IDENTIFICATION

Product Code:	DEC-12-ZR8A-D
Product Name:	DIAL-MS File Commands Program Description
Date Created:	July 1, 1970
Maintainer:	Software Services

LAP6-DIAL is an editor, filing system and assembler for use with the PDP-12 computer. The editor and filing portions are derived from the basic LINC program LAP6[1] by Mary Allen Wilkes of Washington University. The assembly portion is derived from several programs used for the PDP-8 computer including PAL-D[2].

The Digital Equipment Corporation wishes to express to the author, Mary Allen Wilkes (Clark), and the Computer Research Laboratory of Washington University, St. Louis, Missouri, its appreciation for the development set forth in LAP6 as well as its thanks for permission to use parts of the LAP6 program.

---

[1] M. A. Wilkes, LAP6 Handbook, Computer Research Laboratory Tech. Rep. No. 2, Washington University, St. Louis, May 1, 1967.

[2] PAL-D Assembler Programmer's Reference Manual DEC-D8-ASAA-D.

1.0 PROGRAM OVERVIEW

File Commands, referred to as FILECOMS, is comprised of a series of file handling routines, along with Save Program, Add Binary, and Save Binary in its entirety. FILECOMS is called by the Editor whenever a Print Source, Add Binary, Save Program, or Save Binary is requested, or whenever an assembly is requested of a source file. For any of the above cases, FILECOMS uses the MC parameter table and the index to set up the required arguments. When handling files and setting up arguments, FILECOMS checks to see if the request is possible and/or legitimate. If a command cannot or will not be executed, FILECOMS displays a "NO". Once FILECOMS sets up arguments, control is turned over to the appropriate program.

File Commands is on TBLKs 35Ø, 351, 352, and 353 and in MBLKs 0, 1, 2, and 3 of segment 2 when in core. The Editor communicates with FILECOMS relative to the tag FCSA (FILECOMS Starting Address). To execute a Save Program, for example, the Editor reads in the first block of FILECOMS and executes a JMP FCSA+3. The contents of FCSA+3 are JMP SAVPRG. The FCSA table, which must not be moved without the Editor's knowledge, has the following form:

```
(4Ø2Ø)  FCSA,JMP SAVBIN        /SAVE BINARY
         JMP ADDBIN
         JMP .
         JMP SAVPRG        /SAVE PROGRAM
         JMP LIORAS        /ASSEMBLER
         JMP PRNPRG        /PRINT SOURCE
```

At location RPLSTP is the "REPLACE?" table which contains the DIAL codes for the phrase "REPLACE?". The "REPLACE?" table is used in conjunction with the Editor's grid pattern display table (A6) to display the phrase "REPLACE?". Because FILECOMS uses the file area extensively, an explanation of index structure is crucial to understanding the secrets of FILECOMS.

The index is located at tape blocks 346 and 347; the LAP6-DIAL system expects it to be two blocks long. The index must be comprised of contiguous tape blocks and the first tape block must be of the form XX6. An index tape block of 352, for example, would not be compatible with a number of LINCtape group instructions that are scattered throughout the system. The left half of the first word on tape block 346 must be the DIAL code for a slash (57), in order to identify the block as an index. The first 10 words of tape block 346 contain 5757 (//) and are never used for file name storage by the DIAL system; the remaining words on both tape blocks are used to store index information. Each file in the index takes up 10 words, so that a DIAL tape can hold a maximum of 77 octal file names. Unfilled portions of the index contain 5757 in all 10 words. For any given 10 word portion, the first four words are reserved for the file name. The DIAL code for the first character of the name is stored in the

left half of the first word; the second character of the name goes in the right half of the first word, etc. The maximum number of characters in a name is eight. Unused half-words in a name = 77 (?). The name FILENAME would be stored in the following manner:

```
Ø611      FI
14Ø5      LE
16Ø1      NA
15Ø5      ME
```

The name DIAL would be saved as:

```
Ø411      DI
Ø114      AL
7777      (Not
7777      Used)
```

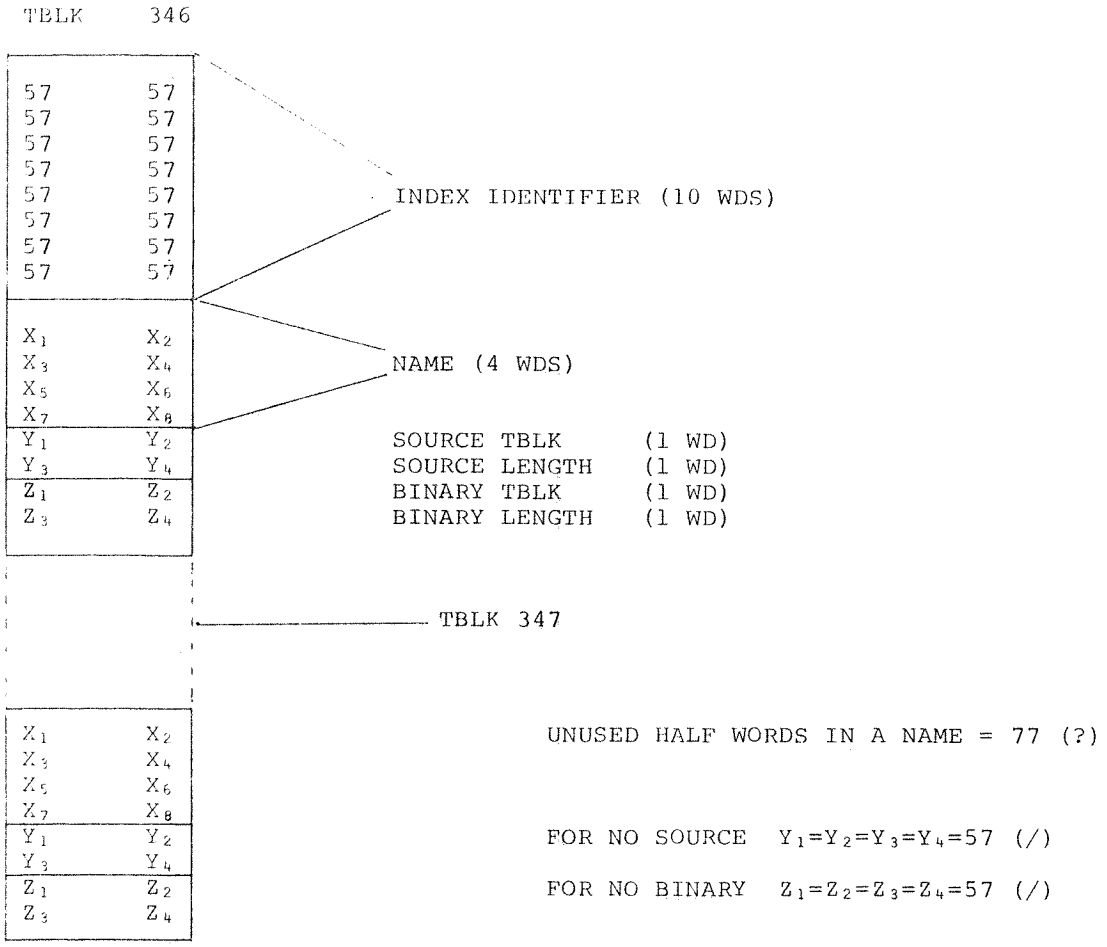
Question marks within a name are treated as ordinary characters. Leading question marks are illegal and may cause unpredictable results. Trailing question marks are ignored.

If a file name refers to a source, the fifth word of the ten word sector holds the first TBLK of the source and the sixth word holds the length. If there is no source, then the fifth and sixth words are both equal to 5757. If a name refers to a binary file, then the seventh word contains the first TBLK of the binary and the eighth word contains its length. If there is no binary, then the seventh and eighth words are both equal to 5757. In words 5, 6, 7, and 10, the low order 9 bits are reserved for tape block and length; bit zero is used to indicate whether source or binary information is present; bits 1 and 2 are not used.

2.0 SAVE PROGRAM AND FILECOMS

When a user requests a Save Program, the Editor exits, reads four tape blocks of FILECOMS (35Ø) into locations 4ØØØ-5777, loads the last tape block used in the Working Area into the AC, and jumps to FCSA+3. Save Program first stores away the complement of the source length at location F1, then checks for line number arguments, displaying "NO" if they have been requested. Next, Save Program reads the index (TBLKs 346, 347) into locations 3ØØØ-3777. At this point, Save Program checks the left half of the first word in the index for a 57; if there is no 57, Save Program fills locations 3ØØØ-3777 with 5757. This move is called making an index. Save Program then calls a subroutine called LOOKUP to check for a name match between the requested name - to be found in the MC parameter table at E6 + 2 to E6 + 5 (2372-2376) - and the index. If there is no name match, Save Program searches the index for a blank entry (a 57 in the left half of the first word). If no blank entry is found, the index is full and Save

INDEX STRUCTURE



UNUSED HALF WORDS IN A NAME = 77 (?)

FOR NO SOURCE $Y_1=Y_2=Y_3=Y_4=57$ (/)

FOR NO BINARY $Z_1=Z_2=Z_3=Z_4=57$ (/)

UNFILLED PORTION OF THE INDEX
5757 IN ALL WORDS

Program displays "NO". If a blank entry is found, locations E6 + 2 to E6 + 5 are placed into its first four words and the gap search commences. If a name match is found, Save Program must determine whether or not the matched entry has an accompanying source (the match could be binary only). To determine source or binary match, Save Program checks bit Ø of the fifth word in the matched entry. If the match is binary only, the gap search commences. If a requested name matches a source, Save Program displays the phrase "REPLACE?" until a key is struck on the Teletype. If the key struck on the Teletype is not an "R", Save Program returns to the Editor; if an R is typed, 5757 is put into the fifth and sixth words of the matched entry to erase the starting TBLK and length information of the old source. Save Program then calls the gap search subroutines.

Once the index has been properly tended to, Save Program must find room in the file area to store the source. There are 600 TBLKs on a DIAL tape that are available for storage. These 600 TBLKs are divided into two sections:

1. The upper file area (TBLKs 47Ø-777)
2. The lower file area (TBLKs Ø-267)

Sources are saved in the file area as close to the index as possible.

The gap search subroutines first scan all upper file entries in the index to determine where the gaps are in the upper file. A gap is any unused portion of the file area. If a file named ONE runs from TBLKs 47Ø-517 and another file named TWO runs from TBLKs 55Ø-6ØØ, then a 30 TBLK gap exists from TBLKs 52Ø-547. If there are no files stored in the upper file area, then there exists a 310 TBLK gap from TBLKs 47Ø-777. The gap search subroutines check words 5-10 of all index entries to compute the file gaps. The best TBLK in the upper file area is 47Ø because it is closest to the index. The gap search routines initially set the best TBLK at 47Ø; then they start searching for gaps. If a gap is found that is smaller than the length of the requested entry, the best TBLK is reset to one greater than the last TBLK of the current entry. For example, if the best TBLK is set to 47Ø and the length of the requested entry is 20 TBLKs and the gap search routines find a source entry (or binary, for that matter) starting in TBLK 5ØØ (word 5) that is fifty TBLKs long (word 6); i.e., the source in question resides in TBLKs 500-547 inclusive, then the gap is from TBLKs 47Ø-477 (10 TBLKs). This is too small to accommodate an entry of 20 TBLKs, so the best TBLK is reset to 55Ø, which is one more than the last TBLK (547) of the current entry. The gap search subroutines scan the index, resetting the best TBLK every time a gap that is too small is encountered.

Once the upper file has been searched, the entire sequence is repeated for the lower file. When searching the lower file, the best TBLK is set to one less than the start of the current entry when an insufficiently long gap is

encountered. Searching the upper and lower files in the manner described above really finds all those TBLKs that will not work. For example, suppose that the best TBLK in the lower file turned out to be 4Ø; this would mean that there are no gaps between TBLKs 41-267 that are large enough to file the requested source.

Once the best TBLKs for upper and lower file have been established by the gap search routines, Save Program must determine whether or not the requested source will fit in the file area. If the length of the requested source is greater than the difference between the lower bound of the lower file (TBLK Ø) and the best TBLK of the lower file, and if it is also greater than the difference between the upper bound of the upper file (TBLK 777) and the best TBLK of the upper file, then the requested source is bigger than all gaps in the file area and it will not fit. For this case, Save Program displays "NO". If the requested source will fit in either the upper or lower file area, Save Program determines which of the two best TBLKs is closest to the index before initiating the actual save.

After file area has been allocated by Save Program, the starting tape block of the requested source is stored in the fifth word of the entry and the source length is stored in the sixth word. Save Program then checks the keyboard. If a key has been struck, the request is inhibited and control is turned over to the Editor; if no key has been struck, the Save Program part of Save Program commences.

Save Program transfers 16 (octal) TBLKs at a time from the working area to the file area until the entire source has been saved. The last transfer may be from 1 to 16 TBLKs, depending on the length of the source being saved.

3.0 PRINT SOURCE AND FILECOMS

Whenever a Print Source is requested, FILECOMS must be called ahead of Print Source to set up arguments for Print Source. The Editor executes this call by reading TBLKs 35Ø-353 of FILECOMS into locations 4ØØØ-5777 and doing a JMP FCSA+5. FILECOMS first checks the left half of E6 + 2 (2373) in the MC parameter table for a 77 (?). A 77 indicates that the Working Area is to be printed; for this case FILECOMS reads the TBLKs containing Print Source (363) and TTY (364) into locations 5ØØØ-5777, moves the source working area unit to E6 + 6, and executes a JMP 1ØØØ. If the Print Source is by file name, FILECOMS checks to see if the requested unit has an index, displaying "NO" if it does not. If the requested unit has an index, FILECOMS calls the routine LOOKUP to check for a name match. If there is no name match, FILECOMS displays "NO". If there is a name match, FILECOMS further checks to see if the name refers to binary only, and if so displays a "NO". If the name match is legal, FILECOMS reads Print Source and TTY into locations 5ØØØ-5777, retrieves the first TBLK of the source

from the index and puts it into AC. FILECOMS then calls Print Source by executing a JMP 1ØØØ.

4.0 THE ASSEMBLER AND FILECOMS

Whenever an assembly is requested of a source in the file area, FILECOMS must be called to set up arguments for the Assembler. The Editor executes this call by reading TBLKs 35Ø-353 into locations 4ØØØ-5777 and doing a JMP FCSA + 4. The index routines are next called by FILECOMS and executed in exactly the same manner as if a Print Source had been requested. FILECOMS places the unit number at location UNITNO (4777), places the QL or LI word at location UNITNO-3 (4774), and puts the starting TBLK of the requested source in the AC. FILECOMS then recalls the Editor to set up further arguments for the Assembler, read it in and start it up.

5.0 SAVE BINARY AND FILECOMS

When a Save Binary is requested, the Editor exits, loads four blocks of FILECOMS into 4ØØØ-5777, and jumps to FCSA, File Commands starting address. Save Binary reads the binary header block (57 in the binary working area) into 54ØØ-5777.

E6, the MC parameter table, is examined to determine the starting mode and address. If none were specified (→SB FILENAME,U↵), E6 contains zero. A LINC-mode halt is stored in the header, and Save Binary jumps to SBØ2Ø to find space for the file. If 8-mode was specified, E6 contains 44ØØ plus the PDP-8 field in which the program is to start. E6+1 contains the 12-bit address within that field. The following instruction sequence is stored in the header:

```
PDP
CIF   X              /WHERE X IS Ø OR 1Ø, DEPENDING ON
                     /THE FIELD SPECIFIED
JMP I 377            /START THE PROGRAM
START                /STARTING ADDRESS FROM E6+1
```

If LINCmode was specified, E6 contains Ø4ØØ plus the starting PDP-8 field, and E6+1 contains the 12-bit address within the field. The following instruction sequence is built in the header:

```
LIF X                /X IS 0-7, DEPENDING ON THE SEGMENT
                     /SPECIFIED.
JMP  START           /START IS THE 10 LOW-ORDER BITS
                     /FROM E6+1.
```

The default starting addresses are field Ø, location 2ØØ for PDP-8 mode and segment 2, location 2Ø for LINCmode. These are set up by the Editor before

calling FILECOMS, if a starting address was not explicitly requested.

After building these instructions in the header, the starting address is checked for validity. It must be less than 2ØØØØ (8K), and in a block which contains assembled code.

Save Binary next obtains the number of binary blocks (location 337 within the header) from the header. One is added to this number, to allow for the header, and the result complemented and stored at F1. This is used as the length of the desired file during the search for file space. The two's complement of the number of binary blocks is calculated and stored at MBCNT, as a loop control during copying. Save Binary then follows the same sequence as Save Program to read the index, find space for the file, and write the updated index. Next, the I/O control blocks are initialized, an auto-index register is set up for the block map in the header, and copying begins.

