F 1

```
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBBB    UUU         UUU     GGGGGGGGGGGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBBB    UUU         UUU     GGGGGGGGGGGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBBB    UUU         UUU     GGGGGGGGGGGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEEEEEEEEEEE     BBBBBBBBBBBB     UUU         UUU  GGG
DDD         DDD  EEEEEEEEEEEE     BBBBBBBBBBBB     UUU         UUU  GGG
DDD         DDD  EEEEEEEEEEEE     BBBBBBBBBBBB     UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG   GGGGGGGGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG  GGGGGGGGGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG  GGGGGGGGGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG         GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG         GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG         GGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUU     GGGGGGGGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUU     GGGGGGGGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUU     GGGGGGGGG
```

FILEID**DBGSOURCE

```
DDDDDDDD    BBBBBBBB      GGGGGGGG    SSSSSSSS      OOOOOO    UU        UU  RRRRRRRR      CCCCCCCC  EEEEEEEEEE
DDDDDDDD    BBBBBBBB      GGGGGGGG    SSSSSSSS      OOOOOO    UU        UU  RRRRRRRR      CCCCCCCC  EEEEEEEEEE
DD      DD  BB      BB  GG          SS          OO      OO  UU        UU  RR      RR  CC          EE
DD      DD  BB      BB  GG          SS          OO      OO  UU        UU  RR      RR  CC          EE
DD      DD  BB      BB  GG          SS          OO      OO  UU        UU  RR      RR  CC          EE
DD      DD  BB      BB  GG          SS          OO      OO  UU        UU  RR      RR  CC          EE
DD      DD  BBBBBBBB    GG            SSSSSS    OO      OO  UU        UU  RRRRRRRR    CC          EEEEEEEE
DD      DD  BBBBBBBB    GG            SSSSSS    OO      OO  UU        UU  RRRRRRRR    CC          EEEEEEEE
DD      DD  BB      BB  GG    GGGGGG        SS  OO      OO  UU        UU  RR    RR    CC          EE
DD      DD  BB      BB  GG    GGGGGG        SS  OO      OO  UU        UU  RR    RR    CC          EE
DD      DD  BB      BB  GG        GG        SS  OO      OO  UU        UU  RR      RR  CC          EE
DD      DD  BB      BB  GG        GG        SS  OO      OO  UU        UU  RR      RR  CC          EE          ....
DDDDDDDD    BBBBBBBB      GGGGGG    SSSSSSSS      OOOOOO    UUUUUUUUUU  RR      RR    CCCCCCCC  EEEEEEEEEE  ....
DDDDDDDD    BBBBBBBB      GGGGGG    SSSSSSSS      OOOOOO    UUUUUUUUUU  RR      RR    CCCCCCCC  EEEEEEEEEE  ....

LL          IIIIII      SSSSSSSS
LL          IIIIII      SSSSSSSS
LL            II      SS
LL            II      SS
LL            II      SS
LL            II      SS
LL            II        SSSSSS
LL            II        SSSSSS
LL            II              SS
LL            II              SS
LL            II              SS
LL            II              SS
LLLLLLLLLL  IIIIII    SSSSSSSS
LLLLLLLLLL  IIIIII    SSSSSSSS
```

```
0 MODULE DBGSOURCE (IDENT = 'V04-000') =
0
1 BEGIN
1
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 ! WRITTEN BY
1 !       Bert Beander    August, 1981
1 !       Ping Sager
1 !       Rich Title
1 !
1 ! MODULE FUNCTION
1 !       This module contains all routines connected with DEBUG's source line
1 !       display feature.  It thus implements the TYPE, EXAMINE/SOURCE, SET
1 !       SOURCE, SHOW SOURCE, and CANCEL SOURCE commands, and it contains all
1 !       routines which look up source lines and open and read source files.
1 !
1
1 REQUIRE 'SRC$:DBGPROLOG.REQ';
1
1 FORWARD ROUTINE
1     DBG$SRC_CANCEL_SOURCE: NOVALUE,     ! Handle the CANCEL SOURCE command
1     DBG$SRC_CLOSE_FILE: NOVALUE,        ! Close a source file
1     DBG$SRC_INIT: NOVALUE,              ! Initialize data structures at DEBUG
1                                         !       initialization time
1     DBG$SRC_LINE_TO_REC: NOVALUE,       ! Translate line number to record number
1     DBG$SRC_LNUM_RANGE: NOVALUE,        ! Returns a module's line number range
1     DBG$SRC_OPEN,                       ! Actually does the open and verifies
1                                         !       the attributes of a source file
1     DBG$SRC_OPEN_FILE: NOVALUE,         ! Open a source file after applying all
1                                         !       directory search rules
1     DBG$SRC_OUTPUT_LINE: NOVALUE,       ! Output a source line in proper format
1     DBG$SRC_READ_FILE: NOVALUE,         ! Read a source line from a source file
1     DBG$SRC_SEARCH,                     ! Actually does the search
1     DBG$SRC_SEARCH_CMD: NOVALUE,        ! Handle the SEARCH command
1     DBG$SRC_SET_MAX_FILES: NOVALUE,     ! Set maximum number of source files
```

```
1                                           !       that can be open at one time
1       DBG$SRC_SET_SOURCE: NOVALUE,        ! Handle the SET SOURCE command
1       DBG$SRC_SHOW_SOURCE: NOVALUE,       ! Handle the SHOW SOURCE command
1       DBG$SRC_TYPE_LINE,                  ! Retrieve and type out a source line
1       DBG$SRC_TYPE_LNUM_SOURCE: NOVALUE,  ! Type a set of source lines given a
1                                           !       module and a line number range
1       DBG$SRC_TYPE_PC_SOURCE: NOVALUE,    ! Type a set of source lines given a
1                                           !       Program Counter range
1       OUTPUT_DIR_LIST : NOVALUE,          ! Outputs a source directory list at
1                                           ! the terminal.
1       TYPE_PC_SOURCE_HANDLER;             ! Handler to catch the message
```

```
EXTERNAL ROUTINE
     DBG$FORMAT_FAO_OUT: NOVALUE,          ! Build an output buffer for signaling
     DBG$GET_MEMORY,                       ! Get a permanent memory block
     DBG$GET_TEMPMEM,                      ! Get a tempory memory block
     DBG$NEWLINE: NOVALUE,                 ! Output the current print buffer and
                                           !      initialize a new print line
     DBG$PC_TO_LINE_LOOKUP,                ! Matches an absolute PC address to a
                                           !      line number
     DBG$PRINT: NOVALUE,                   ! Output text to the print buffer using
                                           !      FAO formatting
     DBG$REL_MEMORY: NOVALUE,              ! Release a permanent memory block
     DBG$SCR_SOURCE_BEGIN: NOVALUE,        ! Begin source output to screen display
     DBG$SCR_SOURCE_END: NOVALUE,          ! End source output to screen display
     DBG$SCR_SOURCE_LINE: NOVALUE,         ! Output a source line to a screen display
     DBG$STA_SYMNAME: NOVALUE,             ! Get module name for a given module RST
                                           !      pointer
     LBR$CLOSE,                            ! Closes an open library
     LBR$FIND,                             ! Looks up a key by RFA in preparation
                                           !      for reading the key's associated
                                           !      text
     LBR$GET_RECORD,                       ! Reads a text record associated with a
                                           !      module in the library
     LBR$LOOKUP_KEY,                       ! Looks up a module in a text library
     LBR$INI_CONTROL,                      ! Initializes a library index for use
                                           !      by all other routines
     LBR$OPEN,                             ! Opens an existing library
     LBR$SET_MODULE,                       ! Reads an module header in a text library
     STR$UPCASE,                           ! Uppercase Conversion
     SYS$FAOL;                             ! Formatted ASCII Output routine to
                                           !      do text formatting

EXTERNAL
     DBG$GL_CHARTBL: REF CHRTBL$TABLE,     ! Pointer to DBGPARSER's Character Table
                                           !      for the currently set language
     DBG$GB_LANGUAGE: BYTE,                ! The current language code
     DBG$GL_SCREEN_SOURCE;                 ! Screen Display ID for source or zero

LITERAL
     MAX_MAX_FILES        = 20,            ! Maximum number of Source File Control
                                           !      Blocks we allow user to create
     NAMBUFSIZE           = 256,           ! Size in bytes of name buffers pointed
                                           !      to by the RMS NAM block
     SEARCH_IDENT_ALL     = 1,             ! SEARCH/IDENT/ALL is active
     SEARCH_IDENT_NEXT    = 2,             ! SEARCH/IDENT/NEXT is active
     SEARCH_STRING_ALL    = 3,             ! SEARCH/STRING/ALL is active
     SEARCH_STRING_NEXT   = 4,             ! SEARCH/STRING/NEXT is active
     SRCBUFSIZE           = 256,           ! Size in bytes of source record buffer
     RFATBL_SIZE          = 300,           ! Size in bytes of the RFA table
     RFATBL_ENTRIES       = 50,            ! Number of entries in the RFA table
     INIT_RFATBL_SPACING  = 5;             ! Number of source records per RFA
                                           ! table entry.

GLOBAL
     DBG$SRC_LEFT_MARGIN : INITIAL(1),     ! Left margin of source output
     DBG$SRC_RIGHT_MARGIN : INITIAL(255),  ! Right margin of source output
     DBG$SRC_TERM_WIDTH: INITIAL(80),      ! Terminal width for source line display
     DBG$SRC_MAX_FILES: INITIAL(5),        ! Maximum number of source files to be
```

```
                                                ! kept open at any one time
    DBG$SRC_NEXT_LNUM: INITIAL(1),              ! Next desired line in the current
                                                !       module to be typed
    DBG$SRC_NEXT_MODRSTPTR: INITIAL(0),         ! Associated module RST pointer
    DBG$SRC_NEXT_STMT: INITIAL(0),              ! Next desired statement number in the
                                                !       current module to be typed
    DBG$SRC_SEARCH_STRING: VECTOR[SRCBUFSIZE,BYTE]
        INITIAL (BYTE (0, REP (SRCBUFSIZE - 1) OF (%C' ')));
                                                ! Contains a counted ASCII search string
                                                !       filled by SEARCH Command.


OWN
    DBG$SRC_DIR_LIST: REF SDSL$HEADER           ! Pointer to linked list of Source Dir-
                           INITIAL(0),          !       ectory Search List Header blocks
    DBG$SRC_FORMFEED_FLAG: INITIAL(0),          ! Flag set to indicate not to skip
                                                !       form feed
    DBG$SRC_MODRSTPTR: REF RST$ENTRY,           ! Pointer to the Module RST Entry for
                                                !       the "current module", the module
                                                !       we are now reading source from
    DBG$SRC_SEARCH_FLAG,                        ! Set to SEARCH_FLAG's value which
                                                !       indicate whether the SEARCH
                                                !       command is in action and the
                                                !       kind of search option is used
    DBG$SRC_SFCB_PTR: REF SFCB$BLOCK            ! Pointer to first Source File Control
                           INITIAL(0),          !       Block on doubly linked list of
                                                !       such blocks
    DBG$SRC_REC_BUF: VECTOR[SRCBUFSIZE,         ! Buffer area to hold the source file
                            BYTE],              !       record
    SOURCE_TO_SCREEN_FLAG;                      ! Flag used to direct source output to
                                                !       a source screen display
```

```
1 GLOBAL ROUTINE DBG$SRC_CANCEL_SOURCE(MODRSTPTR): NOVALUE =
1 !
1 ! FUNCTION
1 !       This routine handles the semantic processing of the CANCEL SOURCE and
1 !       CANCEL SOURCE/MODULE=xxx commands.  It removes the directory list for
1 !       the specified module, or the default directory list if the /MODULE
1 !       qualifier was not present, from the list of directory search lists
1 !       pointed to by DBG$SRC_DIR_LIST.  It thus cancels the effect of the
1 !       SET SOURCE or SET SOURCE/MODULE=xxx command which created that direct-
1 !       ory search list.
1 !
1 ! INPUTS
1 !       MODRSTPTR - A pointer to the Module RST Entry of the module specified
1 !                   on the CANCEL SOURCE/MODULE command.  If the /MODULE quali-
1 !                   fier was not present on the command, MODRSTPTR must be zero.
1 !
1 ! OUTPUTS
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2       MODRSTPTR: REF RST$ENTRY;         ! Pointer to Module RST Entry or zero
2
2     LOCAL
2       HDRPTR : REF SDSL$HEADER,         ! Pointer to a Source Directory Search
2                                         !       List Header Block.
2                                         !       This pointer is used to walk
2                                         !       through the linked list of
2                                         !       header blocks.
2       ENTRY_PTR : REF SDSL$ENTRY,       ! Pointer to a Source Directory Search
2                                         !       List Entry.This pointer is
2                                         !       used to walk through the linked
2                                         !       list of entries, once the
2                                         !       appropriate source directory
2                                         !       search list is found.
2       MODNAMEPTR,                       ! Points to a counted string with the
2                                         !       module name corresponding to
2                                         !       MODRSTPTR. (Needed in case an
2                                         !       error message is signalled).
2       PREV_HDRPTR : REF SDSL$HEADER,    ! This variable contains the previous
2                                         !       Source Directory Search List Header
2                                         !       as we walk through the linked
2                                         !       list. We need to remember this
2                                         !       so we can unlink the header
2                                         !       being cancelled.
2       TEMP_PTR;                         ! Pointer to a source directory list
2                                         !       entry. Used during freeing up
2                                         !       of space.
2
2     ! Attempt to find a directory list for the module specified by MODRSTPTR.
2     !
2     PREV_HDRPTR = DBG$SRC_DIR_LIST;
2     HDRPTR = .DBG$SRC_DIR_LIST;
2     WHILE .HDRPTR NEQ 0 DO
```

```
3           BEGIN
3           IF .HDRPTR [SDSL$L_MODPTR] EQL .MODRSTPTR
3           THEN
3               EXITLOOP;
3           PREV_HDRPTR = .HDRPTR;
3           HDRPTR = .HDRPTR[SDSL$L_FLINK];
2           END;
2
2
2       ! If no source directory search list header block was found for the
2       ! specified module, signal an error to the user.
2       !
2       IF .HDRPTR EQL 0
2       THEN
3           BEGIN
3           IF .MODRSTPTR NEQ 0
3           THEN
4               BEGIN
4               DBG$STA_SYMNAME (.MODRSTPTR, MODNAMEPTR);
4               SIGNAL (DBG$_NODIRLISM, 1, .MODNAMEPTR);
4               END
4
3           ELSE
3               SIGNAL (DBG$_NODIRLIST);
3
2           END;
2
2
2       ! Unlink the Source Directory Search List Header Block for the
2       ! MODRSTPTR module from the Source Directory Search List. Then
2       ! release that block and all associated Source Directory Search
2       ! List Entries to the memory pool
2       !
2       PREV_HDRPTR[SDSL$L_FLINK] = .HDRPTR[SDSL$L_FLINK];
2       ENTRY_PTR = .HDRPTR[SDSL$L_LIST_PTR];
2       WHILE .ENTRY_PTR NEQ 0 DO
3           BEGIN
3           TEMP_PTR = .ENTRY_PTR[SDSL$L_ENT_FLINK];
3           DBG$REL_MEMORY (.ENTRY_PTR);
3           ENTRY_PTR = .TEMP_PTR;
2           END;
2       DBG$REL_MEMORY (.HDRPTR);
2
2
2       ! Reinitialize all Source File Control Blocks. This forces all open source
2       ! files to be reinitialized when accessed so that the modified Source
2       ! Directory Search List will be used to select and open those files.
2       !
