from file open to file close.

Mass Storage -
A disk pack that can be accessed randomly. Extended memory is not considered mass storage.

Master File -
A file containing information about a set of entities; all information about a single entity constitutes a record in a file. A master file is normally kept up to date by a maintenance run.

Open -
A set of preparatory operations performed on a file before input and output can take place; required for all BAM files.

Owncode -
A routine written by the user to process certain conditions. Control passes automatically to user owncode routines defined in the FIT for:

DX	End-of-data condition
EX	Error condition
LX	Tape label processing

Partition -
A group of sections beginning with the first record after the end of the preceding partition and ending with a special record or condition, dependent on the block and record type and storage device. Generally, a partition is greater than a section and less than a file, but it can be equal to either or both.

Permanent File -
A file on a mass storage permanent file device that can be retained for longer than a single job. It is protected against accidental destruction by the system and can be protected against unauthorized access.

Physical Record -
On magnetic tape, information between interrecord gaps. It need not contain a fixed amount of data.

Physical Record Unit (PRU) -
The smallest unit of information that can be transferred between a peripheral storage device and central memory. The PRU size is permanently fixed for all mass storage devices, and SI and I tapes; the concept does not apply to S/L tapes.

PRU Device -
An SI or I format tape or a mass storage device in which information has a physical structure governed by physical record units (PRUs).

Random Access -
Access method by which any record in a file can be accessed at any time. Applies only to mass storage files with an organization other than sequential.

Record -
The largest collection of information passed between BAM and a user program in a single read or write operation. The user defines the structure and characteristics of records within a file by declaring a record format. The beginning and ending points of a record are implicit in each format.

Release System -
A software system delivered to a customer is the release system. In installing a system, the customer, but not an individual applications programmer, can use default values or parameters that differ from the released system.

Rewind -
To position a file at beginning-of-information.

SCOPE 2 -
An operating system on the CONTROL DATA CYBER 70 Model 76 and 7600 Computer Systems. 7000 Record Manager runs under SCOPE 2.

Section -
A division internal to a sequential file. Recognition of a section boundary is affected by block type, record type, and file residence. A section is a group of records beginning with the first record after the end of the preceding section and ending with a special record or condition, dependent on the block and record type and storage device. Generally, a section is greater than a record and less than a partition, but it can be equal to either or both. Sections are not defined on K and E type blocks.

Sequential Access -
A method in which only the record located at the current file position can be accessed. See Random Access.

Sequential (SQ) File -
A file with records in the physical order in which they were written. No logical order exists other than the relative physical record position.

S Tape (Stranger) -
A magnetic tape with recording format of physical records containing the contents of 512 central memory words of information.

SI Tape (System Internal) -
A magnetic tape with recording format of physical records containing the contents of 0 to 512 central memory words of binary information or 0 to 128 words of coded information. Coded SI tapes are not supported under the NOS operating system.

Volume -
A reel of magnetic tape or a disk pack is a volume. A given file can encompass more than one volume.

Word Address -
The relative location of the first word of a record in a word addressable file. Specified as the WA field of the file information table on a call for a read or write operation.

Word Addressable (WA) Files -
Word addressable files are mass storage files containing continuous data or space for data. Words within word addressable files are numbered from 1 to n, each word containing 10 characters. Retrieving or writing of data at any given word within the file is specified by the word number, called the word address.

Working Storage Area -
An area within the user's field length intended for receipt of data from a file or transmission of data to a file.

FILE INFORMATION TABLE STRUCTURE D

A file information table (FIT) must be associated with every file that uses BAM. For normal language requirements, compilers generate the FIT automatically; users writing in higher level languages do not need to be concerned with FITs and their generation. It is the COMPASS user's responsibility to supply the FIT; BAM provides the FILE macro, which creates the table.

Word and bit designations of the FIT fields are illustrated in figure D-1. The fields enclosed in parentheses can be accessed by the FETCH macro but cannot be changed. If a STORE macro is attempted on these fields, an assembly diagnostic results. Blank fields are reserved for CRM or CDC.

The FIT is activated by an OPENM request for the file. After a file is opened, the contents of the FIT can be updated with the FILE control statement or the STORE macro, with information from the processing macros, or by BAM as a result of processing the file. Information in the FIT can be retrieved with the FETCH macro.

The meanings of the FIT fields by word and bit are listed in table D-1. For convenience of the user, the COMPASS symbols are included with the applicable FIT fields. The first ten words of the FIT are used by BAM for communicating with the operating system. Generally, for any particular file organization, record, or block type, only a small portion of the total information specified here is required.

For the reader's convenience, the FIT fields are listed alphabetically with their word positions in table D-2.