Each word of the block map is tested for zero. If it is zero, the corresponding block in the working area is not used, so the input block number is incremented and the loop restarted. If the word is not zero, the corresponding block is read, and MBCNT is tested to see whether it is the last binary block. If it is the last, Save Binary computes the number of blocks currently in the buffer, and goes to SB13Ø to write them. If not the last block, Save Binary increments the input buffer address, and checks to see whether the buffers are full. If not, the input block number is incremented and the loop re-entered. If the buffers are full, or the last block has been read, HDRSW is tested to determine whether the header block has been written. If not, it is written at this time, and the switch is set. Then the buffers are written, and MBCNT is tested to check for completion. If there is more to go, the I/O control blocks are reinitialized and the loop is re-entered. Otherwise, the Editor is called by jumping to 7777 in field 1.

6.0 ADD BINARY AND FILECOMS

When Add Binary is requested, the Editor exits and loads blocks 35Ø-353 into locations 4ØØØ-5777. It then jumps to FCSA+1. Add Binary reads the index of the requested file, and finds the binary requested, displaying "NO" if unsuccessful.

The header and up to 15 blocks of the binary file are read, as well as the header of the binary working area.

The header from the file is scanned by subroutine SCANHD to determine the memory address for which the first block was assembled. If a relocation (other than zero) was specified, it is stored as the new memory address for the program.

The binary data from the file is scanned, and non-zero words moved to the buffer for the binary working area.

When the end of an input block is reached, a running block count (MBLKS) is incremented and checked for end of file. If the end has been reached, the final binary working area block is written, the binary header is written, and control returns to the Editor.

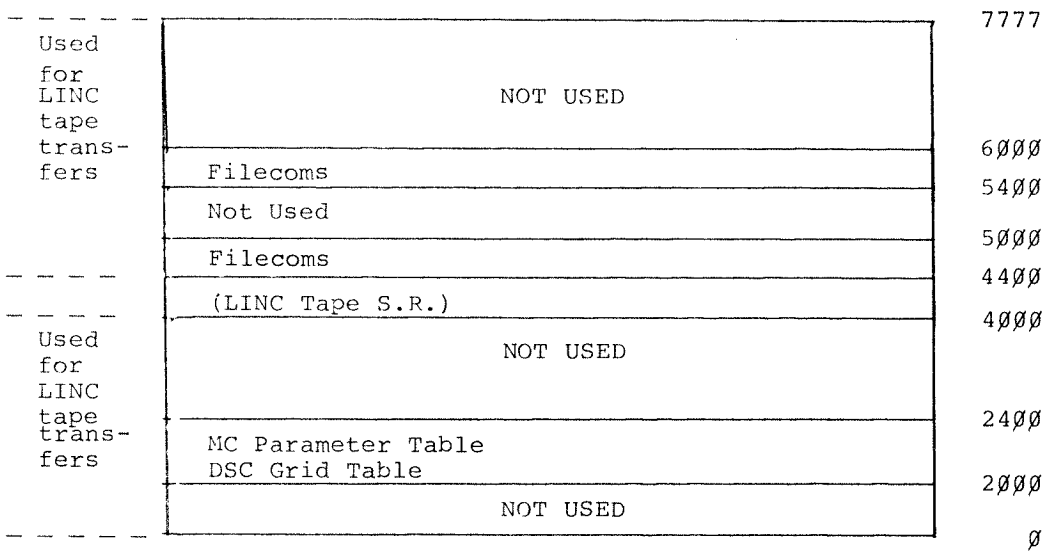
If there remains more binary to add, the header is scanned so that the relocation can be adjusted for skipped blocks and, if the end of the buffer has been reached, the buffer is refilled with up to 15 blocks from the file.

When a non-zero word is found which goes into another block of the binary working area (other than the block currently in core), the old block is written, the header updates, and, if the header indicates that the new block is used, the new block is read in. If there is nothing in the new block, the buffer is cleared.

7.0 FLOW DIAGRAMS (Attached)

8.0 PROGRAM LISTING (Attached)

9.0 MEMORY MAP



ADVIND - The first core location of the subroutine that advances the index pointer (BETA 2) by 10 words at a time and checks for end of index.

AUTO - 8-mode auto-index register (10) used by SB.

A6 - The first core location of the Editor's DSC grid table.

BFAD - The location containing buffer address, shifted right 8.

BFLN - The location containing the length of buffer, in blocks.

BUFFER - The buffer address, shifted right 8.

BUFLEN - The length of the buffer, in blocks.

BUMP - The location containing the increment for index pointer to address desired (source or binary) pointers.

BWA - The pseudo-unit for binary working area.

CHKIND - The first core location of the subroutine that reads in and checks for the validity of an index.

COMPBN - The first core location of the subroutine that compares TBLKs in the index with the prospective best TBLK and sets the best TBLK during the gap search for Save Program.

CR1 - The symbolic code for carriage return; it is equal to 43ØØ and is used in conjunction with the SHD I instruction.

DATSEG - The data segment used throughout FILECOMS. It contains A6, E6, and Index.

DIALU - The pseudo-unit that contains the resident DIAL system (1ØØ).

DRSTRT - The address of DIAL bootstrap routine.

EDRTN - The start of routine to call DIAL bootstrap.

ENTER - The first core location of the subroutine that searches for a blank slot in the index during a Save Program.

E6 - The first core location of the MC parameter table.

FCSA - The first core location of the FILECOMS starting address table.

FILE - The first TBLK in the upper file area above the index that is available for file storage.

FINDSP - The first core location of the subroutine that sets up arguments for upper and lower file during a gap search.

FREE - The first TBLK+1 in the lower file area below the index that is available for file storage (also the first TBLK of the free area on the DIAL tape).

F1 - The core location that holds the one's complement of the length of a source during a Save Program, or one's complement of the length of the binary (including header) during a Save Binary.

GAPSR - The first core location of the main gap search subroutine which is called during a Save Program and Save Binary.

GETBN - The first core location in the subroutine that sets up the starting TBLK and source length during a gap search.

GETPS - The starting point of code which reads in and calls Print Source.

G1 - The first core location of the subroutine that determines the nearest of the two best TBLKs in the upper and lower file area; it is called during a Save Program.

HDRBLK - The TBLK of binary header, relative to binary working area.

HDRIO - The control block for reading and writing header.

HDRSW - The switch is 7777 if the header has not been written; Ø after writing.

HEADER - The core location in the current segment which contains the header.

HI8FLD - The field containing I/O routines and bootstrap.

H1 - The first core location of the subroutine that is called during a Save Program to determine if the best TBLKs will accommodate the length of the requested source.

INDEX - The core location of the index (3ØØØ).

INFIL - The control block used in reading files to be copied.

KBDOPR - The core location in the Editor relative to which FILECOMS effects its return to the Editor after an assembly by file name has been requested.

KBDSEG - The segment containing KBDOPR.

K2 - A tag in the subroutine REPLAC; tests for response from user.

LIORAS - The first core location of the main subroutine that is called by FILECOMS during an assembly by file name.

LOOKUP - The first core location of the subroutine that searches the index for a name match.

LO8FLD - The field in which FILECOMS resides.

MAKIND - The first core location of the subroutine that makes an index.

MAP - The core location in the current segment containing the block map from header.

MBCNT - The location containing minus block count: two's complement of number of blocks to read from working area.

MBFAD - The location containing the minus buffer address, shifted right 8.

MBFEND - The location containing minus (buffer end + 1), shifted right 8.

OUTFIL - The control block for output file during copies.

PDPMOD - The location of the routine for setting up PDP-8 mode start in Save Binary.

PRNPRG - The first core location of the main subroutine that is called by FILECOMS during a Print Source.

PSBLK - The first TBLK of Print Source (relative to DIAL).

PSENT - The entry point of Print Source main code.

PSIN - The control block for reading Print Source.

PSWA - The routine which sets up for printing the source working area.

P1 - Constant ØØØ1.

P2 - Constant ØØØ2.

READ - The page Ø location pointing to Read routine.

REPLAC - The first core location of the subroutine that displays the phrase "REPLACE?" on the scope.

RPLSTR - The first core location of the replace table.

RSTRX - The first core location of the subroutine that writes out the index.

SAVBIN - The first location of Save Binary routine.

SAVPRG - The first location of Save Program routine.

SAYNO - The first core location of the subroutine that displays the word "NO" on the scope.

SBØ1Ø -
SBØ2Ø -
SBØ3Ø -
SBØ4Ø -
SB1ØØ -
SB11Ø -
SB12Ø -
SB13Ø -
SB14Ø -
SB16Ø -

Tags in Save Binary

SETFLD - The location of LIF instruction inserted into the header.

SPØ2Ø -
SPØ3Ø -
SP1ØØ -
SP11Ø -
SP1 Ø -

Tags in Save Program

SRORBN - The first core location of the subroutine that determines whether a name match is source or binary.

SWA - The pseudo-unit that holds the source working area.

SWITCH - A core location that is used by the gap search subroutines in conjunction with the upper and lower file area.

UNITNO - The core location that holds the unit number for assembly by name.

USENO - The core location containing the count of binary blocks used.

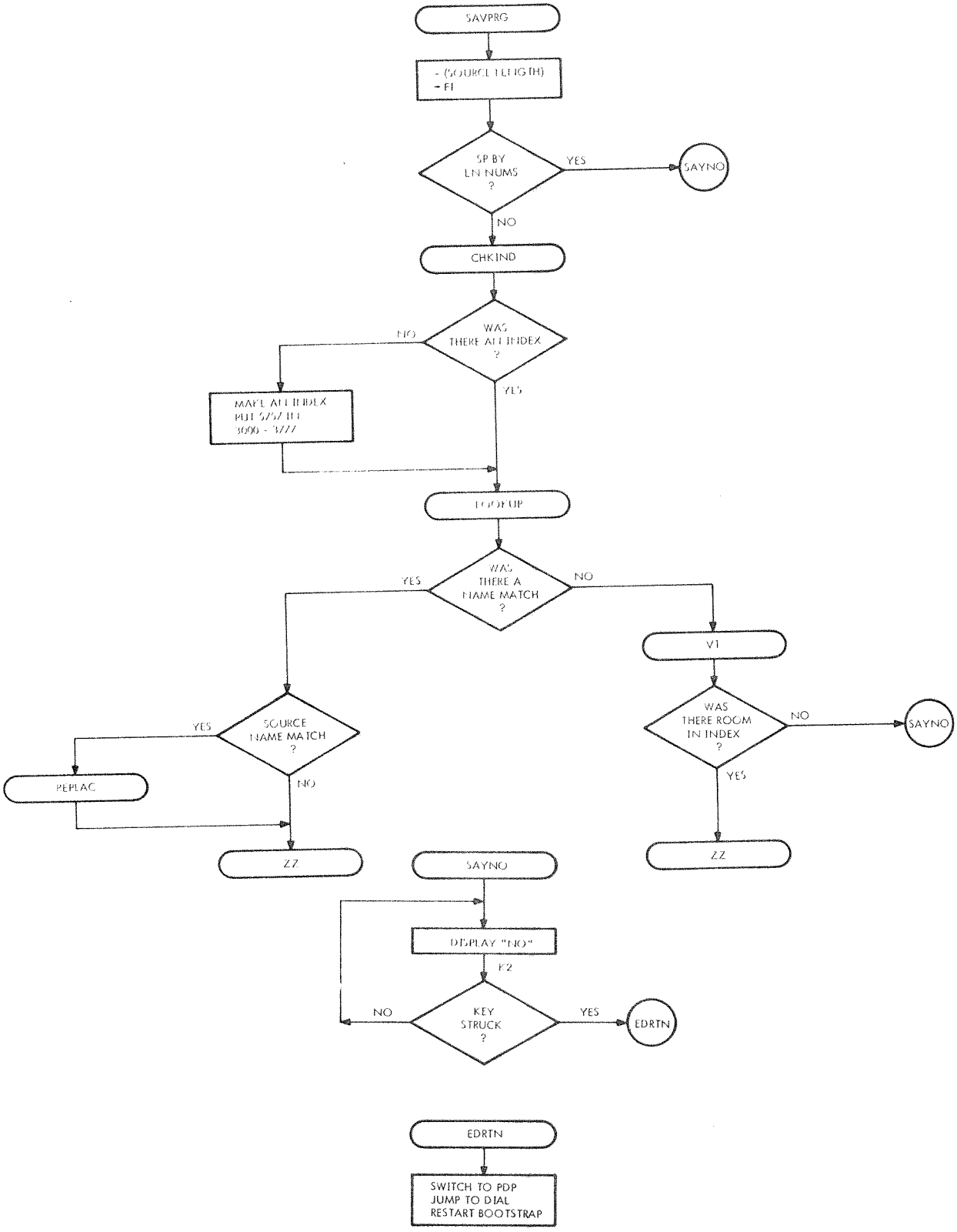
WA - The first TBLK of the Working Area.

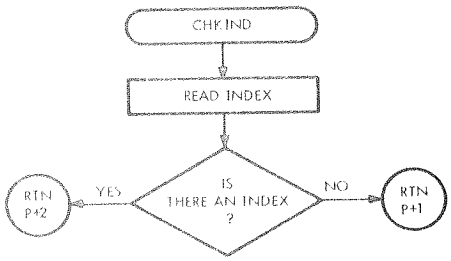
WRITE - The page Ø location pointing to Write routine.

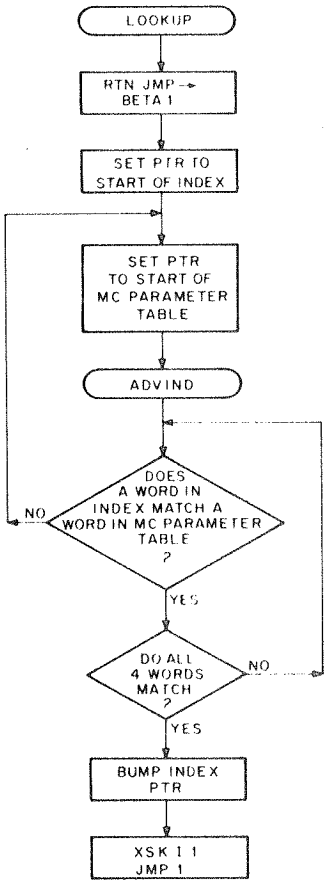
XBLK - The TBLK containing the index.

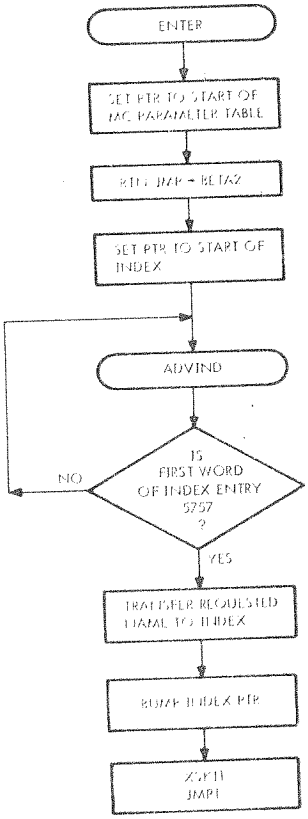
XIO - The control block for reading and writing index.

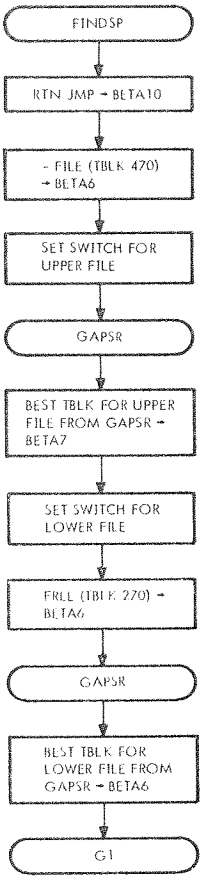
ZERO - Constant zero.