2       DBG$SRC_INIT();
2
2       RETURN;
2
1       END;


                                                        .TITLE  DBGSOURCE
                                                        .IDENT  \V04-000\
```

```
                                   .PSECT  DBG$OWN,NOEXE,  PIC,2

00000000  00000 DBG$SRC_DIR_LIST:
                                   .LONG   0                                  ;
00000000  00004 DBG$SRC_FORMFEED_FLAG:
                                   .LONG   0                                  ;
          00008 DBG$SRC_MODRSTPTR:
                                   .BLKB   4
          0000C DBG$SRC_SEARCH_FLAG:
                                   .BLKB   4
00000000  00010 DBG$SRC_SFCB_PTR:
                                   .LONG   0                                  ;
          00014 DBG$SRC_REC_BUF:
                                   .BLKB   256
          00114 SOURCE_TO_SCREEN_FLAG:
                                   .BLKB   4

                                   .PSECT  DBG$GLOBAL,NOEXE,  PIC,2

00000001  00000 DBG$SRC_LEFT_MARGIN::
                                   .LONG   1                                  ;
000000FF  00004 DBG$SRC_RIGHT_MARGIN::
                                   .LONG   255                                ;
00000050  00008 DBG$SRC_TERM_WIDTH::
                                   .LONG   80                                 ;
00000005  0000C DBG$SRC_MAX_FILES::
                                   .LONG   5                                  ;
00000001  00010 DBG$SRC_NEXT_LNUM::
                                   .LONG   1                                  ;
00000000  00014 DBG$SRC_NEXT_MODRSTPTR::
                                   .LONG   0                                  ;
00000000  00018 DBG$SRC_NEXT_STMT::
                                   .LONG   0                                  ;
       00  0001C DBG$SRC_SEARCH_STRING::
                                   .BYTE   0                                  ;
      20#  0001D                   .BYTE   32[255]                            ;

                                   .EXTRN  DBG$FORMAT_FAO_OUT
                                   .EXTRN  DBG$GET_MEMORY, DBG$GET_TEMPMEM
                                   .EXTRN  DBG$NEWLINE, DBG$PC_TO_LINE_LOOKUP
                                   .EXTRN  DBG$PRINT, DBG$REL_MEMORY
                                   .EXTRN  DBG$SCR_SOURCE_BEGIN
                                   .EXTRN  DBG$SCR_SOURCE_END
                                   .EXTRN  DBG$SCR_SOURCE_LINE
                                   .EXTRN  DBG$STA_SYMNAME
                                   .EXTRN  LBR$CLOSE, LBR$FIND
                                   .EXTRN  LBR$GET_RECORD, LBR$LOOKUP_KEY
                                   .EXTRN  LBR$INI_CONTROL
                                   .EXTRN  LBR$OPEN, LBR$SET_MODULE
                                   .EXTRN  STR$UPCASE, SYS$FAOL
                                   .EXTRN  DBG$GL_CHARTBL, DBG$GB_LANGUAGE
                                   .EXTRN  DBG$GL_SCREEN_SOURCE

                                   .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

     00FC  00000                   .ENTRY  DBG$SRC_CANCEL_SOURCE, Save R2,R3,R4,R5,R6,-; 0290
```

```
                                                        R7                                      ;
          57 00000000'   EF   9E 00002           MOVAB  DBG$SRC_DIR_LIST, R7                    ;
          56 00000000G   00   9E 00009           MOVAB  DBG$REL_MEMORY, R6                      ;
          55 00000000G   00   9E 00010           MOVAB  LIB$SIGNAL, R5                          ;
          5E                  04   C2 00017      SUBL2  #4, SP                                  ;
          53                  67   9E 0001A      MOVAB  DBG$SRC_DIR_LIST, PREV_HDRPTR           ; 0344
          52                  67   D0 0001D      MOVL   DBG$SRC_DIR_LIST, HDRPTR                ; 0345
                              0F   13 00020 1$:  BEQL   2$                                      ; 0346
     04   AC        08        A2   D1 00022      CMPL   8(HDRPTR), MODRSTPTR                    ; 0348
                              08   13 00027      BEQL   2$                                      ;
          53                  52   D0 00029      MOVL   HDRPTR, PREV_HDRPTR                     ; 0351
          52                  62   D0 0002C      MOVL   (HDRPTR), HDRPTR                        ; 0352
                              EF   11 0002F      BRB    1$                                      ; 0346
                              52   D5 00031 2$:  TSTL   HDRPTR                                  ; 0359
                              29   12 00033      BNEQ   4$                                      ;
                    04        AC   D5 00035      TSTL   MODRSTPTR                               ; 0362
                              1B   13 00038      BEQL   3$                                      ;
                              5E   DD 0003A      PUSHL  SP                                      ; 0365
                    04        AC   DD 0003C      PUSHL  MODRSTPTR                               ;
 00000000G  00                02   FB 0003F      CALLS  #2, DBG$STA_SYMNAME                     ;
                              6E   DD 00046      PUSHL  MODNAMEPTR                              ; 0366
                              01   DD 00048      PUSHL  #1                                      ;
               00028CE8       8F   DD 0004A      PUSHL  #167144                                 ;
            65                03   FB 00050      CALLS  #3, LIB$SIGNAL                          ;
                              09   11 00053      BRB    4$                                      ; 0362
               00028CF0       8F   DD 00055 3$:  PUSHL  #167152                                 ; 0370
            65                01   FB 0005B      CALLS  #1, LIB$SIGNAL                          ;
            63                62   D0 0005E 4$:  MOVL   (HDRPTR), (PREV_HDRPTR)                 ; 0380
            53      04        A2   D0 00061      MOVL   4(HDRPTR), ENTRY_PTR                    ; 0381
                              0D   13 00065 5$:  BEQL   6$                                      ; 0382
            54                63   D0 00067      MOVL   (ENTRY_PTR), TEMP_PTR                   ; 0384
                              53   DD 0006A      PUSHL  ENTRY_PTR                               ; 0385
            66                01   FB 0006C      CALLS  #1, DBG$REL_MEMORY                      ;
            53                54   D0 0006F      MOVL   TEMP_PTR, ENTRY_PTR                     ; 0386
                              F1   11 00072      BRB    5$                                      ; 0382
                              52   DD 00074 6$:  PUSHL  HDRPTR                                  ; 0388
            66                01   FB 00076      CALLS  #1, DBG$REL_MEMORY                      ;
     0000V  CF                00   FB 00079      CALLS  #0, DBG$SRC_INIT                        ; 0395
                                   04 0007E      RET                                            ; 0399

; Routine Size:  127 bytes,    Routine Base:  DBG$CODE + 0000
```

```
1 ROUTINE DBG$SRC_CLOSE_FILE(SFCBPTR): NOVALUE =
1 !
1 ! FUNCTION
1 !	This routine closes an open source file.  It accepts a Source File Con-
1 !	trol Block (SFCB) pointer as input, determines whether the correspond-
1 !	ing file is an RMS file or a module in a source library, and closes the
1 !	file.  It returns immediately if the SFCB is not in use.
1 !
1 ! INPUTS
1 !	SFCBPTR - A pointer to the Source File Control Block for the source file
1 !		  to be closed.
1 !
1 ! OUTPUTS
1 !	NONE
1 !
1
2    BEGIN
2
2    MAP
2	SFCBPTR: REF SFCB$BLOCK;		! Pointer to Source File Control Block
2
2    LOCAL
2	STATUS;					! Close status
2
2
2
2    ! Close the file specified by the Source File Control Block pointer
2    ! SFCBPTR.  Determine the kind of the SFCBPTR block and act accordingly.
2    !
2    CASE .SFCBPTR[SFCB$B_KIND] FROM SFCB$K_NOTUSED TO SFCB$K_FILE_UNAVAIL OF
2	SET
2
2
2	! This Source File Control Block is not used or is occupied by an
2	! available file.  Nothing needs to be done.
2	!
2	[SFCB$K_NOTUSED, SFCB$K_FILE_UNAVAIL]:
2	    0;
2
2
2	! This Source File Control Block is for an open RMS source file.  Close
2	! RMS file.
2	!
2	[SFCB$K_RMSFILE]:
3	    BEGIN
3	    STATUS = $CLOSE(FAB = .SFCBPTR[SFCB$L_FABPTR]);
3	    IF NOT .STATUS THEN SIGNAL (.STATUS);
2	    END;
2
2
2	! This Source File Control Block is for a module in an open source
2	! library.  Close the source library.
2	!
2	[SFCB$K_LBRFILE]:
3	    BEGIN
3	    STATUS = LBR$CLOSE(SFCBPTR[SFCB$L_LBRINDEX]);
3	    IF NOT .STATUS THEN SIGNAL(.STATUS);
```

```
;  327      0457  2              END;
;  328      0458  2
;  329      0459  2          [INRANGE, OUTRANGE]:
;  330      0460  2              $DBG_ERROR('DBGSOURCE\DBG$SRC_CLOSE_FILE');
;  331      0461  2
;  332      0462  2          TES;
;  333      0463  2
;  334      0464  2
;  335      0465  2      ! The file is successfully closed.  Mark the Source File Control Block as
;  336      0466  2      ! being 'NOT USED' and return.
;  337      0467  2      !
;  338      0468  2      SFCBPTR[SFCB$B_KIND] = SFCB$K_NOTUSED;
;  339      0469  2      RETURN;
;  340      0470  2
;  341      0471  1      END;


                                                                  .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

24  47  42  44  5C  45  43  52  55  4F  53  47  42  44  1C  00000 P.AAA:  .ASCII  <28>\DBGSOURCE\<92>\DBG$SRC_CLOSE_FILE\     ;
    45  4C  49  46  5F  45  53  4F  4C  43  5F  43  52  53  0000F                                                          ;

                                                                  .EXTRN  SYS$CLOSE

                                                                  .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                          000C 00000 DBG$SRC_CLOSE_FILE:
                                                                  .WORD   Save R2,R3                                  ; 0400
                          53 00000000G  00  9E 00002              MOVAB   LIB$SIGNAL, R3
                          52        04  AC  D0 00009              MOVL    SFCBPTR, R2                                 ; 0429
              03          01        08  A2  8F 0000D              CASEB   8(R2), #1, #3                               ;
     0039     0027      001B          0039     00012 1$:          .WORD   5$-1$,-                                     ;
                                                                          2$-1$,-                                     ;
                                                                          3$-1$,-                                     ;
                                                                          5$-1$                                       ;
                        00000000'   EF  9F 0001A                  PUSHAB  P.AAA                                       ; 0460
                                    01  DD 00020                  PUSHL   #1                                          ;
                        00028362    8F  DD 00022                  PUSHL   #164706                                     ;
                          63        03  FB 00028                  CALLS   #3, LIB$SIGNAL                              ;
                                    1E  11 0002B                  BRB     5$                                          ;
                          18        A2  DD 0002D 2$:              PUSHL   24(R2)                                      ; 0445
             00000000G  00          01  FB 00030                  CALLS   #1, SYS$CLOSE                               ;
                                    0A  11 00037                  BRB     4$                                          ; 0446
                          3C        A2  9F 00039 3$:              PUSHAB  60(R2)                                      ; 0455
             00000000G  00          01  FB 0003C                  CALLS   #1, LBR$CLOSE                               ;
                          05        50  E8 00043 4$:              BLBS    STATUS, 5$                                  ; 0456
                                    50  DD 00046                  PUSHL   STATUS                                      ;
                          63        01  FB 00048                  CALLS   #1, LIB$SIGNAL                              ;
                    08    A2        01  90 0004B 5$:              MOVB    #1, 8(R2)                                   ; 0468
                                        04 0004F                  RET                                                 ; 0471

; Routine Size:  80 bytes,    Routine Base:  DBG$CODE + 007F
```

```
1 GLOBAL ROUTINE DBG$SRC_INIT: NOVALUE =
1 !
1 ! FUNCTION
1 !       This routine initializes the Source File Control Blocks and associated
1 !       FABs and RABs needed by the Source Line Display feature to open and read
1 !       source files.  It also fixes the maximum number of source files DEBUG
1 !       will keep open at any one time.  It is called in two situations.  First,
1 !       it is called during DEBUG initialization to set up the initial chain of
1 !       Source File Control Blocks.  And second, it is called during the proc-
1 !       essing of the SET SOURCE and CANCEL SOURCE commands to reset the Source
1 !       Line Display feature; those commands can change the source file look-up
1 !       rules and thus affect whether the currently open files are the correct
1 !       files to use for source line display.
1 !
1 !       The routine starts by looping through all existing Source File Control
1 !       Blocks (SFCBs) and closing any associated open files.  It calls routine
1 !       DBG$SRC_CLOSE_FILE for each file to be closed.  It then releases all
1 !       SFCBs and associated FABs and RABs on the chain.  After that, it creates
1 !       a new list of SFCBs, again with associated FABs and RABs, and marks each
1 !       SFCB as being not used.  The number of SFCBs it creates, which is also
1 !       the maximum number of open source files henceforth, is specified by the
1 !       variable DBG$SRC_MAX_FILES.  This maximum number can thus be changed by
1 !       setting DBG$SRC_MAX_FILES appropriately and then calling DBG$SRC_INIT.
1 !
1 ! INPUTS
1 !       NONE
1 !
1 ! OUTPUTS
1 !       NONE
1 !
1
2     BEGIN
2
2     LOCAL
2         SFCB_BLINK: REF SFCB$BLOCK,        ! Backward link to previous SFCB
2         SFCB_FLINK: REF SFCB$BLOCK,        ! Forward link to next SFCB
2         SFCBPTR: REF SFCB$BLOCK;           ! Pointer to the current Source File
2                                            !         Control Block (SFCB)
2
2
2
2     ! Make sure the number of new Source File Control Blocks is reasonable.
2     !
3     IF (.DBG$SRC_MAX_FILES LSS 1) OR (.DBG$SRC_MAX_FILES GTR MAX_MAX_FILES)
2     THEN
2         $DBG_ERROR('DBGSOURCE\DBG$SRC_INIT');
2
2
2     ! Go through all current Source File Control Blocks and for each one, close
2     ! the corresponding file, release the FAB, RAB, and other RMS blocks back to
2     ! the memory pool, and then release the Source File Control Block itself.
2     !
2     SFCBPTR = .DBG$SRC_SFCB_PTR;
2     SFCB_FLINK = 0;
2     WHILE .SFCB_FLINK NEQ .DBG$SRC_SFCB_PTR DO
3         BEGIN
3         DBG$SRC_CLOSE_FILE(.SFCBPTR);
```

```
3               DBG$REL_MEMORY(.SFCBPTR[SFCB$L_FABPTR]);
3               DBG$REL_MEMORY(.SFCBPTR[SFCB$L_RABPTR]);
3               DBG$REL_MEMORY(.SFCBPTR[SFCB$L_NAMPTR]);
3               DBG$REL_MEMORY(.SFCBPTR[SFCB$L_XABDATPTR]);
3               DBG$REL_MEMORY(.SFCBPTR[SFCB$L_XABFHCPTR]);
3               DBG$REL_MEMORY(.SFCBPTR[SFCB$L_NAMBUFFER]);
3               DBG$REL_MEMORY(.SFCBPTR[SFCB$L_RFATBLPTR]);
3               SFCB_FLINK = .SFCBPTR[SFCB$L_FLINK];
3               DBG$REL_MEMORY(.SFCBPTR);
3               SFCBPTR = .SFCB_FLINK;
2               END;
2
2
2           ! Now loop DBG$SRC_MAX_FILES times to create that number of brand new Source
2           ! File Control Blocks and the associated RMS blocks.
2           !
2           DBG$SRC_SFCB_PTR = 0;
2           INCR I FROM 1 TO .DBG$SRC_MAX_FILES DO
3               BEGIN
3
3
3               ! Allocate memory for the Source File Control Block and mark it as
3               ! being Not Used.
3               !
3               SFCBPTR = DBG$GET_MEMORY(SFCB$K_SIZE);
3               SFCBPTR[SFCB$B_KIND] = SFCB$K_NOTUSED;
3
3
3               ! Now allocate memory blocks for all the RMS blocks we will need for
3               ! this source file.
3               !