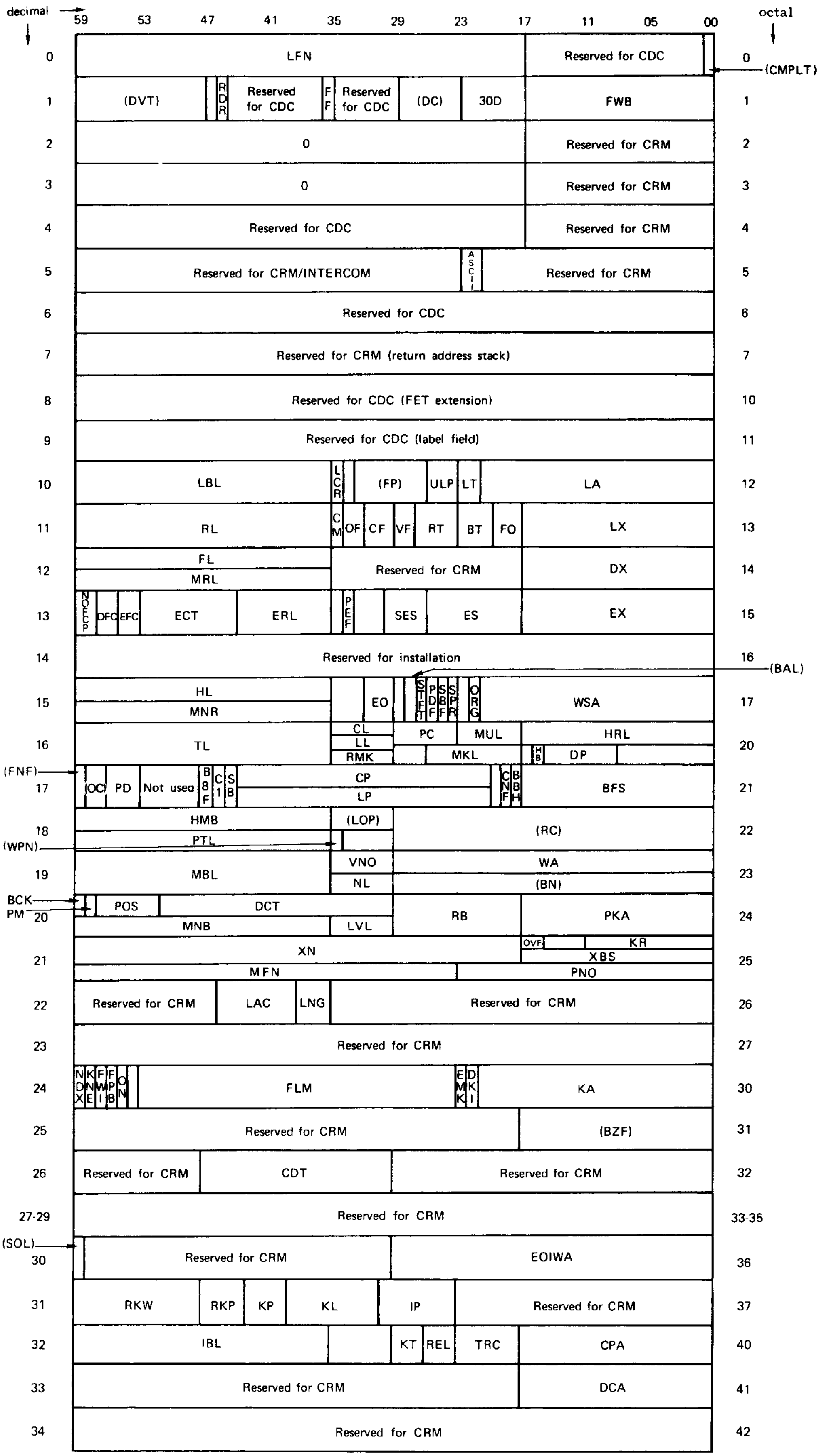

decimal
59 53 47 41 35 29 23 17 11 05 00
octal
0 LFN Reserved for CDC 0 (CMPLT)
1 (DVT) RDR Reserved for CDC FF Reserved for CDC (DC) 30D FWB 1
2 0 Reserved for CRM 2
3 0 Reserved for CRM 3
4 Reserved for CDC Reserved for CRM 4
5 Reserved for CRM/INTERCOM ASCII Reserved for CRM 5
6 Reserved for CDC 6
7 Reserved for CRM (return address stack) 7
8 Reserved for CDC (FET extension) 10
9 Reserved for CDC (label field) 11
10 LBL LCR (FP) ULP LT LA 12
11 RL CM OF CF VF RT BT FO LX 13
12 FL MRL Reserved for CRM DX 14
13 NOFCP DFC EFC ECT ERL PEF SES ES EX 15
14 Reserved for installation 16 (BAL)
15 HL MNR EO STFT PDF SBF SPR ORG WSA 17
16 TL CL LL RMK PC MUL MKL HRL HB DP 20
(FNF)
17 (OC) PD Not useo B8F C1 SB CP LP CNF BBH BFS 21
18 HMB PTL (LOP) (RC) 22
(WPN)
19 MBL VNO NL WA (BN) 23
BCK
PM
20 POS DCT MNB LVL RB PKA 24
21 XN OVF KR XBS MFN PNO 25
22 Reserved for CRM LAC LNG Reserved for CRM 26
23 Reserved for CRM 27
24 NDX KNE FWI FPB ON FLM EMK DKI KA 30
25 Reserved for CRM (BZF) 31
26 Reserved for CRM CDT Reserved for CRM 32
27-29 Reserved for CRM 33-35
(SOL)
30 Reserved for CRM EOIWA 36
31 RKW RKP KP KL IP Reserved for CRM 37
32 IBL KT REL TRC CPA 40
33 Reserved for CRM DCA 41
34 Reserved for CRM 42