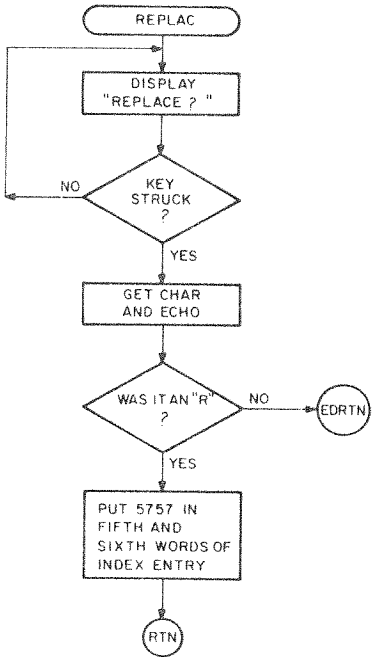

SAVPRG
- (SOURCE LENGTH)
→ F1
SP BY
LN NUMS
?
YES
SAYNO
NO
CHKIND
WAS
THERE AN INDEX
?
NO
YES
MAKE AN INDEX
PUT SPACE
3000 - 3777
LOOK UP
WAS
THERE A
NAME MATCH
?
YES
NO
V1
SOURCE
NAME MATCH
?
YES
NO
WAS
THERE ROOM
IN INDEX
?
NO
YES
SAYNO
REPLAC
ZZ
ZZ
SAYNO
DISPLAY "NO"
F2
KEY
STRUCK
?
NO
YES
EDRTN
EDRTN
SWITCH TO PDP
JUMP TO DIAL
RESTART BOOTSTRAP

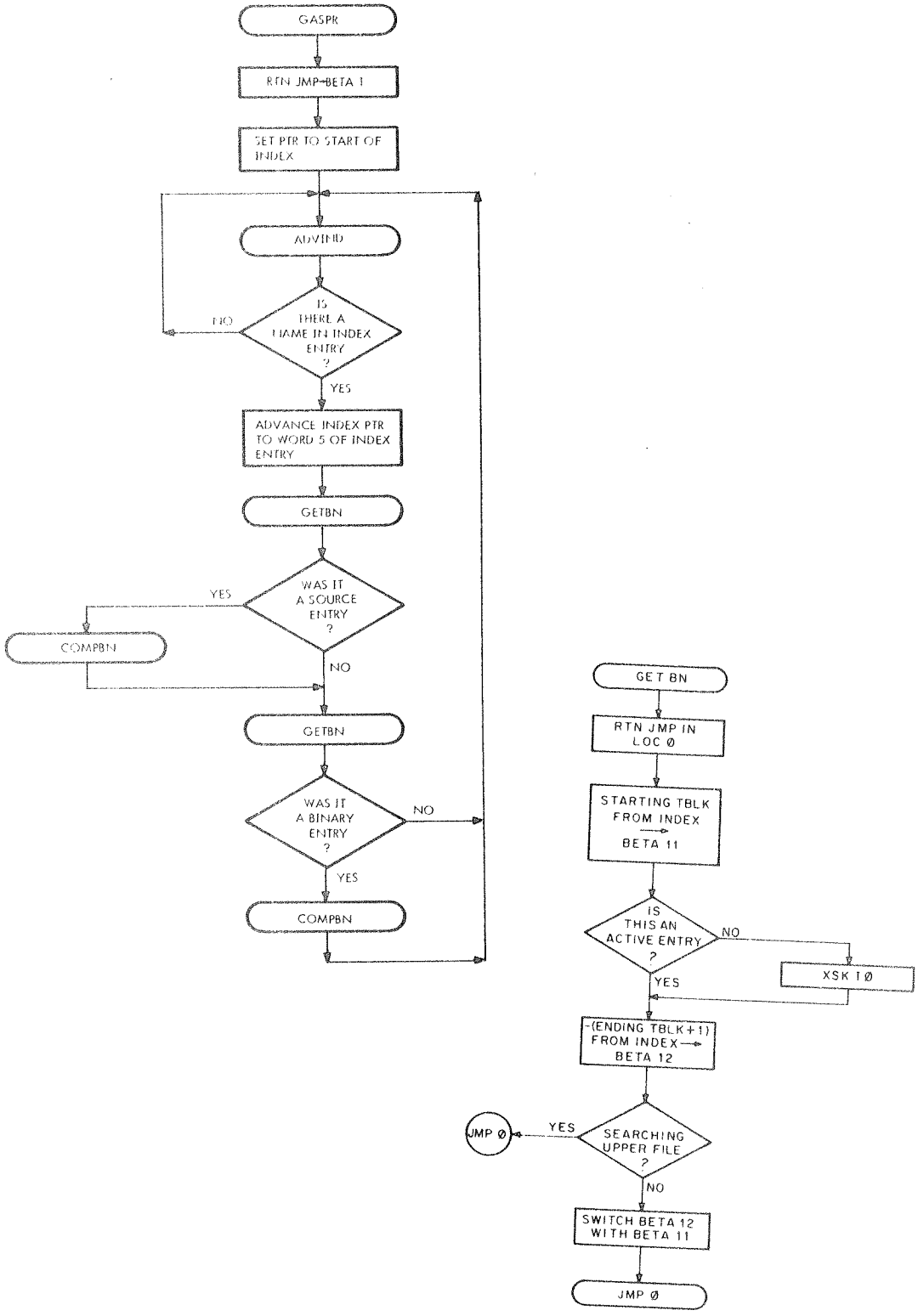

CHKIND
READ INDEX
IS THERE AN INDEX ?
YES
NO
RTN P+2
RTN P+1

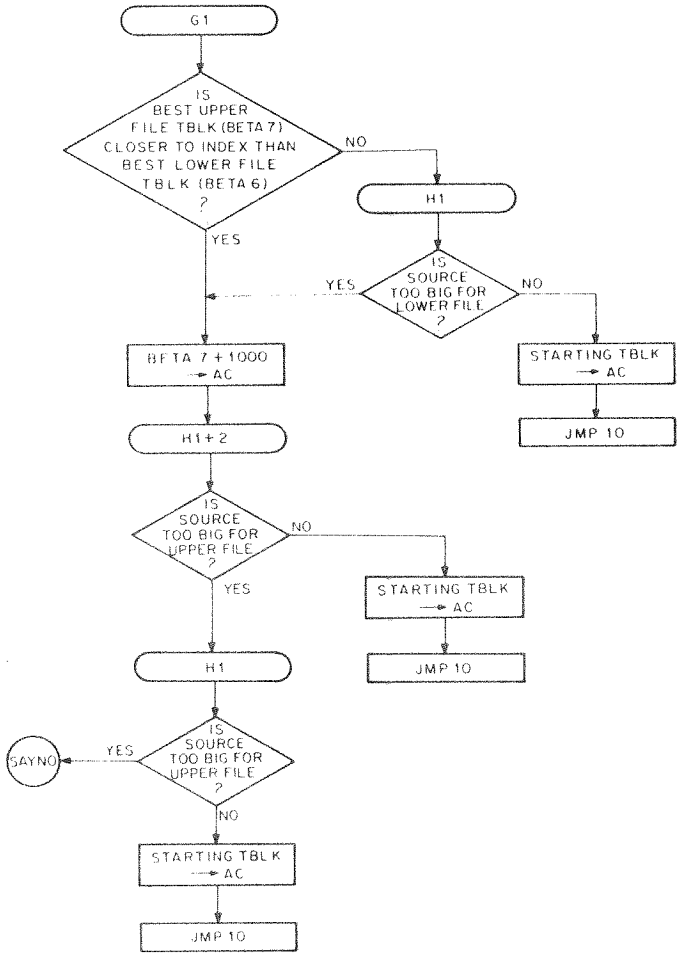

LOOKUP
RTN JMP → BETA 1
SET PTR TO START OF INDEX
SET PTR TO START OF MC PARAMETER TABLE
ADVIND
DOES A WORD IN INDEX MATCH A WORD IN MC PARAMETER TABLE ?
NO
YES
DO ALL 4 WORDS MATCH ?
NO
YES
BUMP INDEX PTR
XSK 1 1
JMP 1

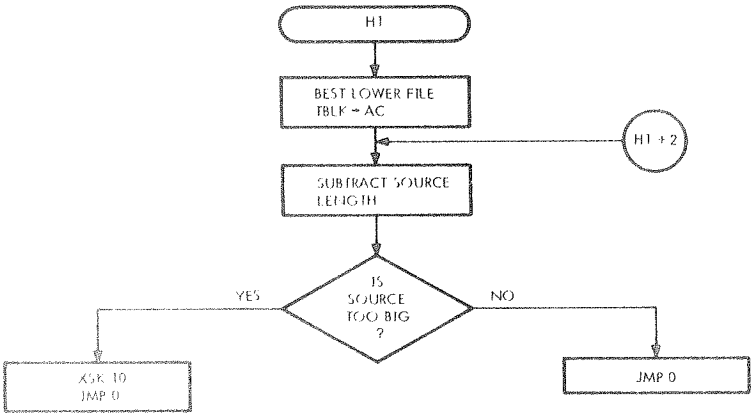

ENTER
SET PTR TO START OF IAC PARAMETER TABLE
RTN JMP → BETA2
SET PTR TO START OF INDEX
ADVIND
IS FIRST WORD OF INDEX ENTRY 5757 ?
NO
YES
TRANSFER REQUESTED LABEL TO INDEX
BUMP INDEX PTR
XCT 11 JMP!

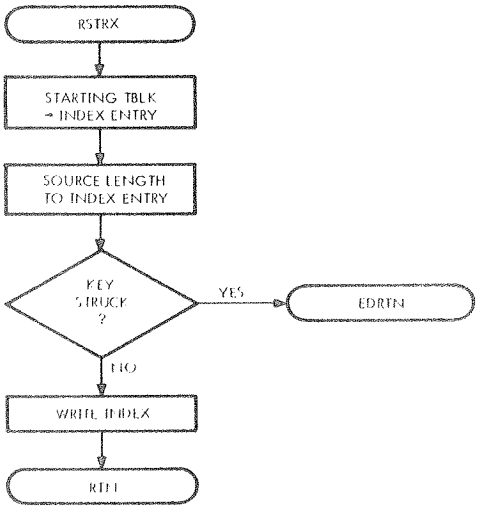

FINDSP
RTN JMP → BETA10
- FILE (TBLK 470) → BETA6
SET SWITCH FOR UPPER FILE
GAPSR
BEST TBLK FOR UPPER FILE FROM GAPSR → BETA7
SET SWITCH FOR LOWER FILE
FRLL (TBLK 270) → BETA6
GAPSR
BEST TBLK FOR LOWER FILE FROM GAPSR → BETA6
G1

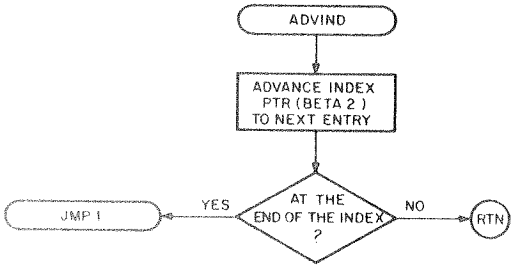

REPLAC
DISPLAY "REPLACE ? "
KEY STRUCK ?
NO
YES
GET CHAR AND ECHO
WAS IT AN "R" ?
NO
EDRTN
YES
PUT 5757 IN FIFTH AND SIXTH WORDS OF INDEX ENTRY
RTN

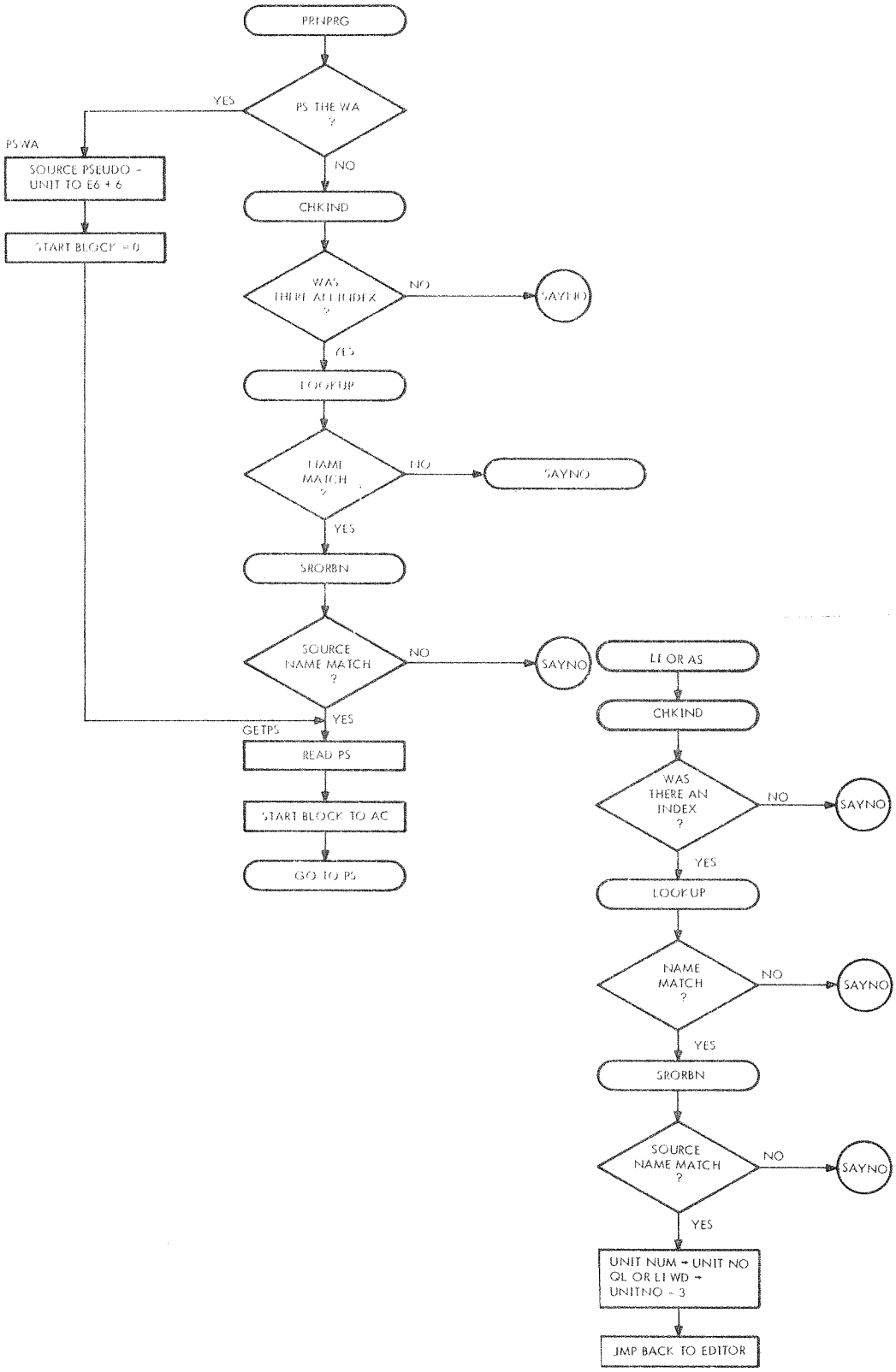

GASPR
RTN JMP→BETA 1
SET PTR TO START OF INDEX
ADVIND
IS THERE A NAME IN INDEX ENTRY ?
NO
YES
ADVANCE INDEX PTR TO WORD 5 OF INDEX ENTRY
GETBN
WAS IT A SOURCE ENTRY ?
YES
COMPBN
NO
GETBN
WAS IT A BINARY ENTRY ?
NO
YES
COMPBN
GET BN
RTN JMP IN LOC Ø
STARTING TBLK FROM INDEX → BETA 11
IS THIS AN ACTIVE ENTRY ?
NO
XSK 1 Ø
YES
-(ENDING TBLK+1) FROM INDEX → BETA 12
JMP Ø
YES
SEARCHING UPPER FILE ?
NO
SWITCH BETA 12 WITH BETA 11
JMP Ø

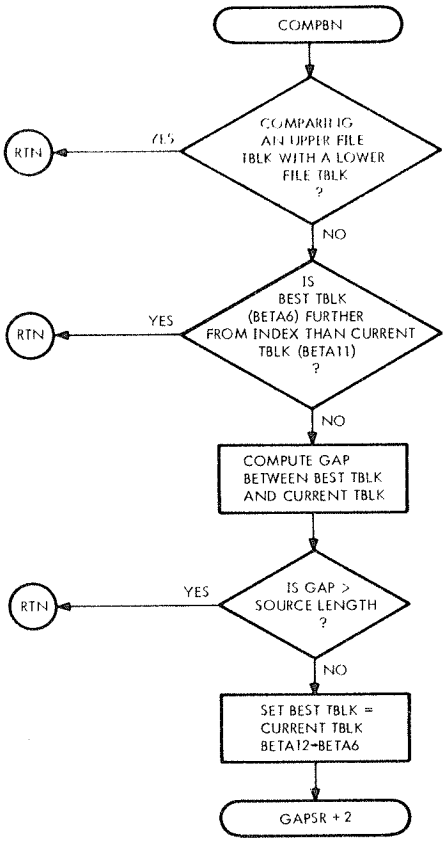

G1
IS BEST UPPER FILE TBLK (BETA 7) CLOSER TO INDEX THAN BEST LOWER FILE TBLK (BETA 6) ?
NO
YES
H1
IS SOURCE TOO BIG FOR LOWER FILE ?
YES
NO
BETA 7 + 1000 → AC
STARTING TBLK → AC
JMP 10
H1+2
IS SOURCE TOO BIG FOR UPPER FILE ?
NO
YES
STARTING TBLK → AC
JMP 10
H1
IS SOURCE TOO BIG FOR UPPER FILE ?
YES
SAYNO
NO
STARTING TBLK → AC
JMP 10