3               SFCBPTR[SFCB$L_FABPTR] = DBG$GET_MEMORY(FAB$C_BLN/4);
3               SFCBPTR[SFCB$L_RABPTR] = DBG$GET_MEMORY(RAB$C_BLN/4);
3               SFCBPTR[SFCB$L_NAMPTR] = DBG$GET_MEMORY(NAM$C_BLN/4);
3               SFCBPTR[SFCB$L_XABDATPTR] = DBG$GET_MEMORY(XAB$C_DATLEN/4);
3               SFCBPTR[SFCB$L_XABFHCPTR] = DBG$GET_MEMORY(XAB$C_FHCLEN/4);
3               SFCBPTR[SFCB$L_NAMBUFFER] = DBG$GET_MEMORY(2*NAMBUFSIZE/4);
3               SFCBPTR[SFCB$L_RFATBLPTR] = DBG$GET_MEMORY(RFATBL_SIZE/4);
3               SFCBPTR[SFCB$W_RFASPACING] = INIT_RFATBL_SPACING;
3               SFCBPTR[SFCB$W_RFACURLEN] = 0;
3
3
3               ! If this is the first new block, make DBG$SRC_SFCB_PTR point to it and
3               ! doubly link it to itself.
3               !
3               IF .DBG$SRC_SFCB_PTR EQL 0
3               THEN
4                   BEGIN
4                   SFCBPTR[SFCB$L_FLINK] = .SFCBPTR;
4                   SFCBPTR[SFCB$L_BLINK] = .SFCBPTR;
4                   DBG$SRC_SFCB_PTR = .SFCBPTR;
4                   END
4
4
4               ! Otherwise, link this block into the end of the already existing doubly
4               ! linked list of Source File Control Blocks.
4               !
```

```
;  457        0586  3         ELSE
;  458        0587  4             BEGIN
;  459        0588  4             SFCB_FLINK = .DBG$SRC_SFCB_PTR;
;  460        0589  4             SFCB_BLINK = .SFCB_FLINK[SFCB$L_BLINK];
;  461        0590  4             SFCBPTR[SFCB$L_FLINK] = .SFCB_FLINK;
;  462        0591  4             SFCBPTR[SFCB$L_BLINK] = .SFCB_BLINK;
;  463        0592  4             SFCB_FLINK[SFCB$L_BLINK] = .SFCBPTR;
;  464        0593  4             SFCB_BLINK[SFCB$L_FLINK] = .SFCBPTR;
;  465        0594  3             END;
;  466        0595  3
;  467        0596  2         END;
;  468        0597  2
;  469        0598  2
;  470        0599  2     ! The Source File Control Block initialization is all done--now return.
;  471        0600  2     !
;  472        0601  2     RETURN;
;  473        0602  2
;  474        0603  1     END;


                                                                  .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

24 47 42 44 5C 45 43 52 55 4F 53 47 42 44 16  0001D P.AAB:   .ASCII  <22>\DBGSOURCE\<92>\DBG$SRC_INIT\              ;
                     54 49 4E 49 5F 43 52 53  0002C                                                                 ;


                                                                  .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                       07FC 00000          .ENTRY  DBG$SRC_INIT, Save R2,R3,R4,R5,R6.R7,R8,R9,-; 0472
                                                                   R10                                              ;
               5A 00000000'   EF   9E 00002          MOVAB   DBG$SRC_MAX_FILES, R10                              ;
               59 00000000'   EF   9E 00009          MOVAB   DBG$SRC_SFCB_PTR, R9                                ;
               58 00000000G   00   9E 00010          MOVAB   DBG$GET_MEMORY, R8                                  ;
               57 00000000G   00   9E 00017          MOVAB   DBG$REL_MEMORY, R7                                  ;
                              6A   D5 0001E          TSTL    DBG$SRC_MAX_FILES                                   ; 0515
                              05   15 00020          BLEQ    1$                                                  ;
               14             6A   D1 00022          CMPL    DBG$SRC_MAX_FILES, #20                              ;
                              15   15 00025          BLEQ    2$                                                  ;
                  00000000'   EF   9F 00027 1$:      PUSHAB  P.AAB                                               ; 0517
                              01   DD 0002D          PUSHL   #1                                                  ;
                  00028362    8F   DD 0002F          PUSHL   #164706                                             ;
     00000000G 00             03   FB 00035          CALLS   #3, LIB$SIGNAL                                      ;
               52             69   D0 0003C 2$:      MOVL    DBG$SRC_SFCB_PTR, SFCBPTR                           ; 0524
                              53   D4 0003F          CLRL    SFCB_FLINK                                          ; 0525
               69             53   D1 00041 3$:      CMPL    SFCB_FLINK, DBG$SRC_SFCB_PTR                        ; 0526
                              3E   13 00044          BEQL    4$                                                  ;
                              52   DD 00046          PUSHL   SFCBPTR                                             ; 0528
          FF63 CF             01   FB 00048          CALLS   #1, DBG$SRC_CLOSE_FILE                              ;
                        18    A2   DD 0004D          PUSHL   24(SFCBPTR)                                         ; 0529
               67             01   FB 00050          CALLS   #1, DBG$REL_MEMORY                                  ;
                        1C    A2   DD 00053          PUSHL   28(SFCBPTR)                                         ; 0530
               67             01   FB 00056          CALLS   #1, DBG$REL_MEMORY                                  ;
                        20    A2   DD 00059          PUSHL   32(SFCBPTR)                                         ; 0531
               67             01   FB 0005C          CALLS   #1, DBG$REL_MEMORY                                  ;
                        24    A2   DD 0005F          PUSHL   36(SFCBPTR)                                         ; 0532
```

```
                  67          01  FB 00062          CALLS   #1, DBG$REL_MEMORY                  ;
                        28    A2  DD 00065          PUSHL   40(SFCBPTR)                         ; 0533
                  67          01  FB 00068          CALLS   #1, DBG$REL_MEMORY                  ;
                        2C    A2  DD 0006B          PUSHL   44(SFCBPTR)                         ; 0534
                  67          01  FB 0006E          CALLS   #1, DBG$REL_MEMORY                  ;
                        10    A2  DD 00071          PUSHL   16(SFCBPTR)                         ; 0535
                  67          01  FB 00074          CALLS   #1, DBG$REL_MEMORY                  ;
                  53          62  D0 00077          MOVL    (SFCBPTR), SFCB_FLINK               ; 0536
                              52  DD 0007A          PUSHL   SFCBPTR                             ; 0537
                  67          01  FB 0007C          CALLS   #1, DBG$REL_MEMORY                  ;
                  52          53  D0 0007F          MOVL    SFCB_FLINK, SFCBPTR                 ; 0538
                              BD  11 00082          BRB     3$                                  ; 0526
                              69  D4 00084 4$:      CLRL    DBG$SRC_SFCB_PTR                    ; 0545
                  56          6A  D0 00086          MOVL    DBG$SRC_MAX_FILES, R6               ; 0546
                              54  D4 00089          CLRL    I                                   ;
                              79  11 0008B          BRB     7$                                  ;
                              10  DD 0008D 5$:      PUSHL   #16                                 ; 0553
                  68          01  FB 0008F          CALLS   #1, DBG$GET_MEMORY                  ;
                  52          50  D0 00092          MOVL    R0, SFCBPTR                         ;
            08    A2          01  90 00095          MOVB    #1, 8(SFCBPTR)                      ; 0554
                              14  DD 00099          PUSHL   #20                                 ; 0560
                  68          01  FB 0009B          CALLS   #1, DBG$GET_MEMORY                  ;
            18    A2          50  D0 0009E          MOVL    R0, 24(SFCBPTR)                     ;
                              11  DD 000A2          PUSHL   #17                                 ; 0561
                  68          01  FB 000A4          CALLS   #1, DBG$GET_MEMORY                  ;
            1C    A2          50  D0 000A7          MOVL    R0, 28(SFCBPTR)                     ;
                              18  DD 000AB          PUSHL   #24                                 ; 0562
                  68          01  FB 000AD          CALLS   #1, DBG$GET_MEMORY                  ;
            20    A2          50  D0 000B0          MOVL    R0, 32(SFCBPTR)                     ;
                              0B  DD 000B4          PUSHL   #11                                 ; 0563
                  68          01  FB 000B6          CALLS   #1, DBG$GET_MEMORY                  ;
            24    A2          50  D0 000B9          MOVL    R0, 36(SFCBPTR)                     ;
                              0B  DD 000BD          PUSHL   #11                                 ; 0564
                  68          01  FB 000BF          CALLS   #1, DBG$GET_MEMORY                  ;
            28    A2          50  D0 000C2          MOVL    R0, 40(SFCBPTR)                     ;
                  7E    80    8F  9A 000C6          MOVZBL  #128, -(SP)                         ; 0565
                  68          01  FB 000CA          CALLS   #1, DBG$GET_MEMORY                  ;
            2C    A2          50  D0 000CD          MOVL    R0, 44(SFCBPTR)                     ;
                  7E    4B    8F  9A 000D1          MOVZBL  #75, -(SP)                          ; 0566
                  68          01  FB 000D5          CALLS   #1, DBG$GET_MEMORY                  ;
            10    A2          50  D0 000D8          MOVL    R0, 16(SFCBPTR)                     ;
            0C    A2          05  D0 000DC          MOVL    #5, 12(SFCBPTR)                     ; 0567
                  50          69  D0 000E0          MOVL    DBG$SRC_SFCB_PTR, R0                ; 0574
                              0C  12 000E3          BNEQ    6$                                  ;
                  62          52  D0 000E5          MOVL    SFCBPTR, (SFCBPTR)                  ; 0577
            04    A2          52  D0 000E8          MOVL    SFCBPTR, 4(SFCBPTR)                 ; 0578
                  69          52  D0 000EC          MOVL    SFCBPTR, DBG$SRC_SFCB_PTR           ; 0579
                              15  11 000EF          BRB     7$                                  ; 0574
                  53          50  D0 000F1 6$:      MOVL    R0, SFCB_FLINK                      ; 0588
                  55    04    A3  D0 000F4          MOVL    4(SFCB_FLINK), SFCB_BLINK           ; 0589
                  62          53  D0 000F8          MOVL    SFCB_FLINK, (SFCBPTR)               ; 0590
            04    A2          55  D0 000FB          MOVL    SFCB_BLINK, 4(SFCBPTR)              ; 0591
            04    A3          52  D0 000FF          MOVL    SFCBPTR, 4(SFCB_FLINK)              ; 0592
                  65          52  D0 00103          MOVL    SFCBPTR, (SFCB_BLINK)               ; 0593
      83          54          56  F3 00106 7$:      AOBLEQ  R6, I, 5$                           ; 0546
                                  04 0010A          RET                                         ; 0603
```

; Routine Size: 267 bytes,   Routine Base: DBG$CODE + 00CF

```
GLOBAL ROUTINE DBG$SRC_LINE_TO_REC(MODRSTPTR, LINENO, NAMBLKPTR, RECNO) : NOVALUE =
!
! FUNCTION
!       This routine accepts a Module RST pointer and a line number.
!       It translates this information to a file specification and record
!       number within the file. It does this by scanning the source
!       correlation records.
!
!       This routine is used by the DBG$NEXECUTE_EDIT routine to implement
!       our EDITH interface.
!
! INPUTS
!       MODRSTPTR       - Module RST pointer
!       LINENO          - The desired line number
!
! OUTPUTS
!       NAMBLKPTR       - Pointer to the name block of the current source file
!       RECNO           - This output parameter is filled in to
!                               be the source record within the file.
!
    BEGIN
    MAP
        NAMBLKPTR : REF VECTOR[1,LONG], ! Returned pointer to current SFCB
        MODRSTPTR: REF RST$ENTRY;       ! Pointer to Module RST Entry of module
                                        !       containing desired source lines

    LOCAL
        BUFFER: VECTOR[140,BYTE],       ! Output buffer for signaling
        BUF_DESC: VECTOR[2,LONG],       ! Output buffer string descriptor
        DSTPTR: REF DST$SRC_COMMAND,    ! Pointer to the Source File Correlation
                                        !       Command
        DSTREC_PTR: REF DST$RECORD,     ! Pointer to the Source File Correlation
                                        !       DST record
        HIGH_LINE_NUM,                  ! Place holder for Ending Line Number
        LENGTH,                         ! Length of current Source File Correlation
                                        !       DST Record
        LINE_NUM,                       ! The current listing line number
        MODNAMEPTR,
        MODSRCTBL: REF VECTOR[,LONG],   ! Pointer to a list of Source File
                                        !       Correlation DST Record pointers
                                        !       for the current module
        SFIT_PTR: REF SFIT$ENTRY,       ! Pointer to the Source File ID Table
        SFCBPTR: REF SFCB$BLOCK,
        SRC_FILEID_TBL,                 ! Pointer to a linked list of File ID
                                        !       Table entries for current module
        SRC_DSTPTR,                     ! Pointer to Declare Source File Command
                                        !       in DST Record
        SRC_REC;                        ! The record number in the current source
                                        !       file of the current source line



    ! Initialization.  Assume that Line Number is set to the beginning of the
    ! listing, Source File Record Number is set to the beginning of the
    ! source file, no Source File ID is found, Statement Mode is not set,
    ! no line is yet typed out, and statement number is set to 0.
    !
    DBG$SRC_FORMFEED_FLAG = FALSE;
```

```
LINE_NUM = 1;
SRC_REC = 1;
SFIT_PTR = 0;
SRC_DSTPTR = 0;
SRC_FILEID_TBL = 0;


! Make sure Module Source Table pointer in module's Module RST Entry
! points to a list of Source File DST Record Pointers.
!
DBG$STA_SYMNAME(.MODRSTPTR, MODNAMEPTR);
IF .MODRSTPTR[RST$L_MODSRCTBL] EQL 0
THEN
    SIGNAL(DBG$_SRCLINNOT,1,.MODNAMEPTR);


! Loop thru Module's Source Line Correlation DST Records.
!
MODSRCTBL = .MODRSTPTR[RST$L_MODSRCTBL];
INCR COUNT FROM 1 TO .MODSRCTBL[0] DO
    BEGIN
    IF .LINE_NUM GTR .LINENO THEN EXITLOOP;


    ! Get Source Line Correlation DST Record pointer.
    !
    DSTREC_PTR = .MODSRCTBL[.COUNT];
    IF .DSTREC_PTR[DST$B_TYPE] NEQ DST$K_SOURCE
    THEN
        $DBG_ERROR('DBGSOURCE\DBG$SRC_LINE_TO_REC 20');


    ! Scans thru Source File DST Record to decode the Source File Correlation
    ! Commands.  Pick up the length of this DST record.  Set pointer to the
    ! first command entry.
    !
    LENGTH = .DSTREC_PTR[DST$B_LENGTH];
    DSTPTR = DSTREC_PTR[DST$A_SRC_FIRST_CMD];
    WHILE (.DSTPTR - .DSTREC_PTR - 1) NEQ .LENGTH DO
        BEGIN
        IF .LINE_NUM GTR .LINENO THEN EXITLOOP;


        ! Pick up the command code defined in Source File DST Record
        ! and use it as a CASE index so each significant command code
        ! can be interpreted individually.
        !
        CASE .DSTPTR[DST$B_SRC_COMMAND] FROM DST$K_SRC_MIN_CMD TO
                                              DST$K_SRC_MAX_CMD OF
            SET


            ! Declare Source File Command.  Allocate some space to build
            ! the File ID Table.
            !
            [DST$K_SRC_DECLFILE]:
                BEGIN
```

```
5              IF .SFIT_PTR NEQ 0 THEN SFIT_PTR[SFIT$L_CURRECNUM] = .SRC_REC;
5              SRC_REC = 1;
5              SFIT_PTR = DBG$GET_TEMPMEM(SFIT$K_SIZE);
5              SFIT_PTR[SFIT$L_FLINK] = .SRC_FILEID_TBL;
5              SFIT_PTR[SFIT$L_FILE_ID] = .DSTPTR[DST$W_SRC_DF_FILEID];
5              SFIT_PTR[SFIT$L_DSTPTR] = .DSTPTR;
5              SFIT_PTR[SFIT$L_CURRECNUM] = .SRC_REC;
5              SRC_FILEID_TBL = .SFIT_PTR;
5              DSTPTR = .DSTPTR + .DSTPTR[DST$B_SRC_DF_LENGTH] + 2;
4              END;
4
4
4          ! Set Source File Command.  This command sets the current
4          ! source file to the file denoted by the File ID given in
4          ! the command.  This File ID must be pre-defined by the
4          ! Declare Source File Command.