Figure D-1. File Information Table

TABLE D-1. STRUCTURE OF THE FIT

Word	Bits	Fit Field	Description	Contents	COMPASS Symbol
0	59-18	LFN	Logical file name of the data file.		
	17-1		Reserved for CDC.		
	0	CMPLT	FET complete bit; cannot be changed by the user.		
1	59-48	DVT	FET device type; cannot be changed by the user.		
	47		Reserved for CDC.		
	46	RDR	Read release.		
	45-37		Reserved for CDC.		
	36	FF	File flush by operating system on abnormal termination.	0 NO Buffer not flushed.	≡NO≡
				1 YES Buffer flushed for an output file having scratch disposition.	≡YES≡
	35-30		Reserved for CDC.		
	29-24	DC	Disposition code; cannot be changed by the user. Refer to operating system manual for possible settings.		
	23-18		Length of FIT minus 5; set to 30_{10}.		
	17-0	FWB	First word address of the user buffer.		
2	59-18		Zero-filled field.		
	17-0		Reserved for CRM.		
3	59-18		Zero-filled field.		
	17-0		Reserved for CRM.		
4	59-34		Reserved for CDC.		
	33-0		Reserved for CRM.		
5	59-24		Reserved for CRM/INTERCOM.		
	23-22	ASCII	ASCII character set bits for INTERCOM terminals.	0 64 character display code 1 95 character ASCII subset 2 128 character ASCII	
	21-0		Reserved for CRM.		
6			Reserved for CDC.		
7			Reserved for CRM (return address stack).		

TABLE D-1. STRUCTURE OF THE FIT (Cont)

Word	Bits	Fit Field	Description	Contents		COMPASS Symbol
8			Reserved for CDC (FET extension).			
9			Reserved for CDC (label fields).			
10	59-36	LBL	Label area length in characters.			
	35	LCR	Label creation flag for input/ output tape.	0	CRT Create new labels.	≡NLCR≡
				1	CHK Check existing labels.	≡ELCR≡
	34		Reserved for CRM.			
	33-27	FP	File position (in octal); cannot be changed by the user.	0	Mid logical record	
				1	EOL End-of-label group BOI Beginning-of-information	≡EOL≡ ≡BOI≡
				2	BOF Beginning-of-file BOV Beginning-of-volume Only set on SKIPBU in connection with DX.	≡BOF≡ ≡BOV≡
				4	EOV End-of-volume	≡EOV≡
				10	EOS End-of-section	≡EOS≡
				20	EOR End-of-record	≡EOR≡
				40	EOP End-of-partition	≡EOP≡
				100	EOI End-of-information	≡EOI≡
	26-24	ULP	User label processing.	000	None	≡NOP≡
				001	V VOL/EOV	≡VP≡
				010	F HDR/EOF	≡FP≡
				011	VF VOL/HDR/EOF/EOV	≡VFP≡
				100	U UVL/UHL/UTL	≡UP≡
				101	VU VOL/UVL/UHL/EOV/UTL	≡VUP≡
				110	FU UVL/HDR/UHL/EOF/UTL	≡FUP≡
				111	VFU All	≡VFUP≡

TABLE D-1. STRUCTURE OF THE FIT (Cont)

Word	Bits	Fit Field	Description	Contents	COMPASS Symbol
	23-22	LT	Label type.	00 S ANSI standard	≡S≡
				01 NS Nonstandard	≡NS≡
				10 UL Unlabeled (default)	≡UL≡
				11 ANY Any	≡ANY≡
	21-0	LA	Label area address.		
11	59-36	RL	Current record length in characters.		
	35	CM	Conversion mode; convert sequential tape files from external to internal code.	0 NO No conversion	≡NO≡
				1 YES Conversion	≡YES≡
	34-33	OF	Open flag; positioning of the file at OPENM time.	00 Rewind (default)	≡ ≡
				01 R Rewind	≡R≡
				10 N No rewind	≡N≡
				11 E Extend	≡E≡
	32-30	CF	Close flag; position at file close.	000 Rewind (default)	≡ ≡
				001 R Rewind	≡R≡
				010 N No rewind	≡N≡
				011 U Unload	≡U≡
				100 RET Return	≡RET≡
				101 DET Detach	≡DET≡
				110 DIS Disconnect	≡DIS≡
	29-28	VF	Volume close flag; position of the file at end-of-volume.	00 Unload (default)	≡ ≡
				01 R Rewind	≡R≡
				10 N No rewind	≡N≡
				11 U Unload	≡U≡

TABLE D-1. STRUCTURE OF THE FIT (Cont)