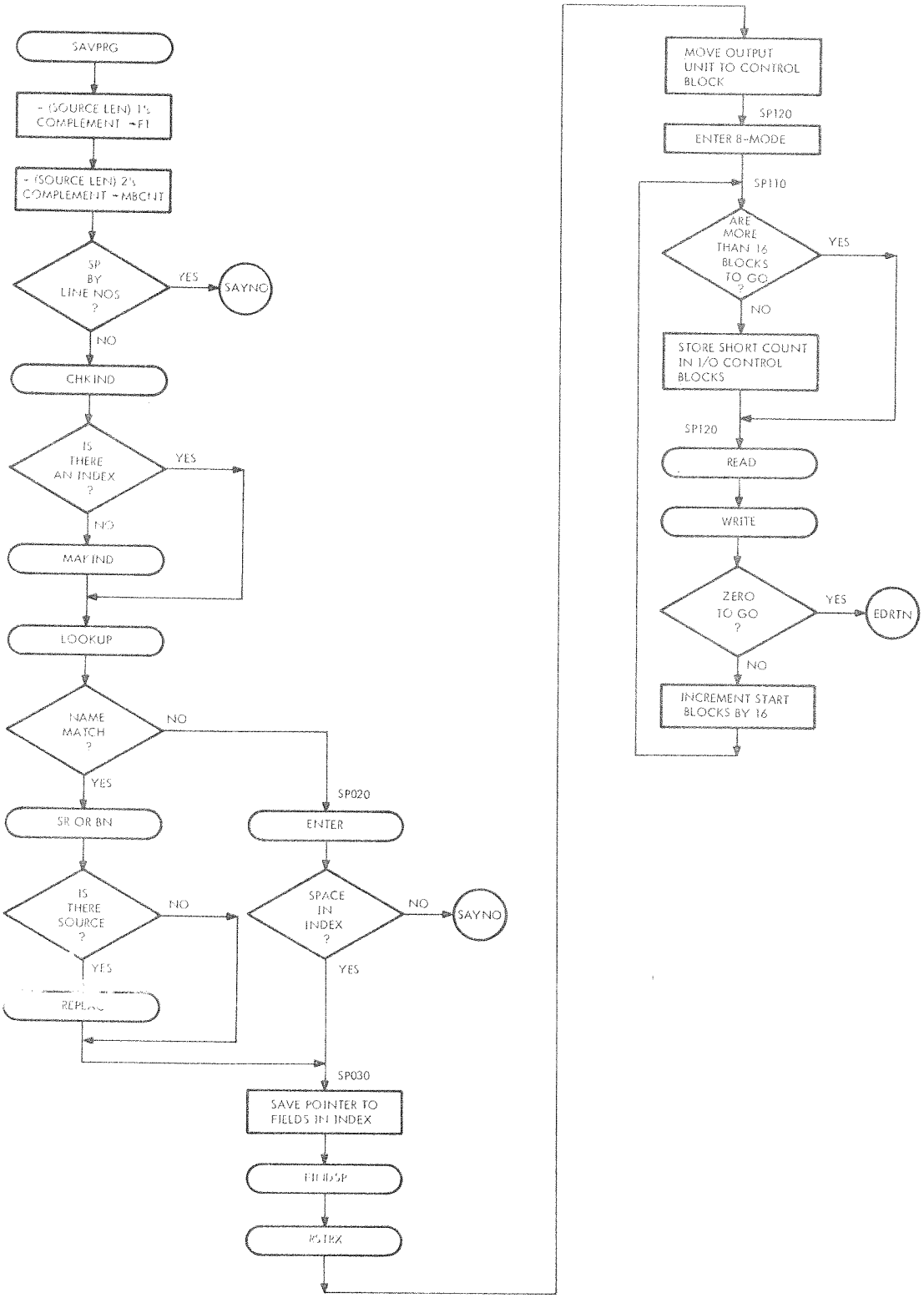

H1
BEST LOWER FILE
TBLK → AC
H1 + 2
SUBTRACT SOURCE
LENGTH
IS
SOURCE
TOO BIG
?
YES
NO
XSK 10
JMP 0
JMP 0

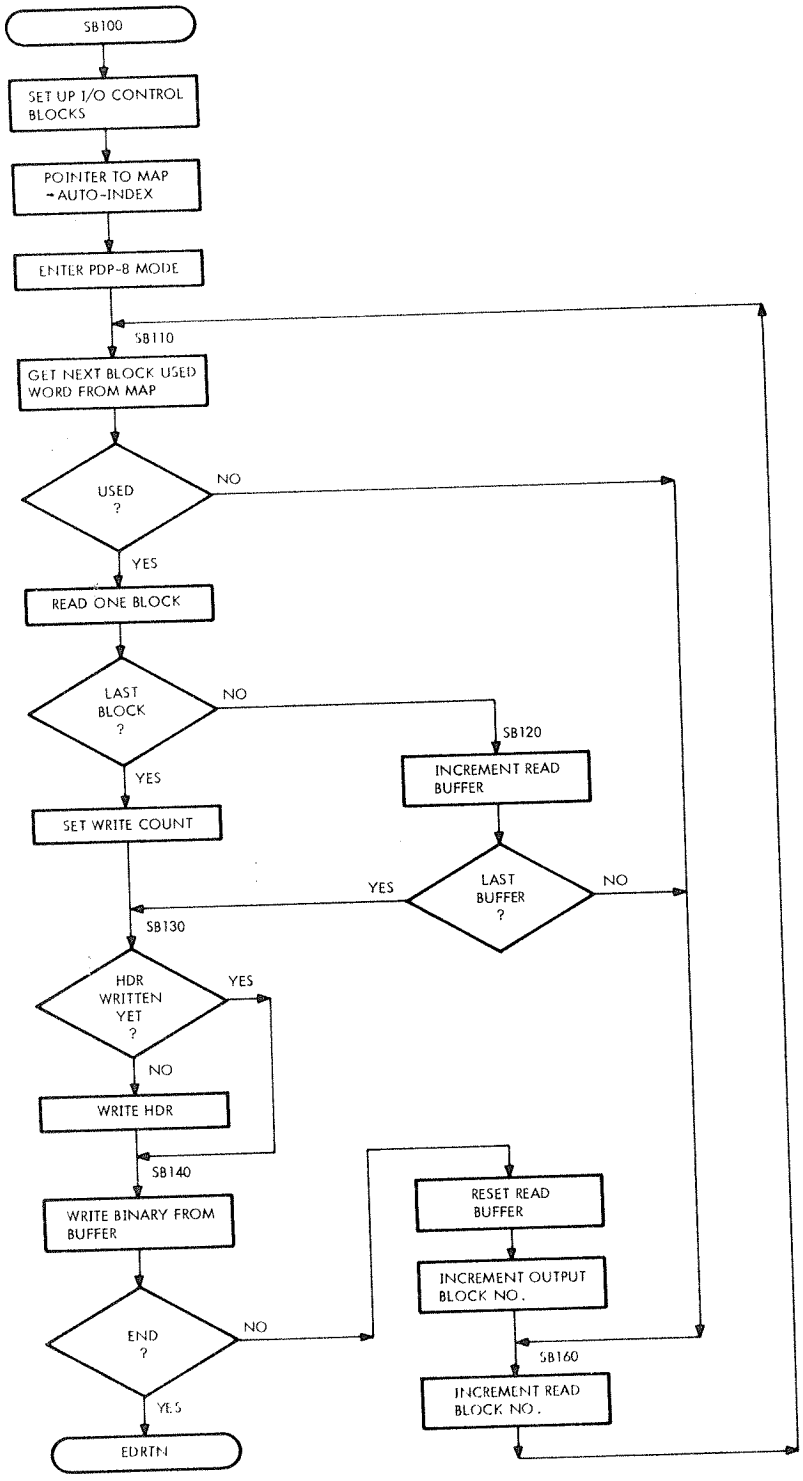

RSTRX
STARTING TBLK
→ INDEX ENTRY
SOURCE LENGTH
TO INDEX ENTRY
KEY
STRUCK
?
YES
EDRTN
NO
WRITE INDEX
RTN1

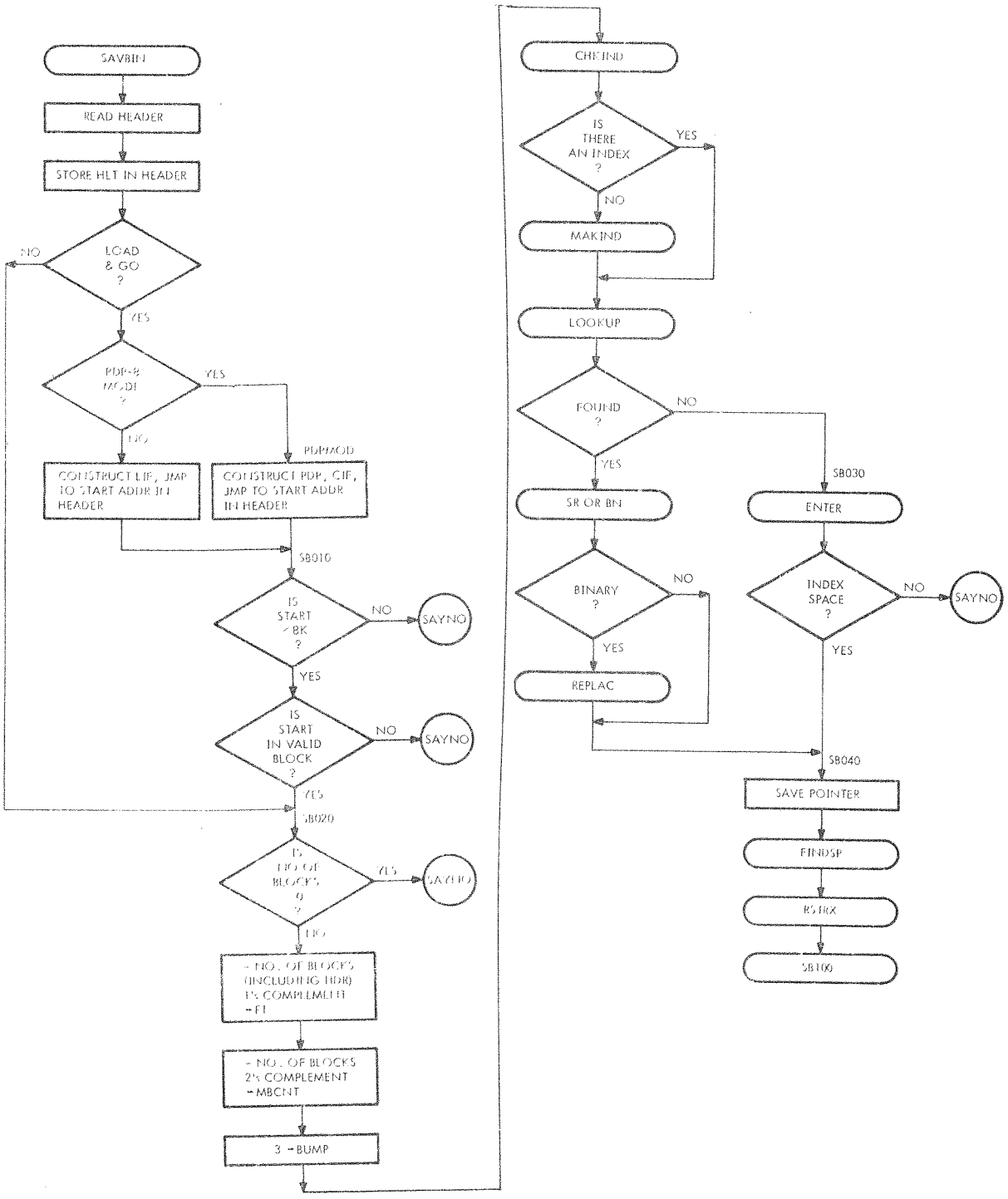

ADVIND
ADVANCE INDEX
PTR (BETA 2)
TO NEXT ENTRY
AT THE
END OF THE INDEX
?
YES
NO
JMP 1
RTN


PRNPRG
PS THE WA ?
YES
PSWA
SOURCE PSEUDO = UNIT TO E6 + 6
START BLOCK = 0
NO
CHKIND
WAS THERE AN INDEX ?
NO
SAYNO
YES
LOOKUP
NAME MATCH ?
NO
SAYNO
YES
SRORBN
SOURCE NAME MATCH ?
NO
SAYNO
YES
GETPS
READ PS
START BLOCK TO AC
GO TO PS
LI OR AS
CHKIND
WAS THERE AN INDEX ?
NO
SAYNO
YES
LOOKUP
NAME MATCH ?
NO
SAYNO
YES
SRORBN
SOURCE NAME MATCH ?
NO
SAYNO
YES
UNIT NUM → UNIT NO
QL OR LI WD → UNITNO - 3
JMP BACK TO EDITOR


COMPBN
COMPARING AN UPPER FILE TBLK WITH A LOWER FILE TBLK ?
YES
RTN
NO
IS BEST TBLK (BETA6) FURTHER FROM INDEX THAN CURRENT TBLK (BETA11) ?
YES
RTN
NO
COMPUTE GAP BETWEEN BEST TBLK AND CURRENT TBLK
IS GAP > SOURCE LENGTH ?
YES
RTN
NO
SET BEST TBLK = CURRENT TBLK BETA12→BETA6
GAPSR + 2


SAVPRG
- (SOURCE LEN) 1's COMPLEMENT →F1
- (SOURCE LEN) 2's COMPLEMENT →MBCNT
SP BY LINE NOS ?
YES
SAYNO
NO
CHKIND
IS THERE AN INDEX ?
YES
NO
MAKIND
LOOKUP
NAME MATCH ?
NO
YES
SP020
SR OR BN
ENTER
IS THERE SOURCE ?
NO
SPACE IN INDEX ?
NO
SAYNO
YES
YES
REPLAC
SP030
SAVE POINTER TO FIELDS IN INDEX
FIELDSP
RSTRX
MOVE OUTPUT UNIT TO CONTROL BLOCK
SP120
ENTER B-MODE
SP110
ARE MORE THAN 16 BLOCKS TO GO ?
YES
NO
STORE SHORT COUNT IN I/O CONTROL BLOCKS
SP120
READ
WRITE
ZERO TO GO ?
YES
EDRTN
NO
INCREMENT START BLOCKS BY 16


SB100
SET UP I/O CONTROL BLOCKS
POINTER TO MAP → AUTO-INDEX
ENTER PDP-8 MODE
SB110
GET NEXT BLOCK USED WORD FROM MAP
USED ?
NO
YES
READ ONE BLOCK
LAST BLOCK ?
NO
YES
SB120
INCREMENT READ BUFFER
LAST BUFFER ?
YES
NO
SET WRITE COUNT
SB130
HDR WRITTEN YET ?
YES
NO
WRITE HDR
SB140
WRITE BINARY FROM BUFFER
END ?
NO
YES
EDRTN
RESET READ BUFFER
INCREMENT OUTPUT BLOCK NO.
SB160
INCREMENT READ BLOCK NO.