4          !
4          [DST$K_SRC_SETFILE]:
5              BEGIN
5              IF .SFIT_PTR NEQ 0 THEN SFIT_PTR[SFIT$L_CURRECNUM] = .SRC_REC;
5              SRC_DSTPTR = 0;
5              SFIT_PTR = .SRC_FILEID_TBL;
5              WHILE .SFIT_PTR NEQ 0 DO
6                  BEGIN
6                  IF .DSTPTR[DST$W_SRC_UNSWORD] EQL .SFIT_PTR[SFIT$L_FILE_ID]
6                  THEN
7                      BEGIN
7                      SRC_DSTPTR = .SFIT_PTR[SFIT$L_DSTPTR];
7                      SRC_REC = .SFIT_PTR[SFIT$L_CURRECNUM];
7                      EXITLOOP;
6                      END;
6
6                  SFIT_PTR = .SFIT_PTR[SFIT$L_FLINK];
5                  END;
5
5
5              ! If this File ID has not yet been declared, signal an
5              ! invalid DST Record.
5              !
5              IF .SRC_DSTPTR EQL 0 THEN SIGNAL(DBG$_INVDSTREC);
5              DSTPTR = .DSTPTR + 3;
4              END;
4
4
4          ! Set Source Record Number Long Command.  Set the current source
4          ! file record number.  Advance DSTPTR.
4          !
4          [DST$K_SRC_SETREC_L]:
5              BEGIN
5              SRC_REC = .DSTPTR[DST$L_SRC_UNSLONG];
5              DSTPTR = .DSTPTR + 5;
4              END;
4
4
4          ! Set Source Record Number Word Command. (More compact form).
4          !
4          [DST$K_SRC_SETREC_W]:
```

```
                BEGIN
                SRC_REC = .DSTPTR[DST$W_SRC_UNSWORD];
                DSTPTR = .DSTPTR + 3;
                END;


            ! Set Line Number Long Command.  Set the current listing line
            ! number.  Advance DSTPTR.
            !
            [DST$K_SRC_SETLNUM_L]:
                BEGIN
                LINE_NUM = .DSTPTR[DST$L_SRC_UNSLONG];
                DSTPTR = .DSTPTR + 5;
                END;


            ! Set Line Number Word Command. (More compact form).
            !
            [DST$K_SRC_SETLNUM_W]:
                BEGIN
                LINE_NUM = .DSTPTR[DST$W_SRC_UNSWORD];
                DSTPTR = .DSTPTR + 3;
                END;


            ! Increment Line Number Byte Command.  Increment the current
            ! listing line number by a byte value given in this command.
            ! Advance DSTPTR.
            !
            [DST$K_SRC_INCRLNUM_B]:
                BEGIN
                LINE_NUM = .LINE_NUM + .DSTPTR[DST$B_SRC_UNSBYTE];
                DSTPTR = .DSTPTR + 2;
                END;


            ! Count Form-Feeds as Source Records Command.  This command says
            ! that form-feeds in the source should be counted as separate
            ! source records; the absence of this command says that such
            ! records should be ignored as control information and should
            ! not be assigned line nubmers.
            !
            [DST$K_SRC_FORMFEED]:
                BEGIN
                DBG$SRC_FORMFEED_FLAG = TRUE;
                DSTPTR = .DSTPTR + 1;
                END;


            ! Define N Separate Lines Word Command, and Define N Separate
            ! Lines Byte Command (More Compact Form).  If a Source File is not
            ! set, signal and give up.  Defines the Source File and Source
            ! Record Numbers for a specified number of Listing Line Numbers.
            ! Return if LINENO = LINE_NUM
            !
            [DST$K_SRC_DEFLINES_W, DST$K_SRC_DEFLINES_B]:
                BEGIN
```

```
IF .SRC_DSTPTR EQL 0 THEN SIGNAL(DBG$_INVDSTREC);

! Calculate the listing line number range.
!
IF .DSTPTR[DST$B_SRC_COMMAND] EQL DST$K_SRC_DEFLINES_W
THEN
    HIGH_LINE_NUM = .LINE_NUM + .DSTPTR[DST$W_SRC_UNSWORD] - 1
ELSE
    HIGH_LINE_NUM = .LINE_NUM + .DSTPTR[DST$B_SRC_UNSBYTE] - 1;


! Test to see if the listing line number exceeds the type
! line range.  If it is, increments the source record number
! listing line number and DST pointer, then returns.
!
IF (.LINE_NUM LEQ .LINENO) AND
   (.LINENO LEQ .HIGH_LINE_NUM)
THEN
    BEGIN
    MAP
        SRC_DSTPTR: REF DST$SRC_COMMAND;
    LOCAL
        FORMFEED_COUNT,
        LENGTH,
        RECORD_NUMBER;


    ! We have a range covering LINENO.
    ! If the language does not count formfeeds as
    ! source, then the source correlation tables
    ! are counting formfeeds differently from
    ! the editor. We need to count the number
    ! of formfeeds so we can correct for this.
    !
    RECORD_NUMBER = .SRC_REC + (.LINENO - .LINE_NUM);
    FORMFEED_COUNT = 0;
    IF NOT .DBG$SRC_FORMFEED_FLAG
    THEN
        BEGIN
        LOCAL
            INPTR: BLOCK[8,BYTE],
            OUTPTR: BLOCK[8,BYTE],
            RFA_TABLE: REF RFATBL$BLOCKVECTOR(RFATBL_ENTRIES),
            SRCNAM: REF $NAM_DECL,
            SRCRAB: REF $RAB_DECL,
            STATUS,
            TXTRFA: VECTOR[2];


        ! Open the file.
        !
        DBG$SRC_OPEN_FILE(.SRC_DSTPTR,
                          SFCBPTR,
                          FALSE);   !<< TRUE ?

```

```
! Determine whether to read from an RMS file or a module in a source library.

SRCNAM = .SFCBPTR[SFCB$L_NAMPTR];
CASE .SFCBPTR[SFCB$B_KIND] FROM SFCB$K_RMSFILE TO SFCB$K_LBRFILE OF
    SET


        ! Read from an RMS source file.
        !
        [SFCB$K_RMSFILE]:
            BEGIN


            ! Set up the RAB pointer.
            !
            SRCRAB = .SFCBPTR[SFCB$L_RABPTR];
            RFA_TABLE = .SFCBPTR[SFCB$L_RFATBLPTR];
            IF .SFCBPTR[SFCB$L_CURRECNUM] EQL 1
            THEN
                BEGIN
                STATUS = $FIND(RAB = .SRCRAB);
                IF NOT .STATUS THEN SIGNAL(.STATUS);
                RFA_TABLE[0,RFATBL$L_RFA0] =
                    .SRCRAB[RAB$L_RFA0];
                RFA_TABLE[0,RFATBL$W_RFA4] =
                    .SRCRAB[RAB$W_RFA4];
                END;


            ! Sequentially read from the first record to the desired record
            ! number.
            !
            SFCBPTR[SFCB$L_CURRECNUM] = 1;
            STATUS = $REWIND(RAB = .SRCRAB);
            IF NOT .STATUS THEN SIGNAL(.STATUS);
            SRCRAB[RAB$B_RAC] = RAB$C_SEQ;
            INCR I FROM .SFCBPTR[SFCB$L_CURRECNUM] TO .RECORD_NUMBER DO
                BEGIN
                WHILE TRUE DO
                    BEGIN
                    STATUS = $GET(RAB = .SRCRAB);
                    IF NOT .STATUS
                    THEN
                        SIGNAL(DBG$_UNAREASRC,2,.SRCNAM[NAM$B_RSL],.SRCNAM[NAM$L_RSA


                    ! Count records with formfeed in
                    ! formfeed_count. Do not count them
                    ! toward RECORD_COUNT.
                    !
                    IF (.DBG$SRC_REC_BUF[0] EQL 12) AND
                       (.SRCRAB[RAB$W_RSZ] EQL 1)
                    THEN
                        FORMFEED_COUNT = .FORMFEED_COUNT + 1
                    ELSE
                        EXITLOOP;
                    END;
```

```
                !
                ! Update the current record number field in SFCB. Set up the buffer
                ! pointer and buffer length.
                !
                SFCBPTR[SFCB$L_CURRECNUM] = .RECORD_NUMBER + 1;
                END;
            END; ! RMS file case alternative

        !
        ! Read from a module in a Source Library File.
        !
        [SFCB$K_LBRFILE]:
            BEGIN

            !
            ! If the current record number is beyond the desired record number,
            ! reset the current record to 1 and set the record file address to the
            ! beginning of the module.
            !
            IF .SFCBPTR[SFCB$L_CURRECNUM] GTR .RECORD_NUMBER
            THEN
                BEGIN
                TXTRFA[0] = .SFCBPTR[SFCB$L_CUR_RFA0];
                TXTRFA[1] = .SFCBPTR[SFCB$W_CUR_RFA4];
                SFCBPTR[SFCB$L_CURRECNUM] = 1;
                STATUS = LBR$FIND(SFCBPTR[SFCB$L_LBRINDEX],TXTRFA);
                IF NOT .STATUS THEN SIGNAL(.STATUS);
                END;

            !
            ! Sequentially read from the current record to the desired record
            ! number.
            !
            INPTR[DSC$W_LENGTH] = SRCBUFSIZE;
            INPTR[DSC$A_POINTER] = DBG$SRC_REC_BUF[0];
           OUTPTR[DSC$A_POINTER] = DBG$SRC_REC_BUF[0];
            INCR I FROM .SFCBPTR[SFCB$L_CURRECNUM] TO .RECORD_NUMBER DO
                BEGIN
                WHILE TRUE DO
                    BEGIN
                    STATUS = LBR$GET_RECORD(SFCBPTR[SFCB$L_LBRINDEX],INPTR,OUTPTR);
                    IF NOT .STATUS
                    THEN
                        SIGNAL(DBG$_UNAREASRC,2,.SRCNAM[NAM$B_RSL],.SRCNAM[NAM$L_RSA

                    IF (.DBG$SRC_REC_BUF[0] EQL 12) AND
                        (.OUTPTR[DSC$W_LENGTH] EQL 1)
                    THEN
                        FORMFEED_COUNT = .FORMFEED_COUNT + 1
                    ELSE
                        EXITLOOP;
                    END;
                END;

            !
            ! Update the current record number field in SFCB.  Set up buffer
            ! pointer and buffer length.
```

```
8                                 !
8                                 SFCBPTR[SFCB$L_CURRECNUM] = .RECORD_NUMBER + 1;
7                                 END;
7
7                             [INRANGE, OUTRANGE]:
7                                 $DBG_ERROR('DBGSOURCE\DBG$SRC_READ_FILE');
7
7                             TES;
7
7                         RECORD_NUMBER = .RECORD_NUMBER + .FORMFEED_COUNT;
6                         END;
6
6                     ! Return the record number and the NAM block pointer
6                     !
6                     NAMBLKPTR[0] = .SFCBPTR[SFCB$L_NAMPTR];
6                     .RECNO = .RECORD_NUMBER;
6                     RETURN;
6                     END
6
5                 ELSE
6                     BEGIN
6                     IF .DSTPTR[DST$B_SRC_COMMAND] EQL DST$K_SRC_DEFLINES_W
6                     THEN
6                         SRC_REC = .SRC_REC + .DSTPTR[DST$W_SRC_UNSWORD]
6                     ELSE
6                         SRC_REC = .SRC_REC + .DSTPTR[DST$B_SRC_UNSBYTE];
5                     END;
5
5                 LINE_NUM = .HIGH_LINE_NUM + 1;
5                 IF .DSTPTR[DST$B_SRC_COMMAND] EQL DST$K_SRC_DEFLINES_W
5                 THEN
5                     DSTPTR = .DSTPTR + 3
5                 ELSE
5                     DSTPTR = .DSTPTR + 2;
4                 END;
4
4
4             ! Any other command code constitutes an error in the DST.
4             !
4             [INRANGE, OUTRANGE]:
4                 SIGNAL(DBG$_INVDSTREC);
4
4             TES;
3         END;                            ! End of WHILE Loop through commands.
2     END;                                ! End of INCR Loop through DST records.
2
2
2 ! If we fall through to here, we did not find the line.
2 !