Word	Bits	Fit Field	Description	Contents	COMPASS Symbol
	27-24	RT	Record type.	0000 W Control word	≡WT≡
				0001 F Fixed length	≡FT≡
				0010 R Record mark	≡RT≡
				0011 Z Zero byte	≡ZT≡
				0100 D Decimal character count	≡DT≡
				0101 T Trailer count	≡TT≡
				0111 U Undefined	≡UT≡
				1000 S System-logical-record	≡ST≡
	23-21	BT	Block type.	000 Internal (default)	≡≡
				001 I Internal	≡IT≡
				010 C Character count	≡CT≡
				011 K Record count	≡KT≡
				100 E Exact records	≡ET≡
	20-18	FO	File organization.	000 SQ Sequential	≡SQ≡
				001 WA Word addressable	≡WA≡
				011 IS Indexed sequential (AAM only)	≡IS≡
				101 DA Direct access (AAM only)	≡DA≡
				110 AK Actual key (AAM only)	≡AK≡
	17-0	LX	Label routine exit address.		
12	59-36	MRL	Maximum record length in characters.		
		FL	Fixed length of an F type record, or full length of a Z type record, in characters.		
	35-18		Reserved for CRM.		
	17-0	DX	End-of-data exit address.		

TABLE D-1. STRUCTURE OF THE FIT (Cont)

Word	Bits	Fit Field	Description	Contents		COMPASS Symbol
13	59	NOFCP	No FILE control statement processing.	0	NO FILE control statement processed at SETFIT or OPENM.	≡NO≡
				1	YES FILE control statement not processed.	≡YES≡
	58		Reserved for CRM.			
	57-56	DFC	Dayfile control for error messages.	0	No dayfile messages except fatal errors	
				1	Error messages to dayfile	
				2	Notes to dayfile	
				3	Errors and notes to dayfile	
	55-54	EFC	Error file control. The FITDMP macro forces EFC=0 to 2 and EFC=1 to 3.	0	No error file messages	
				1	Error messages to error file	
				2	Notes to error file	
				3	Errors and notes to error file	
	53-45	ECT	Trivial error count.			
	44-36	ERL	Trivial error limit.			
	35		Reserved for CRM.			
	34	PEF	Parity error flag.	0	No error	
				1	Parity error	
	33-31		Reserved for CRM.			
	30-27	SES	System parity error severity.	1	Read parity error level 1	
				2	Read parity error level 2	
				3	Read parity error level 3	
				4	Read parity error level 4	
				5	Write parity error level 1	
				6	Write parity error level 2	
	26-18	ES	Error status (octal value).			
	17-0	EX	Error exit address.			
14			Reserved for installation.			
15	59-36	HL	Header length of a T type record in characters.			
		MNR	Minimum record length.			

TABLE D-1. STRUCTURE OF THE FIT (Cont)

Word	Bits	Fit Field	Description	Contents		COMPASS Symbol
	35-33		Reserved for CRM.			
	32-30	EO	Error option.	000	T Terminate file.	≡T≡
				001	D Drop erroneous data.	≡D≡
				010	A Accept.	≡A≡
				100	TD Terminate file and display data.	≡TD≡
				101	DD Drop erroneous data and display data.	≡DD≡
				110	AD Accept erroneous data and display data.	≡AD≡
	29		Reserved for CRM.			
	28	BAL	Buffer allocated by CRM; cannot be changed by the user.			
	27	STFT	Internal SETFIT flag used for CRM processing.			
	26	PDF	SETFIT macro FILE statement flag.	0	FILE control statement not processed before OPENM.	
				1	FILE control statement was processed before OPENM.	
	25	SBF	Suppressed buffer I/O flag.	0	NO Buffer I/O.	≡NO≡
				1	YES Suppress buffer I/O.	≡YES≡
	24	SPR	Suppress read ahead; used by CRM during processing of blocks with read parity error and processing of EOFs on S and L tapes; unconditionally reset to NO at the end of processing.	0	NO Read ahead.	≡NO≡
				1	YES Read only one block at a time.	≡YES≡
	23		Reserved for CRM.			
	22	ORG	Old/new file organization field (AAM only).			
	21-0	WSA	Working storage area address.			
16	59-36	TL	Trailer length in characters; T type record.			
	35-30	CL	Count field length in characters; T type records.			
		LL	Length field length in characters; D type records.			

TABLE D-1. STRUCTURE OF THE FIT (Cont)

Word	Bits	Fit Field	Description	Contents	COMPASS Symbol
		RMK	Record mark character; R type records.		
	29-24	PC	Padding character for sequential files.		
	23-18	MUL	Multiple of characters per K or E type block.		
	26-18	MKL	Major key length in characters (AAM only).		
	17-0	HRL	Hashing routine address (AAM only).		
	16	HB	User header option (AAM only).		
	15-9	DP	Data block padding percent (AAM only).		
17	59	FNF	Fatal/nonfatal flag; cannot be changed by the user.	0 Nonfatal	
				1 Fatal	
	58-57	OC	Open/close flag.	00 Never opened	≡NOP≡
				01 Opened	≡OPE≡
				10 Closed	≡CLO≡
	56-54	PD	Processing direction.	000 Input	≡≡
				001 INPUT Input	≡INPUT≡
				010 OUTPUT Output	≡OUTPUT≡
				011 IO Input/output	≡IO≡
	53-48		Not used.		
	47	B8F	Round RL in PUTs for S type records down to an exact multiple of 8 bits; used in FORM and 8-bit subroutines.	0 NO Round up to an exact multiple of 6 bits.	≡NO≡
				1 YES Round down to an exact multiple of 8 bits.	≡YES≡
	46	C1	COMP-1; format for the length field within a T or D type record.	0 NO Display code	≡NO≡
				1 YES Binary	≡YES≡
	45	SB	Sign overpunch; overpunch option for CL/LL field for T or D type records.	0 NO No overpunch	≡NO≡
				1 YES Overpunch	≡YES≡