SAVBIN
READ HEADER
STORE HLT IN HEADER
LOAD & GO ?
NO
YES
PDP-8 MODE ?
YES
NO
CONSTRUCT LIF, JMP TO START ADDR IN HEADER
PDP8OD
CONSTRUCT PDP, CIF, JMP TO START ADDR IN HEADER
SB010
IS START ≠ BK ?
NO
SAYNO
YES
IS START IN VALID BLOCK ?
NO
SAYNO
YES
SB020
IS NO. OF BLOCKS = 0 ?
YES
SAZERO
NO
- NO. OF BLOCKS (INCLUDING HDR) 1's COMPLEMENT → R1
- NO. OF BLOCKS 2's COMPLEMENT → MBCNT
3 → BUMP
CHKIND
IS THERE AN INDEX ?
YES
NO
MAKIND
LOOKUP
FOUND ?
NO
YES
SR OR BN
BINARY ?
NO
YES
REPLAC
SB030
ENTER
INDEX SPACE ?
NO
SAYNO
YES
SB040
SAVE POINTER
FINDSP
R5TRX
SB100

```
0000                    *20
0001                    / FCMSV4,5    FILE COMMANDS, V4,5
0002                    /
0003                    /       LAP6-DIAL FILE COMMANDS ... SB,SP,AB,LI,QL,AS,PS
0004                    /
0005                    /       MAR 31, 1970
0006                    /
0007                    /       THIS ROUTINE IS A MODIFICATION OF FUNCTIONALLY EQUIVALENT
0010                    /       CODE FROM LAP6.  COMMENTS HAVE BEEN ADDED, TAGS CHANGED FROM
0011                    /       LETTER-DIGIT TO MNEMONIC NAMES, ABSOLUTE MEMORY ADDRESS REFERENCES
0012                    /       REMOVED WHERE POSSIBLE, I/O DISCIPLINE AND BUFFERING IMPOSED,
0013                    /       AND THE WHOLE CONDENSED FROM 7 BLOCKS TO 4,  WE ARE DEEPLY INDEBTED.
0014                    /
0015                    /       RESIDES IN DIAL BLOCKS 50-53
0016                    /       OCCUPIES MEMORY LOCATIONS 4000-5777
0017                    /
0020                    /       16 MBLKS (OCTAL) IN UPPER FIELD ARE USED FOR BUFFERS: 1 0000 - 1 6777
0021                    /       LOCATIONS 1 7000 TO 1 7777 MUST REMAIN INTACT,
0022                    /           THESE LOCATIONS CONTAIN DIAL I/O ROUTINES AND BOOTSTRAP,
0023                    /
0024                    /       THE FOLLOWING AREAS IN THE LOWER PDP-8 FIELD ARE USED:
0025                    /
0026                    /       AUTO-INDEX REGISTER 10 IS USED BY SB AND AB,
0027                    /       PAGE ZERO LOCATIONS 21 AND 22 ARE ASSUMED TO CONTAIN POINTERS
0030                    /           TO THE DIAL I/O ROUTINES IN THE UPPER FIELD,
0031                    /       2000 - 2200 ARE ASSUMED TO CONTAIN THE DIAL CHARACTER DISPLAY TABLE,
0032                    /       2371 - 2377 CONTAIN THE MC PARAMETER TABLE, E6,
0033                    /       3000 - 3777 ARE USED TO READ AND WRITE THE DIAL INDEX,
0034                    /           AND FOR HEADERS DURING AB.
0035                    /       5400 - 5777 ARE USED FOR THE BINARY HEADER BLOCK DURING SB OPERATIONS.
0036                    /       5000 - 5777 ARE USED BY THE MAIN CODE OF PS (PRINT SOURCE).
0037                    /
0040                    /       CODE IS NOT REUSABLE
0041                    /
0042                    /
0043    0020  6026      FCSA,   JMP SAVBIN      /EDITOR
0044    0021  7120              JMP ADDBIN      /CALLS
0045    0022  6022              JMP .           /ALL
0046    0023  6341              JMP SAVPRG      /FCOMS
0047    0024  7023              JMP LIORAS      /HERE
0050    0025  6376              JMP PRNPRG
0051                            EJECT
```

```
0052                    /
0053                    /
0054                    /       SAVE BINARY ENTERS HERE
0055                    /
0056                    /
0057    0026    0002    SAVBIN, PDP             /PMODE TO READ HEADER
0060                            PMODE
0061    4027    6201            CDF L08FLD
0062    4030    6212            CIF HI8FLD
0063    4031    4421            JMS I READ
0064    4032    4323            HDRIO           /PARAMS FOR READING HEADER BLOCK
0065    4033    6141            LINC            /RETURN TO LINC MODE
0066                            LMODE
0067    0034    0641            LDF DATSEG      /FORCE DATA FIELD FOR REF TO E6
0070    0035    0011            CLR             /LINC-MODE HALT TO HEADER
0071    0036    5400            STC HEADER      /IN CASE NOT LOAD AND GO
0072    0037    1000            LDA
0073    0040    2371            E6              /EQ 0 IF NOT L&G
0074                    /  SIGN EQ 0 IF LINC; EQ 1 IF 8MODE; 8 FLD IS IN 9-11
0075    0041    0470            AZE I
0076    0042    6134            JMP SB020       /NOT L&G
0077    0043    0451            APO
0100    0044    6070            JMP PDPMOD      /8 MODE
0101                    /
0102                    /       START IN LINC MODE
0103                    /
0104    0045    0242            ROL 2           /LOW-ORDER BIT OF...
0105    0046    1560            BCL I           /...PDP-8 FIELD
0106    0047    7773            7773            /MAX OF 7 (8K)
0107    0050    1340            STH             /MODIFY LIF INSTRUCTION
0110    0051    4060            SETFLD!4000
0111    0052    1000            LDA
0112    0053    2372            E6+1            /2 LEAST SIG BITS OF IF
0113                                            /12 BIT 8 MODE ADDR
0114    0054    0242            ROL 2
0115    0055    1560            BCL I
0116    0056    7774            7774
0117    0057    1620            BSE I           /MAKE AN LIF
0120    0060    0600    SETFLD, LIF
0121    0061    5400            STC HEADER
0122    0062    1000            LDA
0123    0063    2372            E6+1            /STARTING ADDR
0124    0064    1620            BSE I           /MAKE A LINC-MODE JMP
0125    0065    6000            JMP
0126    0066    5401            STC HEADER+1
0127    0067    6111            JMP SB010       /ALL SET
0130                            EJECT
-
```

```
0131                        /
0132                        /       COME HERE IF PROGRAM IS TO START IN 8-MODE
0133                        /
0134    0070  1020  PDPMOD,  LDA I           /SET UP FOR 8MOD START
0135    0071  0002          PDP
0136    0072  5400          STC HEADER
0137    0073  1020          LDA I
0140    0074  5777          5777            /JMP I 377
0141    0075  5402          STC HEADER+2
0142    0076  1000          LDA             /CHK FIELD
0143    0077  2371          E6
0144    0100  0243          RUL 3
0145    0101  1560          BCL I           /8K FOR THIS VERSION
0146    0102  7767          7767
0147    0103  1620          BSE I           /MAKE A CIF
0150    0104  6202          6202
0151    0105  5401          STC HEADER+1
0152    0106  1000          LDA
0153    0107  2372          E6+1            /STARTING ADDR
0154    0110  5403          STC HEADER+3
0155                    /
0156                    /       CHECK STARTING ADDR FOR VALIDITY
0157                    /           (LT 8K AND IN A VALID BLOCK)
0160                    /
0161    0111  1000  SB010,  LDA             /HIGH ORDER ADDRESS DIGIT
0162    0112  2371          E6
0163    0113  0321          ROR I 1         /MOVE LOWER/UPPER SEGMENT BIT TO LINK
0164    0114  1560          BCL I           /CLEAR NON-ADDRESS BITS
0165    0115  7774          7774
0166    0116  0450          AZE             /SKIP IF START ADD LT 8KK
0167    0117  7071          JMP SAYNO       /OTHERWISE REFUSE
0170    0120  1000          LDA             /GET LOW ORDER ADDRESS
0171    0121  2372          E6+1
0172    0122  1560          BCL I           /TRUNCATE TO START OF MBLK
0173    0123  0377          0377
0174    0124  0265          ROL I 5         /MOVE RELATIVE BLOCK NO (0-37) TO LOW ORDER
0175    0125  1620          BSE I           /INDEX THE ADD INSTRUCTION
0176    0126  3740          ADD MAP
0177    0127  4130          STC ,+1         /STORE AND EXECUTE IT
0200    0130  0000          0               /FILLED BY (ADD MAP+BLK) INSTRUCTION
0201    0131  1460          SAE I           /TEST FOR VALID BLOCK
0202    0132  7777          7777
0203    0133  7071          JMP SAYNO       /INVALID -- REFUSE
0204                        EJECT
```

```
0205                    /
0206                    /       HEADER IS NOW PROPERLY SET UP
0207                    /           LOOK FOR SPACE FOR THE FILE
0210                    /
0211        0134  3737  SB020,  ADD USENO        /NO OF BLKS USED (FROM HEADER)
0212        0135  0470          AZE I           /BOMB IF ZERO
0213        0136  7071          JMP SAYNO
0214        0137  2347          ADD P1          /PLUS ONE FOR HEADER
0215        0140  0017          COM             /FINDSP WANTS IT NEGATIVE
0216        0141  1040          STA
0217        0142  0345          F1              /HOLD FOR SPACE SCAN
0220        0143  2161          ADD P2          /TWOS COMP OF NO OF BLOCKS (WITHOUT HEADER)
0221        0144  4327          STC MBCNT       /HOLD FOR COPY
0222        0145  3127          ADD P3
0223        0146  4536          STC BUMP        /REPLACE BUMP CONSTANT IN LOOKUP
0224        0147  6464          JMP CHKIND      /READ IN THE INDEX BLK
0225        0150  6505          JMP MAKIND      /RTN FOR NO INDEX-MAKE ONE
0226        0151  6517          JMP LOOKUP      /SEARCH INDEX FOR NAME IN E6+2
0227        0152  6156          JMP SB030       /NOT FOUND
0230        0153  6436          JMP SRORBN      /CHK IF SYMBOLIC ONLY
0231        0154  6543          JMP REPLAC      /NO-DISPLAY REPLACE FOR VERIFY
0232        0155  6160          JMP SB040       /BIN ENTRY EMPTY, OR REPLACE REQUESTED
0233        0156  6622  SB030,  JMP ENTER       /PUT NAME IN INDEX
0234        0157  7071          JMP SAYNO       /NO SPACE IN THE INDEX
0235        0160  0044  SB040,  SET 4           /SAVE XR PTR
0236        0161  0002  P2,     2
0237        0162  6647          JMP FINDSP      /FIND BEST SPACE, RETURN TBLK IN AC
0240        0163  6775          JMP RSTRX       /UPDATE & WRT XR
0241        0164  1020  SB100,  LDA I           /BINARY WORK UNIT,,,
0242        0165  0111          BWA
0243        0166  4313          STC INFIL       /,,,TO CONTROL INPUT
0244        0167  2321          ADD OUTFIL+2&1777 /STARTING BLOCK
0245        0170  1040          STA             /,,,IS HEADER TBLK,,,
0246        0171  4325          HDRIO+2
0247        0172  2347          ADD P1          /,,,PLUS 1
0250        0173  4321          STC OUTFIL+2    /,,,IS START OF ACTUAL BINARY
0251        0174  2347          ADD P1          /CONSTANT 1...
0252        0175  4316          STC INFIL+3     /FOR SINGLE-BLOCK INPUT
0253        0176  3017          ADD XIO         /OUTPUT UNIT
0254        0177  1040          STA             /,,,FOR WRITING HEADER...
0255        0200  4323          HDRIO
0256        0201  4317          STC OUTFIL      /,,,AND BINARY
0257        0202  1020          LDA I           /GET ADDRESS OF BLOCK MAP
0260        0203  5737          MAP-1!4000      /MAP IN LINC FIELD 2
0261        0204  0002          PDP             /ENTER PMODE FOR COPY
0262                            PMODE
0263        4205  3010          DCA AUTO        /SETUP AUTO-INDEX REG
0264                            EJECT
```

```
0265                    /
0266                    /       THIS IS THE COPY LOOP FOR SAVBIN BINARY
0267                    /       N.B.  CODE BETWEEN THIS POINT AND EDRIN MUST BE IN ONE PDP-8 PAGE.
0270                    /
0271    4206    1410    SB110,  TAD I AUTO      /GET A WORD FROM MAP
0272    4207    7650            SNA CLA         /SKIP IF BLOCK USED
0273    4210    5254            JMP SB160       /NOT USED -- BUMP BLOCK NO AND LOOP
0274    4211    6212            CIF HI8FLD      /SET INST FIELD FOR READ ROUTINE
0275    4212    4421            JMS I READ      /READ ONE BINARY BLOCK
0276    4213    4313            INFIL           /...FROM BWA
0277    4214    7201            CLA IAC         /CONSTANT ONE
0300    4215    2327            ISZ MBCNT       /TEST FOR LAST BINARY BLOCK
0301    4216    5223            JMP SB120       /DONT STOP YET
0302                    /
0303                    /       WE HAVE JUST READ THE LAST BINARY BLOCK
0304                    /
0305    4217    1333            TAD MBFAD       /SUBTRACT (FIRST BUFF - 1)...
0306    4220    1314            TAD INFIL+1     /...FROM CURRENT BUFFER...
0307    4221    3322            DCA OUTFIL+3    /...FOR OUTPUT BLOCK COUNT
0310    4222    5231            JMP SB130
0311                    /
0312                    /       COME HERE AFTER READING ANY BINARY BLOCK EXCEPT THE LAST
0313                    /
0314    4223    1314    SB120,  TAD INFIL+1     /INCREMENT BUFFER ADDRESS
0315    4224    3314            DCA INFIL+1
0316    4225    1334            TAD MBFEND      /BUFFER LIMIT...
0317    4226    1314            TAD INFIL+1     /...COMPARED TO BUFFER ADDRESS
0320    4227    7710            SPA CLA         /DUMP BUFFERS NOW IF FULL
0321    4230    5254            JMP SB160       /OTHERWISE BUMP INPUT BLOCK
0322                    /
0323                    /       COME HERE TO WRITE THE BUFFERS:
0324                    /           EITHER THEY ARE FULL, OR THERE IS NO MORE TO READ.
0325                    /
0326    4231    1330    SB130,  TAD HDRSW       /SWITCH TELLS WHETHER HEADER HAS BEEN WRITTEN
0327    4232    7650            SNA CLA         /SKIP IF NOT WRITTEN YET
0330    4233    5240            JMP SB140       /BUT IF ITS DONE, DONT RE-WRITE
0331    4234    3330            DCA HDRSW       /SET THE SWITCH
0332    4235    6212            CIF HI8FLD
0333    4236    4422            JMS I WRITE     /WRITE THE HEADER
0334    4237    4323            HDRIO
0335    4240    6212    SB140,  CIF HI8FLD
0336    4241    4422            JMS I WRITE     /WRITE BINARY BLOCKS
0337    4242    4317            OUTFIL
0340    4243    7200            CLA
0341    4244    1327            TAD MBCNT       /GET MINUS REMAINING COUNT
0342    4245    7650            SNA CLA         /SKIP NOT END OF SAVBIN
0343    4246    5336            JMP EDRIN+1     /RETURN TO EDITOR
0344                    /
0345                    /       COME HERE TO RE-INITIALIZE CONTROL BLOCKS
0346                    /           IF MORE BINARY REMAINS TO BE DONE
0347                    /
0350    4247    1332            TAD BFAD        /BUFFER START ADDRESS
0351    4250    3314            DCA INFIL+1
0352    4251    1331            TAD BFLN        /BLOCKS WRITTEN...
0353    4252    1321            TAD OUTFIL+2    /...PLUS START BLOCK...
0354    4253    3321            DCA OUTFIL+2    /...IS NEW START BLOCK
0355    4254    2315    SB160,  ISZ INFIL+2     /GO TO NEXT BLOCK IN BWA
0356    4255    5206            JMP SB110       /AND LOOP TO END
0357                    EJECT
```

```
0360                    /
0361                    /
0362                    /       COPY LOOP FOR SAVE PROGRAM (SP)
0363                    /
0364                    /
0365    4256    0002    SP100,  0002            /ENTER 8-MODE FOR COPY
0366    4257    1331    SP110,  TAD BFLN
0367    4260    1327            TAD MBCNT       /ADD TO MINUS BLOCK COUNT
0370    4261    7510            SPA             /SKIP IF 16 OR LESS TO GO
0371    4262    5271            JMP SP120
0372    4263    7200            CLA
0373    4264    1327            TAD MBCNT       /PICK UP MINUS REMAINING COUNT
0374    4265    7041            CIA             /MAKE POSITIVE
0375    4266    3316            DCA INFIL+3     /STORE SHORT COUNT
0376    4267    1316            TAD INFIL+3     /,,,ALSO,,,
0377    4270    3322            DCA OUTFIL+3    /,,,OUTPUT SHORT
0400    4271    3327    SP120,  DCA MBCNT       /STORE MINUS BLOCKS-TO-GO
0401    4272    6212            CIF HI8FLD      /FIELD CONTAINING I/O HANDLERS
0402    4273    4421            JMS I READ      /READ UP TO 16 BLOCKS
0403    4274    4313            INFIL
0404    4275    6212            CIF HI8FLD
0405    4276    4422            JMS I WRITE     /WRITE THEM OUT
0406    4277    4317            OUTFIL
0407    4300    7200            CLA
0410    4301    1327            TAD MBCNT       /PICK UP MINUS BLOCK COUNT
0411    4302    7650            SNA CLA         /SKIP IF MORE TO GO
0412    4303    5336            JMP EDRTN+1     /OTHERWISE GO TO EDITOR
0413    4304    1331            TAD BFLN
0414    4305    1315            TAD INFIL+2     /INCREMENT STARTING BLOCK
0415    4306    3315            DCA INFIL+2
0416    4307    1331            TAD BFLN
0417    4310    1321            TAD OUTFIL+2    /,,,FOR INPUT AND OUTPUT
0420    4311    3321            DCA OUTFIL+2
0421    4312    5257            JMP SP110       /LOOP
0422                    /
0423                    /     I/O CONTROL BLOCKS
0424                    /
0425    4313    0110    INFIL,  SWA             /UNIT FOR INPUT
0426    4314    0020            BUFFER          /MEM ADDR=1 0000, SHIFTED RIGHT 8
0427    4315    0000            0               /START BLOCK
0430    4316    0016            BUFLEN          /NO OF BLOCKS
0431                    /
0432    4317    0000    OUTFIL, 0               /UNIT FOR OUTPUT
0433    4320    0020            BUFFER          /MEM ADDR=1 0000
0434    4321    0000            0               /STARTING BLOCK
0435    4322    0016            BUFLEN          /COUNT
0436                    /
0437    4323    0111    HDRIO,  BWA             /HEADER IS ON BIN WORK AREA UNIT
0440    4324    0013            13              /MEM ADDR = 5400 (3400 IF AB)
0441    4325    0057            HDRBLK          /BLOCK 47 WITHIN BWA
0442    4326    0001            1               /JUST ONE
0443                    /
0444    4327    0000    MBCNT,  0
0445    4330    7777    HDRSW,  7777
0446    4331    0016    BFLN,   BUFLEN
0447    4332    0020    BFAD,   BUFFER
0450    4333    7760    MBFAD,  -BUFFER
0451    4334    7742    MBFEND, -BUFFER-BUFLEN
0452                    EJECT
```

```
0453                    /
0454                    /     RETURN TO EDITOR
0455                    /
0456    4335  0002  EDRTN,  0002              /RETURN IN 8-MODE
0457    4336  6212          CIF MIBFLD        /DIAL IS IN UPPER MEMORY FIELD
0460    4337  5740          JMP I .+1
0461    4340  7777          DRSTRT            /START OF DIAL RESTART
0462                        LMODE
0463                  /
0464                  /
0465                  /     SAVE PROGRAM (SP) ENTER HERE
0466                  /       LAST BLOCK NO OF WORKING SOURCE IS IN AC
0467                  /       FILE NAME AND UNIT IN MC PARAMETER LIST AT E6
0470                  /
0471                  /
0472    0341  1120  SAVPRG, ADA I
0473    0342  2411          -WA-5000-2
0474    0343  0017          COM
0475    0344  1060          STA I             /- NUM OF TBLKS
0476    0345  0000  F1,     0
0477    0346  1120          ADA I             /GET TWOS COMP
0500    0347  0001  P1,     1
0501    0350  4327          STC MBCNT         /HOLD FOR COPY
0502    0351  0641          LDF DATSEG        /FORCE DATA FIELD FOR REF TO E6
0503    0352  1000          LDA               /CHECK FOR PARTIAL SAVE
0504    0353  2371          E6
0505    0354  0450          AZE
0506    0355  7071          JMP SAYNO         /PARTIAL SAVE ILLEGAL
0507    0356  6464          JMP CHKIND        /CHK FOR INDEX
0510    0357  0505          JMP MAKIND        /NO INDEX -- MAKE ONE
0511    0360  6517          JMP LOOKUP        /DO NAME SEARCH
0512    0361  6365          JMP SP020         /NO NAME MATCH
0513    0362  6436          JMP SRORBN        /CHECK FOR SOURCE
0514    0363  6543          JMP REPLAC        /ASK REPLACE
0515    0364  6367          JMP SP030         /REPLACE WAS REQUESTED
0516    0365  6622  SP020,  JMP ENTER         /PUT NAME IN INDEX
0517    0366  7071          JMP SAYNO         /NO ROOM IN INDEX
0520    0367  0044  SP030,  SET 4
0521    0370  0002          2                 /SAVE PTR TO START AND LEN FIELDS