2 BUF_DESC[0] = 139;
2 BUF_DESC[1] = BUFFER[1];
2 DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC 'line '), 0);
2 DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC '!UL'), .LINENO);
2 DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC ' does'), 0);
2 DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC ' not exist in module !AC'), .MODNAMEPTR);
2 BUFFER[0] = 139 - .BUF_DESC[0];
2 SIGNAL(DBG$_NOLINXXX, 1, BUFFER);
```

```
; 932          1060  1      END;


                                                                        .PSECT  DBG$PLIT,NOWRT, SHR, PIC,0

24 47 42 44 5C 45 43 52 55 4F 53 47 42 44 20 00034 P.AAC:  .ASCII  \ DBGSOURCE\<92>\DBG$SRC_LINE_TO_REC 20\      ;
43 45 52 5F 4F 54 5F 45 4E 49 4C 5F 43 52 53 00043                                                              ;
                                        30 32 20 00052                                                              ;
24 47 42 44 5C 45 43 52 55 4F 53 47 42 44 1B 00055 P.AAD:  .ASCII  <27>\DBGSOURCE\<92>\DBG$SRC_READ_FILE\       ;
      45 4C 49 46 5F 44 41 45 52 5F 43 52 53 00064                                                              ;
                              20 65 6E 69 6C 05 00071 P.AAE:  .ASCII  <5>\line \                                    ;
                                    4C 55 21 03 00077 P.AAF:  .ASCII  <3>\!UL\                                      ;
                              73 65 6F 64 20 05 0007B P.AAG:  .ASCII  <5>\ does\                                    ;
20 6E 69 20 74 73 69 78 65 20 74 6F 6E 20 18 00081 P.AAH:  .ASCII  <24>\ not exist in module !AC\              ;
            43 41 21 20 65 6C 75 64 6F 6D 00090                                                              ;

                                                           .EXTRN  SYS$FIND, SYS$REWIND
                                                           .EXTRN  SYS$GET

                                                           .PSECT  DBG$CODE,NOWRT, SHR, PIC,0

                                   0FFC 00000             .ENTRY  DBG$SRC_LINE_TO_REC, Save R2,R3,R4,R5,R6,-  ; 0604
                                                                   R7,R8,R9,R10,R11                           ;
                    5E       FF2C   CE 9E 00002           MOVAB   -212(SP), SP                                 ;
                      00000000'     EF D4 00007           CLRL    DBG$SRC_FORMFEED_FLAG                        ; 0660
                    57              01 D0 0000D           MOVL    #1, LINE_NUM                                 ; 0661
                    5B              01 D0 00010           MOVL    #1, SRC_REC                                  ; 0662
                                    54 D4 00013           CLRL    SFIT_PTR                                     ; 0663
                                    7E 7C 00015           CLRQ    SRC_FILEID_TBL                               ; 0665
                              28    AE 9F 00017           PUSHAB  MODNAMEPTR                                   ; 0671
                    52        04    AC D0 0001A           MOVL    MODRSTPTR, R2                                ;
                                    52 DD 0001E           PUSHL   R2                                           ;
             00000000G  00          02 FB 00020           CALLS   #2, DBG$STA_SYMNAME                          ;
                              24    A2 D5 00027           TSTL    36(R2)                                       ; 0672
                                    12 12 0002A           BNEQ    1$                                           ;
                              28    AE DD 0002C           PUSHL   MODNAMEPTR                                   ; 0674
                                    01 DD 0002F           PUSHL   #1                                           ;
                      00028CB8      8F DD 00031           PUSHL   #167096                                      ;
             00000000G  00          03 FB 00037           CALLS   #3, LIB$SIGNAL                               ;
                    1C  AE        24 A2 D0 0003E 1$:      MOVL    36(R2), MODSRCTBL                            ; 0679
                              08    AE D4 00043           CLRL    COUNT                                        ; 0682
                            0322    31 00046 2$:          BRW     54$                                          ;
                    08  AC          57 D1 00049 3$:       CMPL    LINE_NUM, LINENO                             ;
                                    03 15 0004D           BLEQ    4$                                           ;
                            0321    31 0004F              BRW     55$                                          ;
                    50        08    AE D0 00052 4$:       MOVL    COUNT, R0                                    ; 0687
                    10  AE     1C BE40 D0 00056           MOVL    @MODSRCTBL[R0], DSTREC_PTR                   ;
              50    10  AE          01 C1 0005C           ADDL3   #1, DSTREC_PTR, R0                           ; 0688
                    9B  8F          60 91 00061           CMPB    (R0), #155                                   ;
                                    15 13 00065           BEQL    5$                                           ;
                      00000000'     EF 9F 00067           PUSHAB  P.AAC                                        ; 0690
                                    01 DD 0006D           PUSHL   #1                                           ;
                      00028362      8F DD 0006F           PUSHL   #164706                                      ;
             00000000G  00          03 FB 00075           CALLS   #3, LIB$SIGNAL                               ;
                    24  AE    10    BE 9A 0007C 5$:       MOVZBL  @DSTREC_PTR, LENGTH                          ; 0697
              53    10  AE          02 C1 00081           ADDL3   #2, DSTREC_PTR, DSTPTR                       ; 0698
```

```
                 50          53     10    AE  C3 00086 6$:      SUBL3   DSTREC_PTR, DSTPTR, R0                     ; 0699
                                          50  D7 0008B          DECL    R0                                         ;
                       24    AE           50  D1 0008D          CMPL    R0, LENGTH                                 ;
                                          B3  13 00091          BEQL    2$                                         ;
                       08    AC           57  D1 00093          CMPL    LINE_NUM, LINENO                           ; 0701
                                          AD  14 00097          BGTR    2$                                         ;
                 0F          01           63  8F 00099          CASEB   (DSTPTR), #1, #15                          ; 0708
    00A7       00A1        0064         002F     0009D 7$:      .WORD   9$-7$,-                                    ;
    0020       00BD        00B6         00AD     000A5                  12$-7$,-                                   ;
    0020       00D2        00D2         0020     000AD                  17$-7$,-                                   ;
    00C7       0020        0020         0020     000B5                  18$-7$,-                                   ;
                                                                        19$-7$,-                                   ;
                                                                        21$-7$,-                                   ;
                                                                        23$-7$,-                                   ;
                                                                        8$-7$,-                                    ;
                                                                        8$-7$,-                                    ;
                                                                        25$-7$,-                                   ;
                                                                        25$-7$,-                                   ;
                                                                        8$-7$,-                                    ;
                                                                        8$-7$,-                                    ;
                                                                        8$-7$,-                                    ;
                                                                        8$-7$,-                                    ;
                                                                        24$-7$                                     ;
                                  0002832A    8F  DD 000BD 8$:      PUSHL   #164650                                    ; 1043
                      00000000G  00           01  FB 000C3          CALLS   #1, LIB$SIGNAL                             ;
                                              BA  11 000CA          BRB     6$                                         ;
                                              54  D5 000CC 9$:      TSTL    SFIT_PTR                                   ; 0718
                                              04  13 000CE          BEQL    10$                                        ;
                       0C    A4               5B  D0 000D0          MOVL    SRC_REC, 12(SFIT_PTR)                      ;
                             5B               01  D0 000D4 10$:     MOVL    #1, SRC_REC                                ; 0719
                                              04  DD 000D7          PUSHL   #4                                         ; 0720
                      00000000G  00           01  FB 000D9          CALLS   #1, DBG$GET_TEMPMEM                        ;
                             54               50  D0 000E0          MOVL    R0, SFIT_PTR                               ;
                             64               6E  D0 000E3          MOVL    SRC_FILEID_TBL, (SFIT_PTR)                 ; 0721
                       04    A4     03        A3  3C 000E6          MOVZWL  3(DSTPTR), 4(SFIT_PTR)                     ; 0722
                       08    A4               53  D0 000EB          MOVL    DSTPTR, 8(SFIT_PTR)                        ; 0723
                       0C    A4               5B  D0 000EF          MOVL    SRC_REC, 12(SFIT_PTR)                      ; 0724
                             6E               54  D0 000F3          MOVL    SFIT_PTR, SRC_FILEID_TBL                   ; 0725
                             50     01        A3  9A 000F6          MOVZBL  1(DSTPTR), R0                              ; 0726
                             53     02      A043  9E 000FA          MOVAB   2(R0)[DSTPTR], DSTPTR                      ;
                                              85  11 000FF 11$:     BRB     6$                                         ; 0708
                                              54  D5 00101 12$:     TSTL    SFIT_PTR                                   ; 0737
                                              04  13 00103          BEQL    13$                                        ;
                       0C    A4               5B  D0 00105          MOVL    SRC_REC, 12(SFIT_PTR)                      ;
                                    04        AE  D4 00109 13$:     CLRL    SRC_DSTPTR                                 ; 0738
                             54               6E  D0 0010C          MOVL    SRC_FILEID_TBL, SFIT_PTR                   ; 0739
                                              19  13 0010F 14$:     BEQL    16$                                        ; 0740
  04    A4       01    A3    10               00  ED 00111          CMPZV   #0, #16, 1(DSTPTR), 4(SFIT_PTR)            ; 0742
                                              0B  12 00118          BNEQ    15$                                        ;
                       04    AE     08        A4  D0 0011A          MOVL    8(SFIT_PTR), SRC_DSTPTR                    ; 0745
                             5B     0C        A4  D0 0011F          MOVL    12(SFIT_PTR), SRC_REC                      ; 0746
                                              05  11 00123          BRB     16$                                        ; 0744
                             54               64  D0 00125 15$:     MOVL    (SFIT_PTR), SFIT_PTR                       ; 0750
                                              E5  11 00128          BRB     14$                                        ; 0740
                                    04        AE  D5 0012A 16$:     TSTL    SRC_DSTPTR                                 ; 0757
                                              28  12 0012D          BNEQ    22$                                        ;
                                  0002832A    8F  DD 0012F          PUSHL   #164650                                    ;
```

```
         00000000G  00        01  FB 00135        CALLS   #1, LIB$SIGNAL                    :
                              19  11 0013C        BRB     22$                               : 0758
                    5B    01  A3  D0 0013E 17$:   MOVL    1(DSTPTR), SRC_REC                : 0767
                              0A  11 00142        BRB     20$                               : 0768
                    5B    01  A3  3C 00144 18$:   MOVZWL  1(DSTPTR), SRC_REC                : 0776
                              0D  11 00148        BRB     22$                               : 0777
                    57    01  A3  D0 0014A 19$:   MOVL    1(DSTPTR), LINE_NUM               : 0786
                    53        05  C0 0014E 20$:   ADDL2   #5, DSTPTR                        : 0787
                              AC  11 00151        BRB     11$                               : 0708
                    57    01  A3  3C 00153 21$:   MOVZWL  1(DSTPTR), LINE_NUM               : 0795
                        0206  31 00157 22$:       BRW     51$                               : 0796
                    50    01  A3  9A 0015A 23$:   MOVZBL  1(DSTPTR), R0                     : 0806
                    57        50  C0 0015E        ADDL2   R0, LINE_NUM                      :
                        0201  31 00161            BRW     52$                               : 0807
         00000000'  EF        01  D0 00164 24$:   MOVL    #1, DBG$SRC_FORMFEED_FLAG         : 0819
                              53  D6 0016B        INCL    DSTPTR                            : 0820
                              90  11 0016D        BRB     11$                               : 0708
                          04  AE  D5 0016F 25$:   TSTL    SRC_DSTPTR                        : 0832
                              0D  12 00172        BNEQ    26$                               :
                   0002832A   8F  DD 00174        PUSHL   #164650                           :
         00000000G  00        01  FB 0017A        CALLS   #1, LIB$SIGNAL                    :
                    51    01  A3  9E 00181 26$:   MOVAB   1(DSTPTR), R1                     : 0839
                          20  AE  D4 00185        CLRL    32(SP)                            : 0837
                    0A        63  91 00188        CMPB    (DSTPTR), #10                     :
                              08  12 0018B        BNEQ    27$                               :
                          20  AE  D6 0018D        INCL    32(SP)                            :
                    50        61  3C 00190        MOVZWL  (R1), R0                          : 0839
                              03  11 00193        BRB     28$                               :
                    50        61  9A 00195 27$:   MOVZBL  (R1), R0                          : 0841
              18    AE    FF A047 9E 00198 28$:   MOVAB   -1(R0)[LINE_NUM], HIGH_LINE_NUM   :
              08    AC        57  D1 0019E        CMPL    LINE_NUM, LINENO                  : 0848
                              05  14 001A2        BGTR    29$                               :
              18    AE    08  AC  D1 001A4        CMPL    LINENO, HIGH_LINE_NUM             : 0849
                              03  15 001A9 29$:   BLEQ    30$                               :
                        019A  31 001AB            BRW     48$                               :
        50    08    AC        57  C3 001AE 30$:   SUBL3   LINE_NUM, LINENO, R0              : 0867
        5A          50        5B  C1 001B3        ADDL3   SRC_REC, R0, RECORD_NUMBER        :
                          0C  AE  D4 001B7        CLRL    FORMFEED_COUNT                    : 0868
              03 00000000'    EF  E9 001BA        BLBC    DBG$SRC_FORMFEED_FLAG, 31$        : 0869
                        0176  31 001C1            BRW     47$                               :
                              7E  D4 001C4 31$:   CLRL    -(SP)                             : 0884
                          30  AE  9F 001C6        PUSHAB  SFCBPTR                           :
                          0C  AE  DD 001C9        PUSHL   SRC_DSTPTR                        :
              0000V  CF       03  FB 001CC        CALLS   #3, DBG$SRC_OPEN_FILE             :
                    52    2C  AE  D0 001D1        MOVL    SFCBPTR, R2                       : 0891
                    56    20  A2  D0 001D5        MOVL    32(R2), SRCNAM                    :
        01          02    08  A2  8F 001D9        CASEB   8(R2), #2, #1                     : 0892
                  00BD       001C    001DE 32$:   .WORD   33$-32$,-                         :
                                                          41$-32$                           :
                 00000000'    EF  9F 001E2        PUSHAB  P.AAD                             : 1008
                              01  DD 001E8        PUSHL   #1                                :
                 00028362     8F  DD 001EA        PUSHL   #164706                           :
         00000000G  00        03  FB 001F0        CALLS   #3, LIB$SIGNAL                    :
                        013C  31 001F7            BRW     46$                               :
                    55    1C  A2  D0 001FA 33$:   MOVL    28(R2), SRCRAB                    : 0904
                    58    10  A2  D0 001FE        MOVL    16(R2), RFA_TABLE                 : 0905
                    01    30  A2  D1 00202        CMPL    48(R2), #1                        : 0906
```

```
                                 21  12 [illegible]        BNEQ    35$                                   ;
                                 55  DD [illegible]        PUSHL   SRCRAB                                ; 0909
              00000000G  00      01  FB [illegible]        CALLS   #1, SYS$FIND                          ;
                         59      50  D0 [illegible]        MOVL    R0, STATUS                            ;
                         09      59  E8 [illegible]        BLBS    STATUS, 34$                           ; 0910