TABLE D-1. STRUCTURE OF THE FIT (Cont)

Word	Bits	Fit Field	Description	Contents		COMPASS Symbol
	44-21	CP	Trailer count beginning character position field of a T type record (numbered from 0).			
		LP	Length field beginning character position of a D type record (numbered from 0).			
	20		Reserved for CRM.			
	19	CNF	Connect file flag.	0	NO File not connected to terminal.	≡NO≡
				1	YES File connected to terminal.	≡YES≡
	18	BBH	Buffer below highest high address (HHA).	0	NO Buffer not below HHA.	≡NO≡
				1	YES Buffer below HHA.	≡YES≡
	17-0	BFS	Buffer size in words.			
18	59-36	HMB	Number of home blocks (AAM only).			
		PTL	Partial transfer length, set by the GETP or PUTP macro.			
	35-30	LOP	Last operation code; the high order bit of LOP is a write bit, indicating whether the last operation wrote data to the file; cannot be changed by the user.	01	OP OPENM	≡OP≡
				02	CM CLOSEM	≡CM≡
				03	GE GET or GETP	≡GE≡
				43	PU PUT or PUTP	≡PU≡
				56	RP REPLACE	≡RP≡
				04	SE SEEK (AAM only)	≡SE≡
				05	SF SKIPF	≡SF≡
				46	DE DELETE (AAM only)	≡DE≡
				07	GN GETN (AAM only)	≡GN≡
				47	WE WEOR	≡WE≡
				10	RE REWINDM	≡RE≡

TABLE D-1. STRUCTURE OF THE FIT (Cont)

Word	Bits	Fit Field	Description	Contents	COMPASS Symbol
				11 GL GETL PL PUTL	≡GL≡ ≡PL≡
				12 SB SKIPB	≡SB≡
				13 CL CLOSEL	≡CL≡
				63 WK WTMK	≡WK≡
				74 EN ENDFILE	≡EN≡
	35	WPN	Write bit. The upper bit of LOP is a 1-bit subfield that can be accessed separately. If the last operation was a write, it is set. This field cannot be changed by the user.		
	29-0	RC	Record count. Count of full records read or written since the file was opened. The count is not adjusted for repositioning and backspacing operations. This field cannot be changed by the user.		
19	59-36	MBL	Maximum block length in characters.		
	35-30	VNO	Current volume number of the multi-volume sequential file.		
		NL	Number of index levels of blocks (AAM only).		
	29-0	BN	Block number of the current block (sequential files); cannot be changed by the user.		
		WA	Current position word address, set by GET and PUT macros.		
20	59	BCK	Block checksums (AAM only).		
	58	PM	Processing mode (AAM only).		
	57-52	POS	Duplicate key position (AAM only).		
	51-30	DCT	Address of the display code to collating sequence conversion table (AAM only).		
	59-36	MNB	Minimum block length in characters.		
	29-18	RB	Number of records per K type block in sequential files.		
	17-0	PKA	Primary key address (AAM only).		

TABLE D-1. STRUCTURE OF THE FIT (Cont)

Word	Bits	Fit Field	Description	Contents	COMPASS Symbol
21	59-18	XN	Logical file name of the alternate key index file associated with the data file (AAM only).		
	17-0	XBS	Index file block size (AAM only).		
	59-24	MFN	Multifile set name.		
	23-0	PNO	Multifile position number; position number of member file on multifile set.		
	17-16	OVF	Direct access file overflow flag (AAM only).		
	11-0	KR	Key value repeat count (AAM only).		
22	59-46		Reserved for CRM.		
	45-40	LAC	Last action performed on the file; used by compiler languages to communicate with each other.		
	39-36	LNG	Last compiler language that used the file.	0 Unknown 1 FORTRAN 2 COBOL 3 PL/I 4-7 Reserved	
	35-0		Reserved for CRM.		
23			Reserved for CRM.		
24	59	NDX	Index flag (AAM only).		
	58	KNE	Key not equal (AAM only).		
	57	FWI	Forced write indicator (AAM only).		
	56	FPB	File position bit (system routine use only).	0 EOI not reached. 1 EOI reached.	
	55	ON	Old or new indexed sequential, direct access, or actual key file (AAM only).		
	54		Reserved for CRM.		
	53-24	FLM	File limit, records per file (AAM only).		
	23	EMK	Embedded key flag (AAM only).		
	22	DKI	Duplicate key indicator (AAM only).		
	21-0	KA	Key address (AAM only).		