0522    0371  6647          JMP FINDSP        /DO GAP SEARCH
0523    0372  6775          JMP RSTRX         /RE-WRITE INDEX
0524    0373  3017          ADD XIO           /OUTPUT UNIT
0525    0374  4317          STC OUTFIL        /.,,TO OUTPUT PARAMETERS
0526    0375  6256          JMP SP100         /GO DO THE COPY
0527                        EJECT
```

```
0530                    /
0531                    /
0532                    /     PRINT SOURCE
0533                    /
0534                    /
0535    0376  0641      PRNPRG, LDF DATSEG
0536    0377  1300              LDH             /CHK MC PARAM
0537    0400  2373              E6+2            /TABLE
0540    0401  1420              SHD I           /FILE ENTRY
0541    0402  7700              7700            /OR WA ?
0542    0403  6413              JMP PSWA        /PRINT THE WA
0543    0404  6464              JMP CHKIND      /NAMED FILE -- READ AND CHECK THE INDEX
0544    0405  7071              JMP SAYNO       /NO INDEX
0545    0406  6517              JMP LOOKUP      /INDEX OK -- FIND THE NAME
0546    0407  7071              JMP SAYNO       /NO NAME MATCH
0547    0410  6436              JMP SRORBN      /IS THERE SOURCE?
0550    0411  6421              JMP GETPS       /YES -- LOAD PS
0551    0412  7071              JMP SAYNO       /BINARY ONLY, CANT DO IT
0552                    /
0553                    /       COME HERE IF SWA TO BE PRINTED
0554                    /
0555    0413  1020      PSWA,   LDA I           /SOURCE WA PSEUDO-UNIT
0556    0414  0110              SWA
0557    0415  1040              STA             /,,,TO MC PARAM LIST
0560    0416  2377              E6+6            /,,,AS I/O UNIT
0561    0417  0062              SET I 2         /STARTING BLOCK IS ZERO
0562    0420  0435              ZERO
0563                    /
0564                    /       READ PS MAIN CODE -- BLOCKS 63, 4 ON DIAL
0565                    /
0566    0421  0002      GETPS,  PDP             /PMODE FOR READ
0567                            PMODE
0570    4422  6212              CIF HI8FLD
0571    4423  4421              JMS I READ      /READ PS AND TELETYPE INTO 5000
0572    4424  4431              PSIN
0573    4425  6141              LINC
0574                            LMODE
0575    0426  0641              LDF DATSEG      /SET DATA FIELD FOR PS
0576    0427  1002              LDA 2           /PICK UP STARTING BLOCK NO OF DESIRED SOURCE
0577    0430  7000              JMP PSENT       /GO TO PS MAIN PROCESSING
0600                    /
0601                    /       PARAMS FOR READING PS AND TTY
0602                    /
0603    0431  0100      PSIN,   DIALU           /DIAL RESIDENCE UNIT
0604    0432  0012              12              /MEM ADDR = 5000
0605    0433  0063              PSBLK           /DIAL BLOCKS 63,64 CONTAIN PS, TTY
0606    0434  0002              2               /GET EM BOTH
0607                    /
0610    0435  0000      ZERO,   0               /CONSTANT ZERO TO USE AS STARTING BLOCK IF SWA PRINTED
0611                            EJECT
```

```
0612
0613                    /
0614                    /      FILE SUBROUTINES
0615                    /
0616                    /
0617
0620                                        /CHECK SOURCE
0621                                        /OR BINARY
0622    0436  0045  SRORBN, SET 5           /POINTER TO START BLOCK
0623    0437  0002         2
0624    0440  1025         LDA I 5          /DOES NAME HAVE
0625    0441  0451         APO              /A SOURCE? (BINARY IF AB OR SB COMMAND)
0626    0442  0220         XSK I 0          /NO, ENTRY IS EMPTY
0627    0443  6000         JMP 0
0630                    /
0631                    /
0632                    /
0633
0634                                        /FIX ADDRESS
0635    0444  1000  ADVIND, LDA             /TO NEXT NAME
0636    0445  0002         2                /IN INDEX
0637    0446  1620         BSE I            /EACH NAME IS
0640    0447  7007         7007             /10 WDS LONG
0641    0450  4002         STC 2
0642    0451  0202         XSK 2            /SCANNED INDEX ?
0643    0452  6000         JMP 0            /NO
0644    0453  6001         JMP 1
0645                    /
0646                    /       CHECK BOUNDS
0647                    /
0650    0454  1000  H1,     LDA
0651    0455  0006         6
0652    0456  2345         ADD F1
0653    0457  0470         AZE I
0654    0460  0017         COM
0655    0461  0451         APO
0656    0462  0220         XSK I 0          /NOT OK
0657    0463  6000         JMP 0            /OK
0660                    /
0661                    /
0662                    /       RCG AND CHECK INDEX
0663                    /
0664    0464  0043  CHKIND, SET 3           /RTN JMP
0665    0465  0000         0
0666    0466  1000         LDA              /PICK UP UNIT NO FROM MC PARAMS
0667    0467  2377         E6+6
0670    0470  5017         STC XIO          /STORE IN INDEX PARAMETERS
0671    0471  0002         PDP
0672                       PMODE
0673    4472  6212         CIF HI8FLD
0674    4473  4421         JMS I READ       /GET THE INDEX
0675    4474  5017         XIO
0676    4475  6141         LINC
0677                       LMODE
0700    0476  0641         LDF DATSEG       /FORCE CORRECT DATA FIELD
0701    0477  1000         LDA              /GET FIRST WORD OF INDEX
0702    0500  3000         INDEX
0703    0501  1420         SHD I            /IS THERE AN
0704    0502  5700         5700             /INDEX ?
0705    0503  0223         XSK I 3          /YES
0706    0504  6003         JMP 3            /NO
```

```
0707                    /
0710                    /
0711                    /       MAKE AN EMPTY INDEX
0712                    /
0713    0505    0041    MAKIND, SET 1
0714    0506    0000            0
0715    0507    0062            SET I 2         /POINTER TO INDEX
0716    0510    2777            INDEX-1
0717    0511    1020            LDA I           /FILL WITH
0720    0512    5757            5757            /5757 (///)
0721    0513    1062            STA I 2
0722    0514    0202            XSK 2           /DONE ?
0723    0515    6513            JMP .-2         /NO
0724    0516    6001            JMP 1
0725                    /
0726                    /
0727                    /       SEARCH FOR A NAME IN INDEX
0730                    /           NAME TO FIND IS AT E6+2
0731                    /
0732    0517    0041    LOOKUP, SET 1
0733    0520    0000            0
0734    0521    0062            SET I 2         /START OF INDEX
0735    0522    3000            INDEX
0736    0523    0063            SET I 3         /LENGTH OF NAME
0737    0524    7773            -4
0740    0525    0064            SET I 4         /START OF
0741    0526    2372            E6+1            /REQUESTED NAME
0742    0527    6444            JMP ADVIND
0743    0530    1022            LDA I 2         /CHK INDEX NAME
0744    0531    1464            SAE I 4         /WITH REQ NAME
0745    0532    6523            JMP LOOKUP+4    /NO MATCH -- TRY NEXT NAME
0746    0533    0223            XSK I 3         /MATCHED ALL ?
0747    0534    6530            JMP .-4         /NO TRY NXT PAIR
0750    0535    1020            LDA I           /INCR POINTER TO ADDR STARTING TBLK
0751    0536    0001    BUMP,   1               /3 IF THIS IS SB OR AB
0752    0537    1140            ADM             /ENTRY POINTER
0753    0540    0002            2
0754    0541    0221            XSK I 1
0755    0542    6001            JMP 1           /RTN
0756                            EJECT
-
```

```
0757                    /
0760                    /
0761                    /       DISPLAY "REPLACE?"
0762                    /
0763                                            /ASK REPLACE
0764    0543    0043    REPLAC, SET 3
0765    0544    0000            0
0766    0545    1020            LDA I           /ENABLE FULL SIZE CHARACTERS
0767    0546    0200            0200
0770    0547    0004            ESF
0771    0550    0061            SET I 1         /SET X-COORDINATE
0772    0551    0234            234
0773    0552    0064            SET I 4         /POINT TO REPLACE TEXT STRING
0774    0553    4614            RPLSTR-4000
0775    0554    1324            LDH I 4
0776    0555    1420            SHD I           /END OF MESSAGE ?
0777    0556    4300            CR1
1000    0557    6573            JMP K2          /YES CHK KBD
1001    0560    0241            ROL 1           /ADDRESS DOUBLE-WORD DISPLAY TABLE
1002    0561    1120            ADA I
1003    0562    2001            A6              /DIAL DISPLAY CHARACTER TABLE
1004    0563    4005            STC 5
1005    0564    1745            DSC 5           /DISPLAY FIRST HALF OF CHARACTER
1006    0565    1765            DSC I 5         /,,,AND SECOND HALF
1007    0566    1020            LDA I
1010    0567    0002            2
1011    0570    2001            ADD 1           /INCREMENT Y-COORDINATE
1012    0571    4001            STC 1
1013    0572    6554            JMP REPLAC+11
1014                    /
1015                    /       TEST FOR KEY STRUCK
1016                    /
1017    0573    0415    K2,     KST
1020    0574    6550            JMP REPLAC+5    /NO KBD -- REFRESH SCOPE
1021    0575    0011            CLR
1022    0576    0004            ESF             /RESET TO HALF SIZE
1023    0577    0500            IOB             /GET CHAR
1024    0600    6036            6036            /KRB
1025    0601    0500            IOB
1026    0602    6046            6046            /ECHO CHAR
1027    0603    1460            SAE I
1030    0604    0322            322             /R ?
1031    0605    6335            JMP EDRTN       /NO RETURN
1032    0606    0045            SET 5           /POINT TO OLD INDEX ENTRY
1033    0607    0002            2
1034    0610    1020            LDA I           /SLASHES TO FILL OLD SPACE
1035    0611    5757            5757
1036    0612    1045            STA 5           /ZAP OLD INDEX ENTRY START BLOCK
1037    0613    1065            STA I 5         /,,,AND LENGTH
1040    0614    6003            JMP 3           /RETURN TO DO REPLACE
1041                    /
1042    0615    2205
1042    0616    2014
1042    0617    0103
1042    0620    0577
1042                    RPLSTR, TEXT "REPLACE?"
1043    0621    4377            4377
1044                            EJECT
```

```
1045                    /
1046                    /       INDEX SLOT SEARCH
1047                    /
1050    0622    0064    ENTER,  SET I 4         /PTR TO
1051    0623    2372            E6+1            /REQ NAME
1052    0624    0041            SET 1           /RTN JMP
1053    0625    0000            0
1054    0626    0062            SET I 2         /PTR TO INDEX
1055    0627    3000            INDEX
1056    0630    6444            JMP ADVIND
1057    0631    0045            SET 5
1060    0632    0002            2
1061    0633    1022            LDA I 2
1062    0634    1460            SAE I           /FOUND A
1063    0635    5757            5757            /BLANK SLOT ?
1064    0636    6630            JMP ENTER+6     /NO TRY NXT SLOT
1065    0637    1024            LDA I 4         /PUT REQ NAME
1066    0640    1065            STA I 5         /IN INDEX
1067    0641    1520            SRO I           /SKP OUT AFT
1070    0642    3567            3567            /4 PASSES
1071    0643    6637            JMP .-4
1072    0644    0042            SET 2           /TO END OF
1073    0645    0005            5               /NAME
1074    0646    6535            JMP BUMP-1      /BUMP PTR AND EXIT
1075                    /
1076                    /
1077                    /       FIND THE BEST SPACE FOR A FILE
1100                    /
1101    0647    0050    FINDSP, SET 10          /B6 IS
1102    0650    0000            0               /DYNAMICALLY
1103    0651    0066            SET I 6         /RESET TO
1104    0652    7307            -FILE           /NEAREST TBLK
1105    0653    1020            LDA I           /SWITCH OFF
1106    0654    7777            7777            /LOWER FILE
1107    0655    4751            STC SWITCH
1110    0656    6714            JMP GAPSR       /RTN WITH B6 SET
1111                                            /TO -CLOSEST BLK
1112    0657    0047            SET 7
1113    0660    0006            6
1114    0661    0011            CLR             /SWITCH ON
1115    0662    4751            STC SWITCH      /LOWER FILE
1116    0663    0066            SET I 6
1117    0664    0270            FREE
1120    0665    6714            JMP GAPSR
1121                            EJECT
-
```

```
1122                    /
1123                    /
1124                    /       FIND NEAREST SPACE OF TWO
1125                    /
1126    0666    1000    G1,     LDA
1127    0667    0007    P7,     7               /UPPER FILE
1130    0670    0017            COM
1131    0671    2006            ADD 6           /LOWER FILE
1132    0672    1120            ADA I
1133    0673    7063            -XBLK-XBLK
1134    0674    0451            APO             /WHICH IS BEST ?
1135    0675    6700            JMP .+3         /CHK UPPER FILE
1136    0676    6454            JMP H1          /CHK LOWER FILE
1137    0677    6010            JMP 10          /OK 1ST TBLK IN AC
1140    0700    1020            LDA I
1141    0701    1000            1000
1142    0702    2007            ADD 7
1143    0703    6456            JMP H1+2        /UPPER FILE
1144    0704    6710            JMP .+4         /OK
1145    0705    6454            JMP H1          /CHK LOWER
1146    0706    6010            JMP 10          /OK
1147    0707    7071            JMP SAYNO       /FILE AREA FULL
1150    0710    1000            LDA
1151    0711    0007            7
1152    0712    0017            COM
1153    0713    6010            JMP 10
1154                                            /GAP SEARCH ALL
1155    0714    0041    GAPSR,  SET 1           /ACTIVE ENTRIES
1156    0715    0000            0               /TO SEE IF
1157    0716    0062            SET I 2         /B6 (BEST TBLK)
1160    0717    7000            INDEX!4000      /HAS TO BE RESET
1161    0720    6444            JMP ADVIND
1162    0721    1322            LDH I 2
1163    0722    1420            SHD I           /IS THERE
1164    0723    5700            5700            /AN ENTRY ?
1165    0724    6720            JMP .-4         /NO DO SOME MORE
1166    0725    0222            XSK I 2         /ADV ENTRY TO
1167    0726    0222            XSK I 2         /TBLK INFO
1170    0727    0222            XSK I 2         /SECTOR
1171    0730    6735            JMP GETBN       /RTN P+2 IF
1172                                            /INVALID
1173    0731    6757            JMP COMPBN
1174    0732    6735            JMP GETBN
1175    0733    6757            JMP COMPBN
1176    0734    6720            JMP GAPSR+4
1177                            EJECT
```

```
1200
1201        0735  1022  GETBN,  LDA I 2          /GET BN
1202        0736  1560          BCL I            /TBLK NUM
1203        0737  7000          7000             /IN LOW ORDER
1204        0740  1040          STA              /9 BITS
1205        0741  0011          11
1206        0742  1122          ADA I 2          /NUM OF TBLKS
1207        0743  0451          APO
1210        0744  0220          XSK I 0          /INACTIVE ENTRY
1211        0745  0017          COM
1212        0746  1040          STA
1213        0747  0012          12
1214        0750  1520          SRO I            /FOR UPPER OR
1215        0751  0000  SWITCH, 0                /LOWER FILE
1216        0752  6000          JMP 0
1217        0753  0052          SET 12
1220        0754  0011          11
1221        0755  4011          STC 11
1222        0756  6000          JMP 0
1223                                             /BN COMPARE
1224        0757  1000  COMPBN, LDA              /WITH B6
1225        0760  0011          11
1226        0761  2006          ADD 6
1227        0762  0470          AZE I
1230        0763  0017          COM
1231        0764  0451          APO
1232        0765  6000          JMP 0
1233        0766  2345          ADD F1
1234        0767  0451          APO
1235        0770  0470          AZE I
1236        0771  6000          JMP 0
1237        0772  0046          SET 6
1240        0773  0012          12
1241        0774  6716          JMP GAPSR+2
1242                            EJECT
-
```

```
1243                    /
1244                    /
1245                    /    RE-WRITE THE INDEX
1246                    /     ENTERED WITH STARTING TBLK IN AC
1247                    /
1250      0775  0053  RSTRX,  SET 13
1251      0776  0000          0
1252      0777  1044          STA 4             /STORE STARTING BLOCK IN INDEX
1253      1000  1560          BCL I
1254      1001  7000          7000
1255      1002  4321          STC OUTFIL+2      /START BLOCK TO PARAMETER LIST
1256      1003  2345          ADD F1            /MINUS FILE LEN
1257      1004  0017          COM
1260      1005  1064          STA I 4           /STORE LEN IN INDEX
1261      1006  0435          KST I             /LAST CHANCE TO
1262                                            /INHIBIT COMMAND
1263      1007  6335          JMP EDRTN
1264      1010  0002          PDP               /CALL WRITE IN 8-MODE
1265                          PMODE
1266      5011  6201          CDF L08FLD
1267      5012  6212          CIF HI8FLD
1270      5013  4422          JMS I WRITE       /RE-WRITE THE INDEX
1271      5014  5017          XIO
1272      5015  6141          LINC
1273                          LMODE
1274      1016  6013          JMP 13
1275                    /
1276                    /       I/O PARAMETERS FOR INDEX MANIPULATION
1277                    /
1300      1017  0000  XIO,     0                 /UNIT NUMBER
1301      1020  0006          6                 /MEM ADDR = 3000
1302      1021  0346          XBLK              /INDEX TBLK = 346
1303      1022  0002          2                 /LENGTH IS 2 BLOCKS
1304                          EJECT
-
```

```
1305                    /
1306                    /    LI, QL, OR AS (BY NAME) ENTER HERE
1307                    /
1310                    /
1311        1023  0011  LIORAS, CLR
1312        1024  0641          LDF DATSEG      /FORCE DATA FIELD TO ACCESS E6
1313        1025  0057          SET 17
1314        1026  2371          E6
1315        1027  1440          SAE             /SKIP IF LN2 IS ZERO
1316        1030  2372          E6+1
1317        1031  7035          JMP ,+4
1320        1032  1440          SAE             /SKIP IF LN1 IS ZERO
1321        1033  2371          E6
1322        1034  7071          JMP SAYNO       /BOMB IF LN2 EQ ZERO, AND LN1 NE ZERO
1323        1035  1000          LDA             /PICK UP LN1
1324        1036  2371          E6
1325        1037  0017          COM
1326        1040  1100          ADA             / LN2 - LN1
1327        1041  2372          E6+1
1330        1042  1200          LAM             /LN2 TO 17
1331        1043  0017          17
1332        1044  0452          LZE             /SKP IF LN2 GE LN1
1333        1045  7071          JMP SAYNO
1334        1046  6464          JMP CHKIND
1335        1047  7071          JMP SAYNO       /NO INDEX
1336        1050  6517          JMP LOOKUP      /SEARCH FOR NAME MATCH
1337        1051  7071          JMP SAYNO       /NO NAME MATCH
1340        1052  6436          JMP SRORBN      /SOURCE OR BINARY ?