                                 59  DD [illegible]        PUSHL   STATUS                                ;
              00000000G  00      01  FB [illegible]        CALLS   #1, LIB$SIGNAL                        ;
                         68  10  A5  D0 00220 34$:         MOVL    16(SRCRAB), (RFA_TABLE)               ; 0912
                    04   A8  14  A5  B0 00224              MOVW    20(SRCRAB), 4(RFA_TABLE)              ; 0914
                    30   A2      01  D0 00229 35$:         MOVL    #1, 48(R2)                            ; 0921
                                 55  DD 0022D              PUSHL   SRCRAB                                ; 0922
              00000000G  00      01  FB 0022F              CALLS   #1, SYS$REWIND                        ;
                         59      50  D0 00236              MOVL    R0, STATUS                            ;
                         09      59  E8 00239              BLBS    STATUS, 36$                           ; 0923
                                 59  DD 0023C              PUSHL   STATUS                                ;
              00000000G  00      01  FB 0023E              CALLS   #1, LIB$SIGNAL                        ;
                             1E  A5  94 00245 36$:         CLRB    30(SRCRAB)                            ; 0924
                    14   AE  01  AA  9E 00248              MOVAB   1(R10), 20(SP)                        ; 0951
          58        30   A2      01  C3 0024D              SUBL3   #1, 48(R2), I                         ;
                                 40  11 00252              BRB     40$                                   ;
                                 55  DD 00254 37$:         PUSHL   SRCRAB                                ; 0929
              00000000G  00      01  FB 00256              CALLS   #1, SYS$GET                           ;
                         59      50  D0 0025D              MOVL    R0, STATUS                            ;
                         18      59  E8 00260              BLBS    STATUS, 38$                           ; 0930
                                 59  DD 00263              PUSHL   STATUS                                ; 0932
                             04  A6  DD 00265              PUSHL   4(SRCNAM)                             ;
                         7E  03  A6  9A 00268              MOVZBL  3(SRCNAM), -(SP)                      ;
                                 02  DD 0026C              PUSHL   #2                                    ;
                       00028D00  8F  DD 0026E              PUSHL   #167168                               ;
              00000000G  00      05  FB 00274              CALLS   #5, LIB$SIGNAL                        ;
                         0C 00000000' EF 91 0027B 38$:     CMPB    DBG$SRC_REC_BUF, #12                  ; 0939
                                 0B  12 00282              BNEQ    39$                                   ;
                         01  22  A5  B1 00284              CMPW    34(SRCRAB), #1                        ; 0940
                                 05  12 00288              BNEQ    39$                                   ;
                             0C  AE  D6 0028A              INCL    FORMFEED_COUNT                        ; 0942
                                 C5  11 0028D              BRB     37$                                   ;
                    30   A2  14  AE  D0 0028F 39$:         MOVL    20(SP), 48(R2)                        ; 0951
          BC             58      5A  F3 00294 40$:         AOBLEQ  RECORD_NUMBER, I, 37$                 ; 0925
                               009B  31 00298              BRW     46$                                   ; 0892
                         5A  30  A2  D1 0029B 41$:         CMPL    48(R2), RECORD_NUMBER                 ; 0966
                                 2A  15 0029F              BLEQ    42$                                   ;
                    30   AE  34  A2  D0 002A1              MOVL    52(R2), TXTRFA                        ; 0969
                    34   AE  38  A2  3C 002A6              MOVZWL  56(R2), TXTRFA+4                      ; 0970
                    30   A2      01  D0 002AB              MOVL    #1, 48(R2)                            ; 0971
                             30  AE  9F 002AF              PUSHAB  TXTRFA                                ; 0972
                             3C  A2  9F 002B2              PUSHAB  60(R2)                                ;
              00000000G  00      02  FB 002B5              CALLS   #2, LBR$FIND                          ;
                         59      50  D0 002BC              MOVL    R0, STATUS                            ;
                         09      59  E8 002BF              BLBS    STATUS, 42$                           ; 0973
                                 59  DD 002C2              PUSHL   STATUS                                ;
              00000000G  00      01  FB 002C4              CALLS   #1, LIB$SIGNAL                        ;
                    40   AE    0100  8F  B0 002CB 42$:     MOVW    #256, INPTR                           ; 0980
                    44   AE 00000000' EF 9E 002D1          MOVAB   DBG$SRC_REC_BUF, INPTR+4              ; 0981
                    3C   AE 00000000' EF 9E 002D9          MOVAB   DBG$SRC_REC_BUF, OUTPTR+4             ; 0982
                         58  3C  A2  9E 002E1              MOVAB   60(R2), R8                            ; 0987
          55        30   A2      01  C3 002E5              SUBL3   #1, 48(R2), I                         ;
                                 41  11 002EA              BRB     45$                                   ;
```

```
                           38  AE  9F 002EC 43$:   PUSHAB  OUTPTR                                      ;
                           44  AE  9F 002EF        PUSHAB  INPTR                                       ;
                               58  DD 002F2        PUSHL   R8                                          ;
              00000000G  00    03  FB 002F4        CALLS   #3, LBR$GET_RECORD                          ;
                         59    50  D0 002FB        MOVL    R0, STATUS                                  ;
                         18    59  E8 002FE        BLBS    STATUS, 44$                                 ; 0988
                               59  DD 00301        PUSHL   STATUS                                      ; 0990
                         04    A6  DD 00303        PUSHL   4(SRCNAM)                                   ;
                    7E   03    A6  9A 00306        MOVZBL  3(SRCNAM), -(SP)                            ;
                               02  DD 0030A        PUSHL   #2                                          ;
                    00028D00   8F  DD 0030C        PUSHL   #167168                                     ;
              00000000G  00    05  FB 00312        CALLS   #5, LIB$SIGNAL                              ;
                    0C 00000000'   EF  91 00319 44$:   CMPB    DBG$SRC_REC_BUF, #12                    ; 0992
                               0B  12 00320        BNEQ    45$                                         ;
                    01   38    AE  B1 00322        CMPW    OUTPTR, #1                                  ; 0993
                               05  12 00326        BNEQ    45$                                         ;
                         0C    AE  D6 00328        INCL    FORMFEED_COUNT                              ; 0995
                               BF  11 0032B        BRB     43$                                         ;
       BB          55          5A  F3 0032D 45$:   AOBLEQ  RECORD_NUMBER, I, 43$                       ; 0983
            30     A2    01    AA  9E 00331        MOVAB   1(R10), 48(R2)                              ; 1004
                   5A    0C    AE  C0 00336 46$:   ADDL2   FORMFEED_COUNT, RECORD_NUMBER               ; 1012
                   50    2C    AE  D0 0033A 47$:   MOVL    SFCBPTR, R0                                 ; 1017
            0C     BC    20    A0  D0 0033E        MOVL    32(R0), @NAMBLKPTR                          ;
            10     BC          5A  D0 00343        MOVL    RECORD_NUMBER, @RECNO                       ; 1018
                               04     00347        RET                                                 ; 0851
                   05    20    AE  E9 00348 48$:   BLBC    32(SP), 49$                                 ; 1024
                         50    61  3C 0034C        MOVZWL  (R1), R0                                    ; 1026
                               03  11 0034F        BRB     50$                                         ;
                         50    61  9A 00351 49$:   MOVZBL  (R1), R0                                    ; 1028
                         5B    50  C0 00354 50$:   ADDL2   R0, SRC_REC                                 ;
       57   18     AE          01  C1 00357        ADDL3   #1, HIGH_LINE_NUM, LINE_NUM                 ; 1031
                   05    20    AE  E9 0035C        BLBC    32(SP), 52$                                 ; 1032
                         53    03  C0 00360 51$:   ADDL2   #3, DSTPTR                                  ; 1034
                               03  11 00363        BRB     53$                                         ;
                         53    02  C0 00365 52$:   ADDL2   #2, DSTPTR                                  ; 1036
                             FD1B  31 00368 53$:   BRW     6$                                          ; 0708
  FCD6      08     AE    01    1C  BE  F1 0036B 54$:   ACBL    @MODSRCTBL, #1, COUNT, 3$               ; 0680
                   48    AE    8B  8F  9A 00373 55$:   MOVZBL  #139, BUF_DESC                          ; 1052
                   4C    AE    51  AE  9E 00378        MOVAB   BUFFER+1, BUF_DESC+4                    ; 1053
                               7E  D4 0037D        CLRL    -(SP)                                       ; 1054
                    00000000'  EF  9F 0037F        PUSHAB  P.AAE                                       ;
                         50    AE  9F 00385        PUSHAB  BUF_DESC                                    ;
              00000000G  00    03  FB 00388        CALLS   #3, DBG$FORMAT_FAO_OUT                      ;
                         08    AC  DD 0038F        PUSHL   LINENO                                      ; 1055
                    00000000'  EF  9F 00392        PUSHAB  P.AAF                                       ;
                         50    AE  9F 00398        PUSHAB  BUF_DESC                                    ;
              00000000G  00    03  FB 0039B        CALLS   #3, DBG$FORMAT_FAO_OUT                      ;
                               7E  D4 003A2        CLRL    -(SP)                                       ; 1056
                    00000000'  EF  9F 003A4        PUSHAB  P.AAG                                       ;
                         50    AE  9F 003AA        PUSHAB  BUF_DESC                                    ;
              00000000G  00    03  FB 003AD        CALLS   #3, DBG$FORMAT_FAO_OUT                      ;
                         28    AE  DD 003B4        PUSHL   MODNAMEPTR                                  ; 1057
                    00000000'  EF  9F 003B7        PUSHAB  P.AAH                                       ;
                         50    AE  9F 003BD        PUSHAB  BUF_DESC                                    ;
              00000000G  00    03  FB 003C0        CALLS   #3, DBG$FORMAT_FAO_OUT                      ;
      50    AE     8B    8F    48  AE  83 003C7        SUBB3   BUF_DESC, #139, BUFFER                  ; 1058
                         50    AE  9F 003CE        PUSHAB  BUFFER                                      ; 1059
```

```
                                        01  DD 003D1        PUSHL   #1                            ;
                              00028E18  8F  DD 003D3        PUSHL   #167448                       ;
                 00000000G  00          03  FB 003D9        CALLS   #3, LIB$SIGNAL                ;
                                            04 003E0        RET                                   ; 1060

; Routine Size:  993 bytes,    Routine Base:  DBG$CODE + 01DA
```

```
1 GLOBAL ROUTINE DBG$SRC_LNUM_RANGE(MODRSTPTR, LOWLNUM, HIGHLNUM): NOVALUE =
1 !
1 ! FUNCTION
1 !       This routine accepts a Module RST Entry pointer as input and returns
1 !       the lowest and the highest line numbers within that module as output.
1 !       It is called by the screen facility to determine the line number range
1 !       associated with the source display for a given module.
1 !
1 !       To perform its function, this routine scans through all the Source Line
1 !       Correlation DST Records for the specified module from beginning to end
1 !       to determine the minimum and maximum declared line numbers.
1 !
1 ! INPUTS
1 !       MODRSTPTR - A pointer to the Module RST Entry of the module whose mini-
1 !                   mum and maximum line numbers are to be looked up.
1 !
1 !       LOWLNUM - The address of a longword to receive the lowest line number
1 !                   in the module.
1 !
1 !       HIGHLNUM - The address of a longword to receive the highest line number
1 !                   in the module.
1 !
1 ! OUTPUTS
1 !       LOWLNUM - The lowest line number in the module is returned to the
1 !                   LOWLNUM location.
1 !
1 !       HIGHLNUM - The highest line number in the module is returned to the
1 !                   HIGHLNUM location.
1 !
1
2     BEGIN
2
2     MAP
2       MODRSTPTR: REF RST$ENTRY,         ! Pointer to Module RST Entry
2       LOWLNUM: REF VECTOR[1,LONG],      ! Pointer to low line number location
2       HIGHLNUM: REF VECTOR[1,LONG];     ! Pointer to high line number location
2
2     LOCAL
2       DSTPTR: REF DST$SRC_COMMAND,      ! Pointer to the Source File Correlation
2                                         !             Command
2       DSTREC_PTR: REF DST$RECORD,       ! Pointer to the Source File Correlation
2                                         !             DST record
2       HIGHEST_LNUM,                     ! Highest line number in module so far
2       LENGTH,                           ! The length of the current DST record
2       LINE_NUM,                         ! The current listing line number
2       LOWEST_LNUM,                      ! Lowest line number in module so far
2       MODSRCTBL: REF VECTOR[,LONG];     ! Pointer to the list of Source File
2                                         !             Correlation DST Record pointers
2                                         !             for the current module
2
2
2
2     ! Make sure Module RST pointer is not null.
2     !
2     IF .MODRSTPTR EQL 0
2     THEN
2       $DBG_ERROR('DBGSOURCE\DBG$SRC_HIGHEST_LNUM 10');
```

```
	! Make sure Module Source Table pointer in module's Module RST Entry
	! points to a list of Source File DST Record Pointers.
	!
	LOWLNUM[0] = 0;
	HIGHLNUM[0] = 0;
	IF .MODRSTPTR[RST$L_MODSRCTBL] EQL 0 THEN RETURN;


	! Initialize flags and counters.
	!
	LINE_NUM = 1;
	LOWEST_LNUM = 1000000000;
	HIGHEST_LNUM = 0;


	! Loop thru Module's Source Line Correlation DST Records.
	!
	MODSRCTBL = .MODRSTPTR[RST$L_MODSRCTBL];
	INCR COUNT FROM 1 TO .MODSRCTBL[0] DO
	    BEGIN


	    ! Get Source Line Correlation DST Record pointer.
	    !
	    DSTREC_PTR = .MODSRCTBL[.COUNT];
	    IF .DSTREC_PTR[DST$B_TYPE] NEQ DST$K_SOURCE
	    THEN
		$DBG_ERROR('DBGSOURCE\DBG$SRC_HIGHEST_LNUM 20');


	    ! Scan through the Source File DST Record to decode the Source File
	    ! Correlation Commands.  Pick up the length of this DST record and set
	    ! DSTPTR to point to the first command entry.
	    !
	    LENGTH = .DSTREC_PTR[DST$B_LENGTH];
	    DSTPTR = DSTREC_PTR[DST$A_SRC_FIRST_CMD];
	    WHILE (.DSTPTR - .DSTREC_PTR) LEQ .LENGTH DO
		BEGIN


		! Pick up the command code defined in Source File DST Record
		! and use it as a CASE index so each significant command code
		! can be interpreted individually.
		!
		CASE .DSTPTR[DST$B_SRC_COMMAND] FROM DST$K_SRC_MIN_CMD TO
						     DST$K_SRC_MAX_CMD OF
		    SET


		    ! Declare Source File Command.
		    !
		    [DST$K_SRC_DECLFILE]:
			DSTPTR = .DSTPTR + .DSTPTR[DST$B_SRC_DF_LENGTH] + 2;

```

```
    ! Set Source File Command.
    !
    [DST$K_SRC_SETFILE]:
        DSTPTR = .DSTPTR + 3;


    ! Set Source Record Number Long Command.
    !
    [DST$K_SRC_SETREC_L]:
        DSTPTR = .DSTPTR + 5;


    ! Set Source Record Number Word Command.
    !
    [DST$K_SRC_SETREC_W]:
        DSTPTR = .DSTPTR + 3;


    ! Set Line Number Long Command.  Set the current listing line
    ! number.  Advance DSTPTR.
    !
    [DST$K_SRC_SETLNUM_L]:
        BEGIN
        LINE_NUM = .DSTPTR[DST$L_SRC_UNSLONG];
        DSTPTR = .DSTPTR + 5;
        END;


    ! Set Line Number Word Command. (More compact form).
    !
    [DST$K_SRC_SETLNUM_W]:
        BEGIN
        LINE_NUM = .DSTPTR[DST$W_SRC_UNSWORD];
        DSTPTR = .DSTPTR + 3;
        END;


    ! Increment Line Number Byte Command.  Increment the current
    ! listing line number by a byte value given in this command.
    ! Advance DSTPTR.
    !
    [DST$K_SRC_INCRLNUM_B]:
        BEGIN
        LINE_NUM = .LINE_NUM + .DSTPTR[DST$B_SRC_UNSBYTE];
        DSTPTR = .DSTPTR + 2;
        END;


    ! Do not skip the form feed.
    !
    [DST$K_SRC_FORMFEED]:
        DSTPTR = .DSTPTR + 1;


    ! Define N Separate Lines Word Command.
    !
    [DST$K_SRC_DEFLINES_W]:
```

```
5                       BEGIN
5                       LOWEST_LNUM = MIN(.LOWEST_LNUM, .LINE_NUM);
5                       LINE_NUM = .LINE_NUM + .DSTPTR[DST$W_SRC_UNSWORD];
5                       HIGHEST_LNUM = MAX(.HIGHEST_LNUM, .LINE_NUM - 1);
5                       DSTPTR = .DSTPTR + 3;
4                       END;
4
4
4                   ! Define N Separate Lines Byte Command.
4                   !
4                   [DST$K_SRC_DEFLINES_B]:
5                       BEGIN
5                       LOWEST_LNUM = MIN(.LOWEST_LNUM, .LINE_NUM);
5                       LINE_NUM = .LINE_NUM + .DSTPTR[DST$B_SRC_UNSBYTE];
5                       HIGHEST_LNUM = MAX(.HIGHEST_LNUM, .LINE_NUM - 1);
5                       DSTPTR = .DSTPTR + 2;
4                       END;
4
4
4                   ! Any other command code constitutes an error in the DST.
4                   !
4                   [INRANGE, OUTRANGE]:
5                       BEGIN
5                       SIGNAL(DBG$_INVDSTREC);
4                       END;
4
4                   TES;
4
3               END;                            ! End of WHILE Loop through commands.
3
2           END;                                ! End of INCR Loop through DST records.
2
2
2       ! Return the lowest and highest line numbers that we found.  Then return
2       ! to the caller.
2       !