TABLE D-1. STRUCTURE OF THE FIT (Cont)

Word	Bits	Fit Field	Description	Contents	COMPASS Symbol
25	59-18		Reserved for CRM.		
	17-0	BZF	Busy FET address; cannot be changed by the user.		
26	59-48		Reserved for CRM.		
	47-30	CDT	Address of the collating sequence to display code conversion table (AAM only).		
	29-0		Reserved for CRM.		
27-29			Reserved for CRM.		
30	59	SOL	S/L tape bit; cannot be changed by the user.		
	58-30		Reserved for CRM.		
	20-0	EOIWA	Word address at EOI for word addressable files.		
31	59-48	RKW	Relative key word (AAM only).		
	47-44	RKP	Relative key position in RKW (AAM only).		
	43-40	KP	Beginning character position of the key (AAM only).		
	39-31	KL	Key length in characters (AAM only).		
			Key length in bits (AAM only).		
			Primary or alternate key length (AAM only).		
	30-24	IP	Index block padding percent (AAM only).		
	23-0		Reserved for CRM.		
32	59-42	IBL	Index block length in characters (AAM only).		
	41-30		Reserved for CRM.		
	29-27	KT	Key type (AAM only).		
	26-24	REL	File position key relation (AAM only).		
	23-18	TRC	Trace transaction count; number of transactions to be traced (AAM only).		
	17-0	CPA	Compression routine address (AAM only).		
33	59-18		Reserved for CRM.		
	17-0	DCA	Decompression routine address (AAM only).		
34			Reserved for CRM.		

TABLE D-2. ALPHABETIZED SUMMARY OF FIT FIELDS

FIT Field	Word	FIT Field	Word	FIT Field	Word
ASCII	5	FP	10	OF	11
BAL	15	FPB	24	ON	24
BBH	17	FWB	1	ORG	15
BCK	20	FWI	24	OVF	21
BFS	17	HB	16	PC	16
BN	19	HL	15	PD	17
BT	11	HMB	18	PDF	15
BZF	25	HRL	16	PEF	13
B8F	17	IBL	32	PKA	20
CDT	26	IP	31	PM	20
CF	11	KA	24	PNO	21
CL	16	KL	31	POS	20
CM	11	KNE	24	PTL	18
CMPLT	0	KP	31	RB	20
CNF	17	KR	21	RC	18
CP	17	KT	32	RDR	1
CPA	32	LA	10	REL	32
C1	17	LAC	22	RKP	31
DC	1	LBL	10	RKW	31
DCA	33	LCR	10	RL	11
DCT	20	LFN	0	RMK	16
DFC	13	LL	16	RT	11
DKI	24	LNG	22	SB	17
DP	16	LOP	18	SBF	15
DVT	1	LP	17	SES	13
DX	12	LT	10	SOL	30
ECT	13	LX	11	SPR	15
EFC	13	MBL	19	STFT	15
EMK	24	MFN	21	TL	16
EO	15	MKL	16	TRC	32
EOIWA	30	MNB	20	ULP	10
ERL	13	MNR	15	VF	11
ES	13	MRL	12	VNO	19
EX	13	MUL	16	WA	19
FF	1	NDX	24	WPN	18
FL	12	NL	19	WSA	15
FLM	24	NOFCP	13	XBS	21
FNF	17	OC	17	XN	21
FO	11				

LOADING BAM

In order to reduce field length, BAM has been divided into functional capsules which are loaded by relocatable control routines at execution time. This method of dynamic loading requires a program to be compatible with Common Memory Manager (CMM). Static loading is available for programs that are not compatible; however, static loading could involve a field length penalty of as much as 1400_8 words. Unless static loading is specified, BAM uses dynamic loading.

More information about Common Memory Manager and the CYBER Loader can be obtained from their respective reference manuals.

DYNAMIC LOADING

For dynamic loading, all macros reference entry points in the controlling routines. The controlling routines, which process parameters and diagnose certain types of errors, are loaded at relocatable load time or overlay generation time. The controlling routines load and transfer control to the Fast Dynamic Loader (FDL) capsule needed to process the macro in fixed-position fixed-length blocks.

It is important to the dynamic loading scheme that the controlling routines not be overlayed. Unknown results, including bad jump addresses to service routines, result if these routines are overlayed. To prevent the controlling routines from being overwritten, they must be part of the (0,0) overlay.

The OPENM/SETFIT capsule is loaded when the first OPENM or SETFIT macro is encountered. If the SETFIT macro is encountered first, the FILE control statement parameters are processed, buffer size is calculated, and control is returned to the user.

When the OPENM macro is encountered, the SETFIT functions are performed if there has not been a previous SETFIT macro. OPENM processing then occurs. The file is opened, FIT consistency checks are performed, label processing occurs, and control is returned to the user. If label processing is required, the controlling routine loads the GETL/PUTL capsule when the first GETL or PUTL macro is encountered. The open and label processing capsules are unloaded when a macro other than OPENM, SETFIT, GETL, PUTL, STORE, or FETCH is encountered. Therefore, for optimum efficiency in loading, the open processing for all files should be completed before other processing is specified.