1341        1053  0456          SKP             /SOURCE IS THERE, CONTINUE
1342        1054  7071          JMP SAYNO       /BINARY ONLY
1343        1055  1000          LDA
1344        1056  2377          E6+6
1345                    /
1346                    /  WARNING -- THIS CODE STORES IN THIS SEGMENT
1347                    /
1352        1057  4777          STC UNITNO
1351        1060  1000          LDA
1352        1061  2000          2000            /QL OR LI WD
1353        1062  4774          STC UNITNO-3
1354        1063  1002          LDA 2           /START BLOCK TO AC
1355        1064  0603          LIF KBDSEG
1356        1065  7377          JMP KBDOPR-1    /RTN TO EDITOR
1357                    EJECT
-
```

```
1360                    /
1361                    /       COME HERE TO DISPLAY "NO"
1362                    /
1363    1066    0000    NOPE,   0               /PMODE ENTRY POINT
1364    1067    6141            6141            /ENTER LINC MODE
1365    1070    7071            JMP .+1
1366    1071    0050    SAYNO,  SET 10          /SAVE RETURN FOR DEBUG
1367    1072    0000            0
1370    1073    1020            LDA I           /GET FULL SIZE
1371    1074    0200            0200
1372    1075    0004            ESF
1373    1076    0011            CLR             /Y-COORDINATE ZERO (MIDDLE SCREEN)
1374    1077    0061            SET I 1         /INITIAL X-COORDINATE
1375    1100    0344            344
1376    1101    1760            DSC I
1377    1102    3077            3077            /DISPLAY N
1400    1103    1760            DSC I
1401    1104    7706            7706
1402    1105    0061            SET I 1         /ADJUST X-COORDINATE TO NEXT CHAR
1403    1106    0374            374
1404    1107    1760            DSC I
1405    1110    4177            4177            /DISPLAY O
1406    1111    1760            DSC I
1407    1112    7741            7741
1410    1113    0415            KST             /WAS A KEY STRUCK?
1411    1114    7077            JMP SAYNO+6     /LOOP UNTIL RESPONSE
1412    1115    0011            CLR
1413    1116    0004            ESF             /RESET TO HALF SIZE
1414    1117    6335            JMP EDRTN       /THEN GO TO EDITOR
1415                            EJECT
-
```

```
1416                    /
1417                    /       ADD BINARY ENTERS HERE
1420                    /
1421    1120  0641  ADDBIN, LDF DATSEG       /FORCE DATA FIELD TO GET E6
1422    1121  1300          LDH               /FIRST CHAR OF NAME
1423    1122  2373          E6+2
1424    1123  1420          SHD I             /WAS NAME SPECIFIED?
1425    1124  7700          7700
1426    1125  7071          JMP SAYNO         /NO - CANT DO IT
1427    1126  1020          LDA I             /SET BUMP CONST
1430    1127  0003  P3,      3
1431    1130  4536          STC BUMP          /...IN LOOKUP
1432    1131  6464          JMP CHKIND        /GET INDEX, CHECK FOR VALIDITY
1433    1132  7071          JMP SAYNO         /NO INDEX - TOO BAD
1434    1133  6517          JMP LOOKUP        /FIND THE NAME
1435    1134  7071          JMP SAYNO         /THE FILE AINT THERE
1436    1135  6436          JMP SRORBN        /IS THERE A BINARY?
1437    1136  0456          SKP               /YES - READ IT
1440    1137  7071          JMP SAYNO         /NO - TOO BAD
1441                    /
1442                    /       WHEN WE GET HERE WEVE FOUND THE FILE TO ADD
1443                    /
1444    1140  1002          LDA 2             /FIRST BLOCK OF FILE...
1445    1141  1040          STA               /...IS HEADER
1446    1142  1166          FHDRIN+2
1447    1143  2347          ADD P1            /SECOND BLOCK
1450    1144  5507          STC BINIPT+2      /...IS START OF BINARY
1451    1145  1022          LDA I 2           /LENGTH...
1452    1146  3163          ADD M1            /...MINUS ONE...
1453    1147  1040          STA               /...IS BINARY BLOCK COUNT
1454    1150  5504          BLOCKS
1455    1151  0017          COM
1456    1152  2347          ADD P1            /TWOS COMPLEMENT
1457    1153  5277          STC MBLKS         /...FOR RELOCATION CONTROL
1460    1154  3017          ADD XIO           /INDEX UNIT...
1461    1155  1040          STA               /...IS ALSO FILE UNIT
1462    1156  1164          FHDRIN
1463    1157  5505          STC BINIPT
1464    1160  2667          ADD P7            /3400 SHIFTED 8
1465    1161  4324          STC HDRIO+1       /READ HEADER AT 3400
1466    1162  7200          JMP AB000         /GO TO NEXT PAGE
1467                    /
1470    1163  7776  M1,     -1
1471    1164  0000  FHDRIN, 0                 /UNIT FOR FILE
1472    1165  0006          6                 /HEADER TO 3000
1473    1166  0000          0
1474    1167  0001          1
1475                    EJECT
-
```

```
1476                            *1200
1477        1200   0002  AB000,  PDP
1500                            PMODE
1501        5201   6212          CIF HI8FLD
1502        5202   4421          JMS I READ        /READ BINARY FILE HEADER
1503        5203   5164          FHDRIN
1504        5204   4700          JMS I PFILBF      /...AND UP TO 15 BLOCKS OF FILE
1505        5205   6212          CIF HI8FLD
1506        5206   4421          JMS I READ        /READ BWA HEADER
1507        5207   4323          HDRIO
1510        5210   4263          JMS SCANHD        /SCAN HEADER FOR STANDARD RELOCATION
1511        5211   1674          TAD I PE6A        /GET RELOCATION FIELD FROM E6
1512        5212   7110          CLL RAR           /MOVE TO LINK
1513        5213   7440          SZA               /SKIP IF FIELD 0 OR 1
1514        5214   4672          JMS I PNOPE       /OTHERWISE ERROR
1515        5215   1673          TAD I PE6         /GET RELOCATION ADDR
1516        5216   7440          SZA               /IF RELOC ADDR...
1517        5217   5222          JMP .+3
1520        5220   7420          SNL               /...AND FIELD ARE ZERO...
1521        5221   5232          JMP AB010         /...DO NOT RELOCATE
1522        5222   0347          AND P377          /ADDR WITHIN BLOCK
1523        5223   3355          DCA RELADR
1524        5224   1673          TAD I PE6
1525        5225   0350          AND P7400         /BLOCK NO - HIGH ORDER BIT IN LINC
1526        5226   7006          RTL               /  MOVE
1527        5227   7006          RTL               /  TO
1530        5230   7004          RAL               /  LOW ORDER
1531        5231   3356          DCA RELBLK        /RELOCATED BLOCK ADDR
1532                            EJECT
*
```

```
1533                    /
1534                    /       NOW BEGINS THE REAL STUFF
1535                    /           SEARCH FOR NON-ZERO IN INPUT
1536                    /
1537    5232    6211    AB010,  CDF HI8FLD
1540    5233    1751    AB020,  TAD I PINDAT    /GET ONE BINARY WORD
1541    5234    7640            SZA CLA         /DO WE MOVE IT?
1542    5235    4301            JMS MOVEWD      /YUP
1543    5236    2351            ISZ PINDAT      /INCR POINTER
1544    5237    1351            TAD PINDAT      /PICK IT UP
1545    5240    0347            AND P377        /TEST FOR EOB
1546    5241    7640            SZA CLA         /SKIP IF END OF BLOCK
1547    5242    5233            JMP AB020       /ELSE LOOP
1550                    /
1551                    /       END OF INPUT BLOCK
1552                    /
1553    5243    6201            CDF LO8FLD      /INDIRECTS TO THIS FIELD
1554    5244    2277            ISZ MBLKS       /ANY MORE INPUT?
1555    5245    5254            JMP NXTBLK      /YES - GO GET EM
1556    5246    4745            JMS I PWRBLK    /THATS ALL - WRITE OUT CURRENT BUFFER
1557    5247    6212    AB030,  CIF HI8FLD
1560    5250    4422            JMS I WRITE     /...AND THE HEADER
1561    5251    4323            HDRIO
1562    5252    5653            JMP I .+1       /GO BACK TO EDIT
1563    5253    4336            EDRTN+1
1564                    /
1565                    /       GET NEXT INPUT BLOCK
1566                    /
1567    5254    7300    NXTBLK, CLA CLL         /ALL CLEAR
1570    5255    1351            TAD PINDAT      /INPUT DATA ADDR
1571    5256    1276            TAD BUFEND      /SET LINK IF END OF BUFFER
1572    5257    7630            SZL CLA         /SKIP IF MORE IN CORE
1573    5260    4700            JMS I PFILBF    /ELSE REFILL BUFFERS
1574    5261    4263            JMS SCANHD      /ADJUST RELOCATION POINTERS
1575    5262    5232            JMP AB010       /CONTINUE
1576                    /
1577                    /       SCAN INPUT HEADER TO FIND NEXT USED BLOCK
1600                    /
1601    5263    0000    SCANHD, 0
1602    5264    2356            ISZ RELBLK      /INCR RELOCATION BLOCK NO
1603    5265    2275            ISZ PINHDR      /INCR BLOCK MAP POINTER
1604    5266    1675            TAD I PINHDR    /GET A WORD OF MAP
1605    5267    7650            SNA CLA         /IS BLOCK USED?
1606    5270    5264            JMP SCANHD+1    /NO - TRY NEXT
1607    5271    5663            JMP I SCANHD    /YES - RETURN
1610                    /
1611    5272    5066    PNOPE,  NOPE            /PMODE SAYNO
1612    5273    2371    PE6,    E6
1613    5274    2372    PE6A,   E6+1
1614    5275    3340    PINHDR, 3340            /POINTER TO INPUT FILE HEADER CONTROL WORDS
1615    5276    1000    BUFEND, -7000           /CAUTION: THIS IS SNEAKY
1616    5277    0000    MBLKS,  0
1617    5300    5456    PFILBF, FILBUF
1620                    EJECT
-
```

```
1621                    /
1622                    /       WEVE GOT A NON-ZERO WORD
1623                    /
1624    5301    0000    MOVEWD, 0
1625    5302    1351            TAD PINDAT      /GET ADDR OF WORD
1626    5303    0347            AND P377        /BLOCK-RELATIVE ADDRESS
1627    5304    1355            TAD RELADR      /ADD RELOCATION
1630    5305    3352            DCA POUTDT      /OUTPUT DATA PTR
1631    5306    1352            TAD POUTDT
1632    5307    0350            AND P7400       /TEST FOR CARRY
1633    5310    7650            SNA CLA         /SKIP ON CARRRY
1634    5311    5316            JMP MW0100
1635    5312    1352            TAD POUTDT      /BLOCK ADDR...
1636    5313    0347            AND P377        /...TRUNCATED TO 8 BITS
1637    5314    3352            DCA POUTDT
1640    5315    7001            IAC             /SET CARRY
1641    5316    1356    MW0100, TAD RELBLK      /BLOCK RELOCATION
1642    5317    3353            DCA OUTBLK      /GIVES BIN BLOCK
1643    5320    1353            TAD OUTBLK
1644    5321    7041            CIA             /PREP TO COMPARE
1645    5322    1354            TAD CURBLK      /...TO CURRENT BLOCK IN BUFFER
1646    5323    7650            SNA CLA         /IN THIS BLOCK?
1647    5324    5341            JMP MOVEIT      /YES - DO THE MOVE NOW
1650    5325    6201            CDF L08FLD      /NO - GET THE RIGHT ONE
1651    5326    1354            TAD CURBLK      /ANYTHING IN BUFFER?
1652    5327    7700            SMA CLA         /SKIP IF NOT
1653    5330    4745            JMS I PWRBLK    /ELSE OUTPUT
1654    5331    1353            TAD OUTBLK      /NEXT BLOCK OF INTEREST
1655    5332    1346            TAD M40         /IS IT LEGAL?
1656    5333    7700            SMA CLA         /SKIP IF OK
1657    5334    5247            JMP AB030       /ELSE ABORT
1660    5335    1353            TAD OUTBLK      /IF OK, IT IS...
1661    5336    3354            DCA CURBLK      /...NEW CURRENT
1662    5337    1353            TAD OUTBLK
1663    5340    4744            JMS I PGETBK    /GET BWA BLOCK INTO BUFFER
1664                    /
1665                    /       NOW, MOVE ONE WORD
1666                    /
1667    5341    1751    MOVEIT, TAD I PINDAT
1670    5342    3752            DCA I POUTDT
1671    5343    5701            JMP I MOVEWD    /AND RETURN
1672                    /
1673                    /
1674    5344    5400    PGETBK, GETBLK
1675    5345    5427    PWRBLK, WRTBLK
1676    5346    7740    M40,    -40
1677    5347    0377    P377,   377
1700    5350    7400    P7400,  7400
1701    5351    0000    PINDAT, 0
1702    5352    0000    POUTDT, 0
1703    5353    0000    OUTBLK, 0
1704    5354    7777    CURBLK, -1
1705    5355    0000    RELADR, 0
1706    5356    7777    RELBLK, -1
1707                    EJECT
-
```

```
1710                            *5400
1711                    /
1712                    /       FILL BWA BUFFER
1713                    /
1714    5400    0000    GETBLK, 0
1715    5401    3254            DCA BWAIO+2     /BLOCK WANTED
1716    5402    1254            TAD BWAIO+2
1717    5403    1251            TAD POUTHD      /ADDRESS CTL WD
1720    5404    3246            DCA TMP2
1721    5405    1646            TAD I TMP2      /GET CONTROL WD
1722    5406    7650            CLA SNA         /SKIP IF BLK USE
1723    5407    5216            JMP CLRBUF      /ELSE CLEAR BUFF
1724                    /
1725                    /       READ THE APPROPRIATE BLOCK OF BWA INTO WORKING BUFFER
1726                    /
1727    5410    6212            CIF HI8FLD
1730    5411    4421            JMS I READ      /GET IT
1731    5412    5452            BWAIO
1732    5413    6211            CDF HI8FLD      /SET UPPER FIELD FOR MOVE
1733    5414    7200            CLA
1734    5415    5600            JMP I GETBLK    /RETURN
1735                    /
1736                    /       THIS BLOCK OF BWA IS NOT USED
1737                    /           CLEAR THE BUFFER
1740                    /
1741    5416    1247    CLRBUF, TAD M400        /SET COUNT
1742    5417    3246            DCA TMP2
1743    5420    7040            CMA             /-1 TO AC
1744    5421    3010            DCA AUTO        /SET AUTO-INDEX
1745    5422    6211            CDF HI8FLD
1746    5423    3410            DCA I AUTO
1747    5424    2246            ISZ TMP2
1750    5425    5223            JMP .-2         /LOOP TILL CLEAR
1751    5426    5600            JMP I GETBLK    /THEN RETURN
1752                    /
1753                    /       WRITE THE CURRENT BLOCK TO BWA
1754                    /
1755    5427    0000    WRTBLK, 0
1756    5430    6212            CIF HI8FLD
1757    5431    4422            JMS I WRITE     /SEND IT OUT
1760    5432    5452            BWAIO
1761    5433    7200            CLA
1762    5434    1254            TAD BWAIO+2     /BLOCK NO...
1763    5435    1251            TAD POUTHD      /...PLUS HEADER ADDRESS...
1764    5436    3246            DCA TMP2        /...IS ADDR OF CONTROL WORD
1765    5437    1646            TAD I TMP2      /GET IT
1766    5440    7640            SZA CLA         /TEST FOR ALREADY USED
1767    5441    5627            JMP I WRTBLK    /YES - RETURN
1770    5442    7040            CMA             /NO - UPDATE HDR
1771    5443    3646            DCA I TMP2      /SET USED INDICATOR
1772    5444    2650            ISZ I PBUSE     /INCR BLOCKS USED COUNT
1773    5445    5627            JMP I WRTBLK    /NOW RETURN
1774                    /
1775    5446    0000    TMP2,   0
1776    5447    7400    M400,   -400
1777    5450    3737    PBUSE,  USENO!2000
2000    5451    3740    POUTHD, MAP!2000
2001    5452    0111    BWAIO,  111             /BWA UNIT
2002    5453    0020            BUFFER          /ADDR = 1 0000
2003    5454    0000            0               /BLOCK NO
2004    5455    0001            1               /ONE AT A TI
2005                            EJECT
```

```
2006                    /
2007                    /       FILL BUFFERS FROM THE BINARY FILE
2010                    /
2011        5456  0000  FILBUF, 0
2012        5457  1304          TAD BLOCKS      /GET REMAINING COUNT
2013        5460  1301          TAD M15         /SUBTRACT BUFFER LENGTH
2014        5461  7500          SMA             /SKIP IF NOT ENOUGH TO FILL BUFFER
2015        5462  5266          JMP FB010       /ELSE READ 15
2016        5463  7200          CLA
2017        5464  1304          TAD BLOCKS      /REMAINING BLKS
2020        5465  3310          DCA BINIPT+3    /TO I/O LIST
2021        5466  3304  FB010,  DCA BLOCKS      /STORE NEW REMAINING COUNT
2022        5467  6212          CIF HI8FLD
2023        5470  4421          JMS I READ      /READ UP TO 15
2024        5471  5505          BINIPT
2025        5472  7200          CLA
2026        5473  1307          TAD BINIPT+2    /ADJUST STRT BLK
2027        5474  1310          TAD BINIPT+3    /BY LENGTH
2030        5475  3307          DCA BINIPT+2
2031        5476  1302          TAD INSTRT      /START OF FIRST BLOCK READ
2032        5477  3703          DCA I PPINDT    /RESET POINTER
2033        5500  5656          JMP I FILBUF    /RETURN
2034                    /
2035        5501  7763  M15,    -15
2036        5502  0400  INSTRT, 400
2037        5503  5351  PPINDT, PINDAT
2040        5504  0000  BLOCKS, 0
2041        5505  0000  BINIPT, 0               /UNIT WITH FILE
2042        5506  0021          BUFFER+1        /ADDR = 1 0400
2043        5507  0000          0               /START BLOCK
2044        5510  0015          15              /NO TO READ
2045                            EJECT
-
```

```
                /
                /
                /     SPECIAL SYMBOLS
                /
                /
                        WA=370
                        FILE=470            /START OF UPPER FILE AREA
                        E6=2371             /MC PARAMETER LIST
                        UNITNO=777
                        XBLK=346            /TBLK OF DIAL INDEX
                        INDEX=3000          /CORE ADDR OF DIAL INDEX
                        FREE=270            /END+1 OF LOWER FILE AREA
                        A6=2001             /START OF DIAL CHARACTER TABLE
                        HEADER=1400         /CORE ADDR OF HEADER BLOCK
                        USENO=HEADER+337 /COUNT OF BLOCKS USED IN BWA
                        MAP=HEADER+340  /START OF BLOCK USAGE MAP IN HEADER
                        KBDOPR=1400         /GO HERE AFTER ASSEMBLY PRE-PROCESSING
                        PSENT=1000          /ENTRY POINT OF PS MAIN CODE
                        DRSTRT=7777         /UPPER SEGMENT DIAL RESTART ADDR
                        CR1=4300
                        DIALU=100           /PSEUDO-UNIT CONTAINING DIAL
                        PSBLK=63            /START BLOCK OF PS MAIN CODE ON DIAL UNIT
                        SWA=110             /SOURCE WORK AREA UNIT
                        BWA=111             /BINARY UNIT
                        HDRBLK=57           /TBLK OF HEADER IN BWA
                        AUTO=10             /AUTO-INDEX REGISTER USED BY SB
                        READ=21             /PAGE ZERO ADDRESSES,,,
                        WRITE=22            /,,,FOR I/O HANDLERS
                        BUFFER=20           /BUFFER ADDRESS (1 0000) SHIFTED RIGHT 8
                        BUFLEN=16           /NO OF BLOCKS FOR BUFFER
                        HI8FLD=10           /PDP-8 FIELD CONTAINING I/O AND DIAL RESTART
                        LO8FLD=0            /PDP-8 FIELD CONTAINING FCOMS
                        DATSEG=1            /LINC DATA SEGMENT USED BY FCOMS
                        KBDSEG=3            /INSTRUCTION SEGMENT CONTAINING KBDOPR
                /
                        ASMIFM 6000-.
                WARNING // IF THIS ROUTINE EXCEEDS 2000 WORDS OCTAL, EDITOR MUST BE MODIFIED
                / TO READ MORE BLOCKS, AND LAP6-DIAL MUST HAVE SPACE FOR THEM.
                /
                /
                /     END OF FCOMSV4
                /
```