2       IF .LOWEST_LNUM GTR .HIGHEST_LNUM THEN LOWEST_LNUM = 0;
2       LOWLNUM[0] = .LOWEST_LNUM;
2       HIGHLNUM[0] = .HIGHEST_LNUM;
2       RETURN;
2
1       END;


                                                                .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

24 47 42 44 5C 45 43 52 55 4F 53 47 42 44 21 0009A P.AAI:  .ASCII  \!DBGSOURCE\<92>\DBG$SRC_HIGHEST_LNUM 10\   ;
55 4E 4C 5F 54 53 45 48 47 49 48 5F 43 52 53 000A9                                                          ;
                                 30 31 20 4D 000B8                                                          ;
24 47 42 44 5C 45 43 52 55 4F 53 47 42 44 21 000BC P.AAJ:  .ASCII  \!DBGSOURCE\<92>\DBG$SRC_HIGHEST_LNUM 20\   ;
55 4E 4C 5F 54 53 45 48 47 49 48 5F 43 52 53 000CB                                                          ;
                                 30 32 20 4D 000DA                                                          ;


                                                                .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0
```

```
                                07FC 00000        .ENTRY  DBG$SRC_LNUM_RANGE, Save R2,R3,R4,R5,R6,R7,-;  1061
                                                          R8,R9,R10
                5A 0000000G  00  9E 00002        MOVAB   LIB$SIGNAL, R10                              :
                52       04  AC  D0 00009        MOVL    MODRSTPTR, R2                                ;  1115
                         11  12 0000D            BNEQ    1$                                           :
                00000000'  EF  9F 0000F          PUSHAB  P.AAI                                        ;  1117
                         01  DD 00015            PUSHL   #1                                           :
                00028362  8F  DD 00017           PUSHL   #164706                                      :
                6A       03  FB 0001D            CALLS   #3, LIB$SIGNAL                               :
                08       BC  D4 00020  1$:       CLRL    @LOWLNUM                                     ;  1123
                0C       BC  D4 00023            CLRL    @HIGHLNUM                                    ;  1124
                24       A2  D5 00026            TSTL    36(R2)                                       ;  1125
                         01  12 00029            BNEQ    2$                                           :
                         04 0002B                RET                                                  :
                53       01  D0 0002C  2$:       MOVL    #1, LINE_NUM                                 ;  1130
                55 3B9ACA00 8F  D0 0002F         MOVL    #1000000000, LOWEST_LNUM                     ;  1131
                58       24  A2  D0 00036        MOVL    36(R2), MODSRCTBL                            ;  1137
                         56  7C 0003A            CLRQ    HIGHEST_LNUM                                 ;  1132
                    00DA  31 0003C  3$:          BRW     25$                                          ;  1138
                54     6847  D0 0003F  4$:       MOVL    (MODSRCTBL)[COUNT], DSTREC_PTR               ;  1144
          9B    8F       01  A4  91 00043        CMPB    1(DSTREC_PTR), #155                          ;  1145
                         11  13 00048            BEQL    5$                                           :
                00000000'  EF  9F 0004A          PUSHAB  P.AAJ                                        ;  1147
                         01  DD 00050            PUSHL   #1                                           :
                00028362  8F  DD 00052           PUSHL   #164706                                      :
                6A       03  FB 00058            CALLS   #3, LIB$SIGNAL                               :
                59       64  9A 0005B  5$:       MOVZBL  (DSTREC_PTR), LENGTH                         ;  1154
                52       02  A4  9E 0005E        MOVAB   2(R4), DSTPTR                                ;  1155
          50    52       54  C3 00062  6$:       SUBL3   DSTREC_PTR, DSTPTR, R0                       ;  1156
                59       50  D1 00066            CMPL    R0, LENGTH                                   :
                         D1  14 00069            BGTR    3$                                           :
          0F    01       62  8F 0006B            CASEB   (DSTPTR), #1, #15                            ;  1164
 0079     003A  0079   002B    0006F  7$:        .WORD   10$-7$,-                                     :
 0020     0045  003F   0036    00077                     20$-7$,-                                     :
 0020     007E  0052   0020    0007F                     13$-7$,-                                     :
 004E     0020  0020   0020    00087                     20$-7$,-                                     :
                                                         12$-7$,-                                     :
                                                         14$-7$,-                                     :
                                                         15$-7$,-                                     :
                                                         8$-7$,-                                      :
                                                         8$-7$,-                                      :
                                                         17$-7$,-                                     :
                                                         21$-7$,-                                     :
                                                         8$-7$,-                                      :
                                                         8$-7$,-                                      :
                                                         8$-7$,-                                      :
                                                         8$-7$,-                                      :
                                                         16$-7$                                       :
                0002832A  8F  DD 0008F  8$:      PUSHL   #164650                                      ;  1255
                6A       01  FB 00095            CALLS   #1, LIB$SIGNAL                               :
                         C8  11 00098  9$:       BRB     6$                                           ;  1164
                50       01  A2  9A 0009A  10$:  MOVZBL  1(DSTPTR), R0                                ;  1172
                52    02 A042  9E 0009E          MOVAB   2(R0)[DSTPTR], DSTPTR                        :
                         BD  11 000A3  11$:      BRB     6$                                           :
                53       01  A2  D0 000A5  12$:  MOVL    1(DSTPTR), LINE_NUM                          ;  1198
                52       05  C0 000A9  13$:      ADDL2   #5, DSTPTR                                   ;  1199
```

```
                                   B4  11 000AC          BRB     6$                                      ; 1164
                   53      01      A2  3C 000AE  14$:    MOVZWL  1(DSTPTR), LINE_NUM                     ; 1207
                                   34  11 000B2          BRB     20$                                     ; 1208
                   50      01      A2  9A 000B4  15$:    MOVZBL  1(DSTPTR), R0                           ; 1218
                   53              50  C0 000B8          ADDL2   R0, LINE_NUM                            ;
                                   57  11 000BB          BRB     24$                                     ; 1219
                                   52  D6 000BD  16$:    INCL    DSTPTR                                  ; 1226
                                   A1  11 000BF          BRB     6$                                      ;
                   50              55  D0 000C1  17$:    MOVL    LOWEST_LNUM, R0                         ; 1233
                   53              50  D1 000C4          CMPL    R0, LINE_NUM                            ;
                                   03  15 000C7          BLEQ    18$                                     ;
                   50              53  D0 000C9          MOVL    LINE_NUM, R0                            ;
                   55              50  D0 000CC  18$:    MOVL    R0, LOWEST_LNUM                         ;
                   50      01      A2  3C 000CF          MOVZWL  1(DSTPTR), R0                           ; 1234
                   53              50  C0 000D3          ADDL2   R0, LINE_NUM                            ;
                   51      FF      A3  9E 000D6          MOVAB   -1(R3), R1                              ; 1235
                   50              56  D0 000DA          MOVL    HIGHEST_LNUM, R0                        ;
                   51              50  D1 000DD          CMPL    R0, R1                                  ;
                                   03  18 000E0          BGEQ    19$                                     ;
                   50              51  D0 000E2          MOVL    R1, R0                                  ;
                   56              50  D0 000E5  19$:    MOVL    R0, HIGHEST_LNUM                        ;
                   52              03  C0 000E8  20$:    ADDL2   #3, DSTPTR                              ; 1236
                                   AB  11 000EB          BRB     9$                                      ; 1164
                   50              55  D0 000ED  21$:    MOVL    LOWEST_LNUM, R0                         ; 1244
                   53              50  D1 000F0          CMPL    R0, LINE_NUM                            ;
                                   03  15 000F3          BLEQ    22$                                     ;
                   50              53  D0 000F5          MOVL    LINE_NUM, R0                            ;
                   55              50  D0 000F8  22$:    MOVL    R0, LOWEST_LNUM                         ;
                   50      01      A2  9A 000FB          MOVZBL  1(DSTPTR), R0                           ; 1245
                   53              50  C0 000FF          ADDL2   R0, LINE_NUM                            ;
                   51      FF      A3  9E 00102          MOVAB   -1(R3), R1                              ; 1246
                   50              56  D0 00106          MOVL    HIGHEST_LNUM, R0                        ;
                   51              50  D1 00109          CMPL    R0, R1                                  ;
                                   03  18 0010C          BGEQ    23$                                     ;
                   50              51  D0 0010E          MOVL    R1, R0                                  ;
                   56              50  D0 00111  23$:    MOVL    R0, HIGHEST_LNUM                        ;
                   52              02  C0 00114  24$:    ADDL2   #2, DSTPTR                              ; 1247
                                   8A  11 00117          BRB     11$                                     ; 1156
     FF20          57      01      68  F1 00119  25$:    ACBL    (MODSRCTBL), #1, COUNT, 4$              ; 1138
                   56              55  D1 0011F          CMPL    LOWEST_LNUM, HIGHEST_LNUM               ; 1268
                                   02  15 00122          BLEQ    26$                                     ;
                                   55  D4 00124          CLRL    LOWEST_LNUM                             ;
              08   BC              55  D0 00126  26$:    MOVL    LOWEST_LNUM, @LOWLNUM                   ; 1269
              0C   BC              56  D0 0012A          MOVL    HIGHEST_LNUM, @HIGHLNUM                 ; 1270
                                       04 0012E          RET                                             ; 1273

; Routine Size:  303 bytes,    Routine Base:  DBG$CODE + 05BB
```

```
1 ROUTINE DBG$SRC_OPEN(SFCB_KIND,SRCDSTPTR,PROPER_FLAG) =
1 !
1 ! FUNCTION
1 !       This routine is called by DBG$SRC_OPEN_FILE.  It actually does the
1 !       open for RMS or LBR file.
1 !
1 !       The opened file is then checked to see if it is the right file.  This
1 !       consists of checking the Creation Date and Time and several other file
1 !       attributes.  If this search fails to find the desired file, the proper
1 !       flag is set to indicate the status.  If file could not be open at all,
1 !       then the returned status from RMS routines is returned.
1 !
1 !       If the desired file is a module within a source library, then checking
1 !       is modified to check the library in the given directory which has a
1 !       module of the desired name which in turn has the proper insertion date.
1 !       If this checking fails to find the desired module in the library,
1 !       the proper flag is set to indicate the proper module is not found.
1 !       If no library at all could be opened, then status from LBR is returned.
1 !
1 ! INPUTS
1 !       SFCB_KIND - Indictates whether the file to be opened is an RMS file
1 !                   (SFCB_KIND = 2) or an LBR file (SFCB_KIND = 3).
1 !
1 !       SRCDSTPTR - Pointer to the Declare Source File DST command which declares
1 !                   the file to be opened.
1 !
1 ! OUTPUTS
1 !       PROPER_FLAG - 0: Proper file is found,
1 !                     non-0: file or module in the library is not proper file.
1 !
1 !                     The ''proper'' is defined as the first version with the
1 !                     same creation date and time, the same file size, and
1 !                     the same record format and file organization as specified
1 !                     in the Source File Correlation DST Record.  The ''proper''
1 !                     file is found means such file is found during the search
1 !                     directory dir1, ..., dirN process.  If the ''proper''
1 !                     version of the file is not found in any of dir1, ..., dirN,
1 !                     the latest version of the file in the first directory on
1 !                     the list to have a version of it is used.  A warning is
1 !                     issued.  If no file can be found at all, a warning
1 !                     message to this effect is issued.
1 !
1 !       STATUS - returned value from RMS or LBR, return value for this routine.
1 !
1
2     BEGIN
2
2     MAP
2       PROPER_FLAG: REF VECTOR[1],       ! Flag set to indicate the status of
2                                         !         the opened file
2       SRCDSTPTR: REF DST$SRC_COMMAND;   ! Pointer to the Declare Source File
2                                         !         DST command
2
2
2     LOCAL
2       BUFDESC: BLOCK[8,BYTE],           ! String descriptor, it contains length
2                                         !         and address of the buffer
```

```
        BUFLEN,                             ! Contains the length of the returned
                                            !       module header
        DSTCDT: REF VECTOR[2],              ! Pointer to the creation date and time
                                            !       in the DST Declare Source File
                                            !       command
        LIBMOD_PTR: REF DST$SRC_CMDTRLR,    ! Points to the Source Library Module
                                            !       name in the Source File
                                            !       Correlation DST Record
        MHDPTR: REF BLOCK[,BYTE],           ! Points to Module Header
        MODPTR: BLOCK[8,BYTE],              ! Length and address of module name
        SRCXABDAT: REF $XABDAT_DECL,        ! XABDAT for RMS file
        SRCXABFHC: REF $XABFHC_DECL,        ! XABFHC for RMS file
        STATUS,                             ! RMS status or LBR status
        TXTRFA: VECTOR[2],                  ! The RFA of the module text in library
        XABCDT: REF VECTOR[2],              ! Pointer to the creation date and time
                                            !       in the RMS XAB block
        XABRDT: REF VECTOR[2];              ! Pointer to the revision date and time
                                            !       in the RMS XAB block



    ! Determine whether we have an RMS or LBR file.
    !
    PROPER_FLAG[0] = 0;
    CASE .SFCB_KIND FROM SFCB$K_RMSFILE TO SFCB$K_LBRFILE OF
        SET


        ! Open an RMS source file and check its attributes to see if this is
        ! the exact file the DST Declare Source File Command specifies.
        !
        [SFCB$K_RMSFILE]:
            BEGIN


            ! Open the file and connect the RAB to the FAB.
            !
            STATUS = $OPEN(FAB = .DBG$SRC_SFCB_PTR[SFCB$L_FABPTR]);
            IF NOT .STATUS THEN RETURN .STATUS;
            STATUS = $CONNECT(RAB = .DBG$SRC_SFCB_PTR[SFCB$L_RABPTR]);
            IF NOT .STATUS THEN RETURN .STATUS;


            ! Check that the file has the proper creation date and time.  (We
            ! also accept a matching revision date and time for compatibility
            ! with the usage prior to VMS V4.0.)  If not, set PROPER_FLAG to 1.
            !
            SRCXABDAT = .DBG$SRC_SFCB_PTR[SFCB$L_XABDATPTR];
            SRCXABFHC = .DBG$SRC_SFCB_PTR[SFCB$L_XABFHCPTR];
            XABCDT = SRCXABDAT[XAB$Q_CDT];
            XABRDT = SRCXABDAT[XAB$Q_RDT];
            DSTCDT = SRCDSTPTR[DST$Q_SRC_DF_RMS_CDT];
            IF (.XABCDT[0] NEQ .DSTCDT[0] OR .XABCDT[1] NEQ .DSTCDT[1]) AND
               (.XABRDT[0] NEQ .DSTCDT[0] OR .XABRDT[1] NEQ .DSTCDT[1])
            THEN
                PROPER_FLAG[0] = 1;
```

```
            ! Also check that the file's byte size and record format and file
            ! organization is as specified in the DST.  If not, set PROPER_FLAG
            ! to 2, 3, or 4.
            !
            IF .SRCXABFHC[XAB$L_EBK] NEQ .SRCDSTPTR[DST$L_SRC_DF_RMS_EBK]
            THEN
                PROPER_FLAG[0] = 2;

            IF .SRCXABFHC[XAB$W_FFB] NEQ .SRCDSTPTR[DST$W_SRC_DF_RMS_FFB]
            THEN
                PROPER_FLAG[0] = 3;

            IF .SRCXABFHC[XAB$B_RFO] NEQ .SRCDSTPTR[DST$B_SRC_DF_RMS_RFO]
            THEN
                PROPER_FLAG[0] = 4;

            END;


        ! Open an Source Library File.  Then look up and open the desired
        ! module within that source library.  Also compare the module's
        ! inseration date and time with that in the DST Declare Source
        ! File command.
        !
        [SFCB$K_LBRFILE]:
            BEGIN


            ! Open the desired source library.
            !
            STATUS = LBR$INI_CONTROL(DBG$SRC_SFCB_PTR[SFCB$L_LBRINDEX],
                UPLIT(LBR$C_READ), UPLIT(LBR$C_TYP_TXT), .DBG$SRC_SFCB_PTR[SFCB$L_NAMPTR]);
            IF NOT .STATUS THEN RETURN .STATUS;
            STATUS = LBR$OPEN(DBG$SRC_SFCB_PTR[SFCB$L_LBRINDEX]);
            IF NOT .STATUS THEN RETURN .STATUS;


            ! Look up the desired module and open the module header.
            !
            LIBMOD_PTR
                = SRCDSTPTR[DST$A_SRC_DF_FILENAME] + .SRCDSTPTR[DST$B_SRC_DF_FILENAME];
            MODPTR[DSC$W_LENGTH] = .LIBMOD_PTR[DST$B_SRC_DF_LIBMODNAME];
            MODPTR[DSC$A_POINTER] = LIBMOD_PTR[DST$A_SRC_DF_LIBMODNAME];
            STATUS = LBR$LOOKUP_KEY(DBG$SRC_SFCB_PTR[SFCB$L_LBRINDEX],MODPTR,TXTRFA);
            IF NOT .STATUS THEN RETURN .STATUS;


            ! Get module's header information.
            !
            BUFDESC[DSC$W_LENGTH] = SRCBUFSIZE;
            BUFDESC[DSC$A_POINTER] = DBG$SRC_REC_BUF[0];
            STATUS = LBR$SET_MODULE(DBG$SRC_SFCB_PTR[SFCB$L_LBRINDEX],TXTRFA,
                BUFDESC,BUFLEN);
            IF NOT .STATUS THEN RETURN .STATUS;
```

```
3            ! Save the module's start Record File Address (RFA) in the Source
3            ! File Control Block.
3            !
3            DBG$SRC_SFCB_PTR[SFCB$L_CUR_RFA0] = .TXTRFA[0];
3            DBG$SRC_SFCB_PTR[SFCB$W_CUR_RFA4] = .TXTRFA[1];
3
3            ! Check the module insertion date and time.  If it is not the same
3            ! as that in the DST Declare Source File Command, set PROPER_FLAG
3            ! to 1.
3            !