When the first macro is encountered that requires a buffer, a buffer is allocated through CMM in a fixed-position fixed-length block. If the buffer below highest high address (BBH) field of the FIT is set to YES, CMM is requested to allocate the buffer below the highest high address (HHA). The HHA is the end of the longest overlay. If the BBH field is set to YES, the file must be closed with the CF field set to U, RET, or DET before another overlay is loaded. If the BBH field is set using the FILE macro, references are issued to the additional CMM routines necessary to process this feature. However, if the BBH field is set using the STORE macro, the FILE control statement, or some other means, the user must reference the additional CMM routines. This can be done by using either the COMPASS LDSET pseudo-op or the LDSET control statement as follows:

```
LDSET USE=CMM.AGR
LDSET(USE=$CMM.AGR$...)
```

The capsules required to perform the function specified by the macro are then loaded; control transfers to the capsules and back to the user. Except for the SKIP capsules, the capsules required to process these types of functions remain in memory until all files requiring them have been closed. The capsules required for SKIP are loaded while a series of skips is being performed and unloaded when a macro other than SKIP is encountered.

The CLOSEM capsule is loaded when the CLOSEM macro is encountered. It closes the file and buffer space is released if the CF field is set to U, RET, or DET; this must be specified if the BBH field is set to YES. The CLOSEM capsule unloads any capsules no longer needed for processing and unloads itself after it closes the last file.

STATIC LOADING

Static loading is provided in cases where the user is managing memory. It should only be used as a short term conversion aid. Long term support of this feature is not to be provided. There are two methods for designating which capsules need to be statically loaded; one is control statement oriented, and one is macro oriented.

STATIC LOADING WITH CONTROL STATEMENTS

To specify static loading with control statements, the option STAT must be specified on the LDSET control statement; the USE and OMIT parameters must be specified on the FILE control statement. A FILE control statement must be used for each file to insure that all necessary routines are loaded. The FO, RT, and BT parameters must be specified on a previous FILE control statement or on the same FILE control statement as the USE and OMIT parameters. They cannot be specified on a FILE control statement following the FILE control statement which specified the USE and OMIT parameters.

The formats of the USE and OMIT parameters are:

- USE=mn_1/mn_2/.../mn_n
- OMIT=mn_1/mn_2/.../mn_n

where mn is a macro name. Terminal users must use TGET and TPUT to load special terminal I/O capsules. The functions of the USE and OMIT parameters are listed in table E-1. The USE and OMIT parameters can be used on more than one FILE control statement for one file; the result is cumulative. If the STAT option is specified on the LDSET control statement and no USE parameter is specified on the FILE control statement, no functions are loaded.

TABLE E-1. USE AND OMIT PARAMETER FUNCTIONS

Parameter	No list of Macros	List of Macros
USE	All capsules are loaded.	Capsules performing functions specified by the macro list are loaded.
OMIT	All previously loaded capsules are removed.	Capsules performing functions specified by the macro list are removed.

In the example shown in figure E-1, the program to write the file ATAPE uses static loading and contains the macros OPENM, PUT, CLOSEM, and ENDFILE. The program to read the file ATAPE also uses static loading. The macros PUT and ENDFILE are not contained in that program; the OMIT parameter specifies that those capsules are not to be loaded. The GET macro is contained in the program, and the capsule for that macro is to be loaded. The USE parameter is still in effect for the macros OPENM and CLOSEM.

```
    .
    .
    .
FILE(ATAPE,FO=SQ,RT=Z,BT=C,USE=OPENM/PUT/
    CLOSEM/ENDFILE)
LDSET(STAT=ATAPE)
Load set to write file.
FILE(ATAPE,OMIT=PUT/ENDFILE,USE=GET)
LDSET(STAT=ATAPE)
Load set to read file.
    .
    .
    .
```

Figure E-1. Static Loading Example

STATIC LOADING WITH THE STLD.RM MACRO

The STLD.RM macro is another method of specifying static loading. (The LDST.RM macro, which was valid in CYBER Record Manager Version 1.4 is treated as a no-op.) The format of the STLD.RM macro is shown in figure E-2. It must be specified once for each file organization.

```
(fo)        STLD.RM   USERT=(rt1,rt2, . . . ,rtn),
                      USEBT=(bt1,bt2, . . . ,btn),
                      USE=(mn1,mn2, . . . , mnn),
                      OMIT=(CMM or FDL)

fo          File organization.

rt          Record types of files.

bt          Block types of files.

mn          Macros used in program.

CMM or FDL  CMM omits CMM and FDL.
            FDL omits FDL.
```

Figure E-2. STLD.RM Macro Format

USE OF LIST-OF-FILES F

The NOS and NOS/BE operating systems maintain a pointer to the list-of-files, which is a table of the names and FIT addresses of all active files for each control point. This pointer is set and accessed by the SETLOF and GETLOF macros. A complete description of this feature can be found in volume 2 of the NOS reference manual and the System Programmer's Reference Manual for the NOS/BE operating system.

BAM maintains and uses this list-of-files. To alter this list, a user must follow a procedure that is compatible with BAM.

BAM maintains an entry point in its relocatably loaded routines called LOF$RM. The content of this entry point is the address of the current list-of-files. The purpose of this pointer is to minimize the number of GETLOF monitor calls required. The user is encouraged to use this pointer instead of calling the GETLOF macro.