NO ERRORS

```
SYMBOL   VALUE  DEF   REFERENCES

AB00     5200  1477  1466
AB10     5232  1537  1521 1575
AB20     5233  1540  1547
AB30     5247  1557  1657
ADDBIN   5120  1421  0044
ADVIND   4444  0634  0742 1056 1161
AUTO     0010  2077  0263 0271 1744 1746
A6       2001  2062  1003
BFAD     4332  0447  0350
BFLN     4331  0446  0352 0366 0413 0416
BINIPT   5505  2041  1450 1463 2020 2024 2026 2027 2030
BLOCKS   5504  2040  1454 2012 2017 2021
BUFEND   5276  1615  1571
BUFFER   0020  2102  0426 0433 0447 0450 0451 2002 2042
BUFLEN   0016  2103  0430 0435 0446 0451
BUMP     4536  0751  0223 1074 1431
BWA      0111  2075  0242 0437
BWAIO    5452  2001  1715 1716 1731 1760 1762
CHKIND   4464  0663  0224 0507 0543 1334 1432
CLRBUF   5416  1741  1726
COMPBN   4757  1224  1173 1175
CR1      4300  2071  0777
CURBLK   5354  1704  1645 1651 1661
DATSEG   0001  2106  0067 0502 0535 0575 0677 1312 1421
DIALU    0100  2072  0603
DRSTRT   7777  2070  0461
EDRTN    4335  0456  0343 0412 1031 1263 1414 1563
ENTER    4622  1050  0233 0516 1064
E6       2371  2055  0073 0112 0123 0143 0153 0162 0171 0504 0537 0560 0666 0741 1051 1314 1316 1321 1324 1327 1344 1423 1612 1
FB10     5466  2021  2015
FCSA     4020  0043
FHDRIN   5164  1471  1446 1462 1503
FILBUF   5456  2011  1617 2033
FILE     0470  2054  1104
FINDSP   4647  1101  0237 0522
FREE     0270  2061  1117
F1       4345  0476  0217 0651 1233 1256
GAPSR    4714  1155  1110 1120 1176 1241
GETBLK   5400  1714  1674 1734 1751
GETBN    4735  1201  1171 1174
GETPS    4421  0566  0550
G1       4666  1126
HDRBLK   0057  2076  0441
HDRIO    4523  0437  0064 0246 0255 0334 1465 1507 1561
HDRSW    4330  0445  0326 0331
HEADER   1400  2063  0071 0121 0126 0136 0141 0151 0154 2064 2065
HI8FLD   0010  2104  0062 0274 0332 0335 0401 0404 0457 0570 0672 1267 1501 1505 1537 1557 1727 1732 1745 1756 2022
H1       4454  0647  1136 1143 1145
INDEX    3000  2060  0701 0716 0735 1055 1160
INFIL    4513  0425  0243 0252 0276 0306 0314 0315 0317 0351 0355 0375 0376 0403 0414 0415
INSTRT   5502  2036  2031
KBDOPR   1400  2066  1356
KBDSEG   0003  2107  1355
K2       4573  1017  1000
LIORAS   5023  1311  0047
LOOKUP   4517  0732  0226 0511 0545 0745 1336 1434
LO8FLG   0000  2105  0061 1266 1553 1650
MAKIN    4505  0713  0225 0510
MAP      1740  2065  0176 0260 2000
MBCNT    4327  0444  0221 0300 0341 0367 0373 0400 0410 0501
```

```
SYMBOL  VALUE  DEF   REFERENCES

MBFEND  4334   0451  0316
MBLKS   5277   1616  1457 1554
MOVEIT  5341   1667  1647
MOVEWD  5301   1624  1542 1671
MW010   5316   1641  1634
M1      5163   1470  1452
M15     5501   2035  2013
M40     5346   1676  1655
M400    5447   1776  1741
NOPE    5066   1363  1611
NXTBLK  5254   1567  1555
OUTBLK  5353   1703  1642 1643 1654 1660 1662
OUTFIL  4317   0432  0244 0250 0256 0307 0337 0353 0354 0377 0406 0417 0420 0525 1255
PBUSE   5450   1777  1772
PDPMOD  4070   0134  0100
PE6     5273   1612  1515 1524
PE6A    5274   1613  1511
PFILBF  5300   1617  1504 1573
PGETBK  5344   1674  1663
PINDAT  5351   1701  1540 1543 1544 1570 1625 1667 2037
PINHDR  5275   1614  1603 1604
PNOPE   5272   1611  1514
POUTDT  5352   1702  1630 1631 1635 1637 1670
POUTHD  5451   2000  1717 1763
PPINDT  5503   2037  2032
PRNPRG  4376   0535  0050
PSBLK   0063   2073  0605
PSENT   1000   2067  0577
PSIN    4431   0603  0572
PSWA    4413   0555  0542
PWRBLK  5345   1675  1556 1653
P1      4347   0500  0214 0247 0251 1447 1456
P2      4161   0236  0220
P3      5127   1430  0222
P377    5347   1677  1522 1545 1626 1636
P7      4667   1127  1464
P7400   5350   1700  1525 1632
READ    0021   2100  0063 0275 0402 0571 0673 1502 1506 1730 2023
RELADR  5355   1705  1523 1627
RELBLK  5356   1706  1531 1602 1641
REPLAC  4543   0764  0231 0514 1013 1020
RPLSTR  4615   1042  0774
RSTRX   4775   1250  0240 0523
SAVBIN  4026   0057  0043
SAVPRG  4541   0472  0046
SAYNO   5071   1366  0167 0203 0213 0234 0506 0517 0544 0546 0551 1147 1322 1333 1335 1337 1342 1411 1426 1433 1435 1440
SB010   4111   0161  0127
SB020   4134   0211  0076
SB030   4156   0233  0227
SB040   4160   0235  0232
SB100   4164   0241
SB110   4206   0271  0356
SB120   4223   0314  0301
SB130   4231   0326  0310
SB140   4240   0335  0330
SB160   4254   0355  0273 0321
SCANHD  5263   1601  1510 1574 1606 1607
SETFLD  4060   0120  0110
SP020   4365   0516  0512
```

```
SYMBOL   VALUE  DEF   REFERENCES

SP120    4271   0400  0371
SRORBN   4436   0621  0230 0513 0547 1340 1436
SWA      0110   2074  0425 0556
SWITCH   4751   1215  1107 1115
TMP2     5446   1775  1720 1721 1742 1747 1764 1765 1771
UNITNO   0777   2056  1350 1353
USENO    1737   2064  0211 1777
WA       0370   2053  0473
WRITE    0022   2101  0333 0336 0405 1270 1560 1757
WRTBLK   5427   1755  1675 1767 1773
XBLK     0346   2057  1133 1133 1302
XIO      5017   1300  0253 0524 0667 0674 1271 1460
ZERO     4435   0610  0562
```