If a user program that coexists with BAM moves the list-of-files, it must update the LOF$RM pointer in addition to calling the SETLOF macro. Also, if a user program adds a new entry to the end of the list-of-files, it must insure that the next word is zero because BAM does not initialize the list-of-files block to zero.

For interactive jobs, BAM puts the file that it uses for output to connected files (ZZZZZOU) in the first word of the list-of-files table. This is a requirement of the NOS operating system. If a file name is put in the first word of the list, the user cannot depend on that name remaining in the first word. If a user program uses BAM through a terminal under the NOS operating system, it cannot write to a terminal file that is not a BAM file in the same job step. The user program cannot move or destroy the ZZZZZOU entry in the first word of the list-of-files.

FILE INTERCHANGEABILITY

The following tape formats are interchangeable between 7000 Record Manager and BAM:

- Binary files having S type records, or Z type records and C type blocks, are interchangeable, provided the value of the maximum block length (MBL) field is 5120.

- Files having W, F, U, D, T, R, or Z type records and I, C, K, or E type blocks are interchangeable (except for Z type records with C type blocks), provided such files are accessed via BAM on S/L devices.

The file formats that are not interchangeable are as follows:

- 7000 Record Manager does not read 7-track coded Z type record tapes.

- 7000 Record Manager does not read L tapes having a block length greater than the individual station limits.

- 7000 Record Manager does not correctly read a file having W, F, U, D, T, or R type records recorded on other than S or L tapes.

- 7000 Record Manager requires macro parameters placed in registers to be in X registers; BAM macro parameters can be in any user registers.

- BAM does not read a tape file with C or I type blocks if the value of the MBL field is not equal to 5120.

- BAM does not read a tape having embedded tapemarks. (WTMK under 7000 Record Manager does write a tapemark rather than a level 17_8 on a file with S type records or with Z type records and C type blocks. For interchangeability, use of WTMK is not recommended; the ENDFILE macro should be used instead.)

- BAM does not read other than an L tape if the value of the MBL field is other than 5120.

Refer to the table on labeling conventions (section 6) for additional information on labels.

SYSTEM MIGRATION GUIDELINES H

This appendix contains programming practices recommended by CDC for users of the software described in this manual. When possible, application programs based on this software should be designed and coded in conformance with these recommendations.

Two forms of guidelines are given. The general guidelines minimize application program dependence on the specific characteristics of a hardware system. The feature use guidelines ensure the easiest migration of an application program to future hardware or software systems.

GENERAL GUIDELINES

Good programming techniques always include the following practices to avoid hardware dependency:

- Avoid programming with hardcoded constants. Manipulation of data should never depend on the occurrence of a type of data in a fixed multiple such as 6, 10, or 60.

- Do not manipulate data based on the binary representation of that data. Characters should be manipulated as characters, rather than as octal display-coded values or as 6-bit binary digits. Numbers should be manipulated as numeric data of a known type, rather than as binary patterns within a central memory word.

- Do not identify or classify information based on the location of a specific value within a specific set of central memory word bits.

- Avoid using COMPASS in application programs. COMPASS and other machine-dependent languages can complicate migration to future hardware or software systems. Migration is restricted by continued use of COMPASS for stand-alone programs, by COMPASS subroutines embedded in programs using high-level languages, and by COMPASS owncode routines used with CDC standard products. COMPASS should only be used to create part or all of an application program when the function cannot be performed in a high-level language or when execution efficiency is more important than any other consideration.

FEATURE USE GUIDELINES

The recommendations in the remainder of this appendix ensure the easiest migration of an application program for use on future hardware or software systems. These recommendations are based on known or anticipated changes in the hardware or software system, or comply with proposed new industry standards or proposed changes to existing industry standards.

BASIC ACCESS METHODS

The Basic Access Methods (BAM) offer several features within which choices must be made. The following paragraphs indicate preferred usage.

File Organizations

The recommended file organization is sequential (SQ). For files with word-addressable (WA) organization, use an accessing technique that can be easily modified to byte addresses.

Block Types

The recommended block type is C.

Record Types

The recommended record types are F for fixed length records and W for variable length records. For purely coded files that are to be listed, Z type records can be used.

Block Size

Set the maximum block length (MBL) to 640 characters for mass storage files and 5120 characters for tape files.

Host Language Input/Output

Use of host language input/output statements (for example, a FORTRAN READ statement) to process BAM files is always a safe procedure. Host language statements provide appropriate default values for record type, block type, and block size. Do not use the CYBER Record Manager FORTRAN interface routines to process sequential files.

INDEX

COMMENT SHEET

MANUAL TITLE: CYBER Record Manager Basic Access Methods Version 1.5
Reference Manual

PUBLICATION NO.: 60495700

REVISION: G

This form is not intended to be used as an order blank. Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments on the back (please include page number references).

_____ Please reply _____ No reply necessary

FOLD FOLD

CUT ALONG LINE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
Publications and Graphics Division
P.O. BOX 3492
Sunnyvale, California 94088-3492

FOLD FOLD

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.
FOLD ON DOTTED LINES AND TAPE

NAME:

COMPANY:

STREET ADDRESS:

CITY/STATE/ZIP:

TAPE TAPE