3            MHDPTR = DBG$SRC_REC_BUF[0];
3            XABCDT = MHDPTR[MHD$L_DATIM];
3            DSTCDT = SRCDSTPTR[DST$Q_SRC_DF_RMS_CDT];
3            IF .XABCDT[0] NEQ .DSTCDT[0] OR .XABCDT[1] NEQ .DSTCDT[1]
3            THEN
3                PROPER_FLAG[0] = 1;
3
2            END;
2
2
2        ! Any other kind of Source File Control Block is an internal error.
2        !
2        [INRANGE, OUTRANGE]:
2            $DBG_ERROR('DBGSOURCE\DBG$SRC_OPEN');
2
2        TES;
2
2    ! The source file or module was successfully opened although the file
2    ! attributes (date and time, etc.) may not have matched those in the
2    ! DST.  Return the last status we got from RMS or LBR.
2    !
2    RETURN .STATUS;
2
1    END;


                                                              .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

                                               000DE          .BLKB   2
                                   00000001    000E0 P.AAK:   .LONG   1                                  ;
                                   00000004    000E4 P.AAL:   .LONG   4                                  ;
24  47  42  44  5C  45  43  52  55  4F  53  47  42  44  16    000E8 P.AAM:   .ASCII  <22>\DBGSOURCE\<92>\DBG$SRC_OPEN\   ;
                            4E  45  50  4F  5F  43  52  53    000F7                                      ;

                                                              .EXTRN  SYS$OPEN, SYS$CONNECT

                                                              .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                          01FC 00000 DBG$SRC_OPEN:
                                                              .WORD   Save R2,R3,R4,R5,R6,R7,R8          ; 1274
                  58 00000000'   EF   9E 00002               MOVAB   P.AAM, R8                          ;
                  57 00000000'   EF   9E 00009               MOVAB   DBG$SRC_SFCB_PTR, R7               ;
                  5E             1C   C2 00010               SUBL2   #28, SP                            ;
                  56        0C   AC   D0 00013               MOVL    PROPER_FLAG, R6                    ; 1354
```

```
                                    66  D4 00017        CLRL    (R6)                                       ;
            01         02       04  AC  CF 00019        CASEL   SFCB_KIND, #2, #1                          ; 1355
                     008B     0016         0001E 1$:    .WORD   2$-1$,-
                                                                9$-1$
                                    58  DD 00022        PUSHL   R8                                         ; 1469
                                    01  DD 00024        PUSHL   #1                                         ;
                          00028362  8F  DD 00026        PUSHL   #164706                                    ;
          00000000G    00           03  FB 0002C        CALLS   #3, LIB$SIGNAL                             ;
                                    04     00033        RET                                                ;
                       50           67  D0 00034 2$:    MOVL    DBG$SRC_SFCB_PTR, R0                       ; 1368
                                18  A0  DD 00037        PUSHL   24(R0)                                     ;
          00000000G    00           01  FB 0003A        CALLS   #1, SYS$OPEN                               ;
                       7B           50  E9 00041        BLBC    STATUS, 10$                                ; 1369
                       51           67  D0 00044        MOVL    DBG$SRC_SFCB_PTR, R1                       ; 1370
                                1C  A1  DD 00047        PUSHL   28(R1)                                     ;
          00000000G    00           01  FB 0004A        CALLS   #1, SYS$CONNECT                            ;
                       79           50  E9 00051        BLBC    STATUS, 11$                                ; 1371
                       52           67  D0 00054        MOVL    DBG$SRC_SFCB_PTR, R2                       ; 1378
                       51       24  A2  D0 00057        MOVL    36(R2), SRCXABDAT                          ;
                       55       28  A2  D0 0005B        MOVL    40(R2), SRCXABFHC                          ; 1379
                       53       14  A1  9E 0005F        MOVAB   20(R1), XABCDT                             ; 1380
                       51           0C  C0 00063        ADDL2   #12, XABRDT                                ; 1381
                       54       08  AC  D0 00066        MOVL    SRCDSTPTR, R4                              ; 1382
                       52       05  A4  9E 0006A        MOVAB   5(R4), DSTCDT                              ;
                       62           63  D1 0006E        CMPL    (XABCDT), (DSTCDT)                         ; 1383
                                    07  12 00071        BNEQ    3$                                         ;
                 04    A2       04  A3  D1 00073        CMPL    4(XABCDT), 4(DSTCDT)                       ;
                                    0F  13 00078        BEQL    5$                                         ;
                       62           61  D1 0007A 3$:    CMPL    (XABRDT), (DSTCDT)                         ; 1384
                                    07  12 0007D        BNEQ    4$                                         ;
                 04    A2       04  A1  D1 0007F        CMPL    4(XABRDT), 4(DSTCDT)                       ;
                                    03  13 00084        BEQL    5$                                         ;
                       66           01  D0 00086 4$:    MOVL    #1, (R6)                                   ; 1386
                 0D    A4       10  A5  D1 00089 5$:    CMPL    16(SRCXABFHC), 13(R4)                      ; 1393
                                    03  13 0008E        BEQL    6$                                         ;
                       66           02  D0 00090        MOVL    #2, (R6)                                   ; 1395
                 11    A4       14  A5  B1 00093 6$:    CMPW    20(SRCXABFHC), 17(R4)                      ; 1397
                                    03  13 00098        BEQL    7$                                         ;
                       66           03  D0 0009A        MOVL    #3, (R6)                                   ; 1399
                 13    A4       08  A5  91 0009D 7$:    CMPB    8(SRCXABFHC), 19(R4)                       ; 1401
                                    01  12 000A2        BNEQ    8$                                         ;
                                    04     000A4        RET                                                ;
                       66           04  D0 000A5 8$:    MOVL    #4, (R6)                                   ; 1403
                                    04     000A8        RET                                                ; 1355
                       51           67  D0 000A9 9$:    MOVL    DBG$SRC_SFCB_PTR, R1                       ; 1420
                                20  A1  DD 000AC        PUSHL   32(R1)                                     ;
                                FC  A8  9F 000AF        PUSHAB  P.AAL                                      ;
                                F8  A8  9F 000B2        PUSHAB  P.AAK                                      ;
                                3C  A1  9F 000B5        PUSHAB  60(R1)                                     ; 1419
          00000000G    00           04  FB 000B8        CALLS   #4, LBR$INI_CONTROL                        ;
                       56           50  E9 000BF 10$:   BLBC    STATUS, 12$                                ; 1421
            7E         67           3C  C1 000C2        ADDL3   #60, DBG$SRC_SFCB_PTR, -(SP)               ; 1422
          00000000G    00           01  FB 000C6        CALLS   #1, LBR$OPEN                               ;
                       73           50  E9 000CD 11$:   BLBC    STATUS, 14$                                ; 1423
                       54       08  AC  D0 000D0        MOVL    SRCDSTPTR, R4                              ; 1429
                       51       14  A4  9A 000D4        MOVZBL  20(R4), R1                                 ;
                       51       15 A441 9E 000D8        MOVAB   21(R4)[R1], LIBMOD_PTR                     ;
```

```
          0C   AE          61  9B 000DD          MOVZBW  (LIBMOD_PTR), MODPTR                     ; 1430
          10   AE     01   A1  9E 000E1          MOVAB   1(R1), MODPTR+4                          ; 1431
                      04   AE  9F 000E6          PUSHAB  TXTRFA                                   ; 1432
                      10   AE  9F 000E9          PUSHAB  MODPTR                                   ;
     7E        67          3C  C1 000EC          ADDL3   #60, DBG$SRC_SFCB_PTR, -(SP)             ;
  00000000G    00          03  FB 000F0          CALLS   #3, LBR$LOOKUP_KEY                       ;
               49          50  E9 000F7          BLBC    STATUS, 14$                              ; 1433
          14   AE   0100   8F  B0 000FA          MOVW    #256, BUFDESC                            ; 1438
          18   AE     04   A7  9E 00100          MOVAB   DBG$SRC_REC_BUF, BUFDESC+4               ; 1439
                           5E  DD 00105          PUSHL   SP                                       ; 1440
                      18   AE  9F 00107          PUSHAB  BUFDESC                                  ;
                      0C   AE  9F 0010A          PUSHAB  TXTRFA                                   ;
     7E        67          3C  C1 0010D          ADDL3   #60, DBG$SRC_SFCB_PTR, -(SP)             ;
  00000000G    00          04  FB 00111          CALLS   #4, LBR$SET_MODULE                       ;
               28          50  E9 00118 12$:     BLBC    STATUS, 14$                              ; 1442
               51          67  D0 0011B          MOVL    DBG$SRC_SFCB_PTR, R1                     ; 1448
          34   A1     04   AE  D0 0011E          MOVL    TXTRFA, 52(R1)                           ;
          38   A1     08   AE  B0 00123          MOVW    TXTRFA+4, 56(R1)                         ; 1449
               51     04   A7  9E 00128          MOVAB   DBG$SRC_REC_BUF, MHDPTR                  ; 1456
               53     08   A1  9E 0012C          MOVAB   8(R1), XABCDT                            ; 1457
               52     05   A4  9E 00130          MOVAB   5(R4), DSTCDT                            ; 1458
               62          63  D1 00134          CMPL    (XABCDT), (DSTCDT)                       ; 1459
                           07  12 00137          BNEQ    13$                                      ;
          04   A2     04   A3  D1 00139          CMPL    4(XABCDT), 4(DSTCDT)                     ;
                           03  13 0013E          BEQL    14$                                      ;
               66          01  D0 00140 13$:     MOVL    #1, (R6)                                 ; 1461
                               04 00143 14$:     RET                                              ; 1480

; Routine Size:  324 bytes,    Routine Base:  DBG$CODE + 06EA


; 1355          1481  1
```

```
ROUTINE DBG$SRC_OPEN_FILE(FILDSTPTR, SFCBPTR, STEP_FLAG): NOVALUE =
!
! FUNCTION
!	This routine opens a specified source file (unless the file is already
!	opened) and sets up the corresponding Source File Control Block so that
!	the file can be read with subsequent calls on DBG$SRC_READ_FILE.  In the
!	process, it applies all source directory search rules to make sure it
!	opens the file in the directory desired by the user.
!
!	The routine starts by searching the list of Source File Control Blocks
!	for the input DST pointer.  If such a control block is found, the file
!	is already open and the Source File Control Block (SFCB) pointer can be
!	returned immediately.  The SFCB is moved to the Most Recently Used posi-
!	tion in the doubly linked SFCB list before DBG$SRC_OPEN_FILE returns.
!
!	Otherwise, an SFCB must be allocated for this file.  This is done by
!	finding the Least Recently Used SFCB and closing the file using that
!	control block (if any).  The SFCB is then moved to the Most Recently
!	Used position.  Next, the Source Directory Search List for the current
!	module (which may be the default search list) is found in the linked
!	list pointed to by DBG$SRC_DIR_LIST.
!
!	At this point, the routine is ready to search all directories on the
!	list to locate the desired source file.  For each directory, it does
!	a $PARSE and loops over a $SEARCH system service.  Each file found by
!	$SEARCH is checked to see if it is the right file.  This consists of
!	checking the Creation Date and Time and several other file attributes.
!	If this search fails to find the desired file, the first file returned
!	by $SEARCH for the first directory on the list is used even though its
!	date and time or other attributes may be wrong.  An informational
!	message is signalled in this case.  If no file at all could be found,
!	an error message to that effect is signalled.
!
!	If the desired file is a module within a source library, the search is
!	modified to find the first library in the directory list which has a
!	module of the desired name which in turn has the proper insertion date.
!
!	If no Source Directory Search List was specified for this module (either
!	directly or by default), the file name as it appears in the Declare
!	Source File DST command is taken to be the file we want to open.
!
!	Once the desired file has been identified, that file is opened via the
!	RMS $OPEN and $CONNECT services.  Or, if the source file is a module in
!	a source library, the library and its module are opened via the appro-
!	priate LIBRARIAN (LBR$xxx) calls.  DBG$SRC_OPEN_FILE then returns.
!
!	The routine actually does the open and verify the attributes is in
!	"DBG$SRC_OPEN".
!
! INPUTS
!	FILDSTPTR - A pointer to the Declare Source File command in the Source
!		    File Correlation DST Record which declares the source file to
!		    be opened.  This command includes the desired file name and
!		    other file attributes.
!
!	SFCBPTR - A longword location to receive a pointer to the Source File
!		  Control Block set up for the desired file.
```

```
!
!       STEP_FLAG - Flag set to true to indicate the call is from STEP command.
!
! OUTPUTS
!       SFCBPTR - A Pointer to the Source File Control Block set up for the
!                 opened source file is returned to SFCBPTR.  This pointer
!                 can then be used when later calling DBG$SRC_READ_FILE.
!

    BEGIN

    MAP
        FILDSTPTR:                      ! Pointer to the Declare Source File DST
                REF DST$SRC_COMMAND,    !       command which declares the file
                                        !       to be opened
        SFCBPTR: REF VECTOR[1];         ! Pointer to longword to receive the
                                        !       Source File Control Block pointer

    LOCAL
        BAD_FILE,                       ! Flag to indicate there is a bad
                                        !       file stuck in the SFCB
        BLINK: REF SFCB$BLOCK,          ! Backward link
        DEFAULT: VECTOR[SRCBUFSIZE,BYTE],       ! Contains Default File Name
                                                !       length and filename for RMS
        FIRST_FILE: VECTOR[SRCBUFSIZE,BYTE],    ! Contains first file returned FROM
                                                !       $SEARCH the directory list
        FIRST_TIME,                     ! Flag set to indicate first time $SEARCH
        FLINK: REF SFCB$BLOCK,          ! Forward link
        LIBMOD_PTR: REF DST$SRC_CMDTRLR,! Trailer field in the Declare
                                        !       Source File Command
        LRUPTR: REF SFCB$BLOCK,         ! Pointer to Least Recently Used Source
                                        !       File Control Block
        PROPER_FLAG,                    ! Flag set to indicate proper file/module
                                        !       is used
        SDSL_PTR: REF SDSL$HEADER,      ! Pointer to Source Directory Search
                                        !       List Header Block
        SDSL_DIRPTR: REF SDSL$ENTRY,    ! Pointer to Source Directory Search
                                        !       List Entry
        SFCB_KIND,                      ! Indicate SFCB used for RMS file or
                                        !       Source Library file
        SRCFAB: REF $FAB_DECL,          ! FAB for RMS file
        SRCNAM: REF $NAM_DECL,          ! NAM for RMS file
        STATUS;                         ! RMS routine status



    ! In here there is a case that the file may not be available.
    ! If this file is not available, then signal the way
    ! out now, when the handler catches this signal, unwinds, so the normal
    ! operation will continue on -- for example, if stepping by inst., we do
    ! not want to see the message coming out for each step when the file
    ! is not available for the source.
    !
    IF .STEP_FLAG
    THEN
        BEGIN
        IF (.DBG$SRC_SFCB_PTR[SFCB$B_KIND] EQL SFCB$K_FILE_UNAVAIL) AND
```

```
                (.DBG$SRC_SFCB_PTR[SFCB$L_DSTPTR] EQL .FILDSTPTR)
            THEN
                SIGNAL(DBG$_FILEUNAL);

            END;


        ! Check to see if the given file is the Most Recently Used file.  If it
        ! is, simply return.  There is no need to reorder the SFCB list.
        !
        IF (.DBG$SRC_SFCB_PTR[SFCB$B_KIND] EQL SFCB$K_RMSFILE OR
            .DBG$SRC_SFCB_PTR[SFCB$B_KIND] EQL SFCB$K_LBRFILE) AND
            (.DBG$SRC_SFCB_PTR[SFCB$L_DSTPTR] EQL .FILDSTPTR)
        THEN
            BEGIN
            SFCBPTR[0] = .DBG$SRC_SFCB_PTR;
            RETURN;
            END;


        ! Start at the location pointed to by DBG$SRC_SFCB_PTR and circle thru the
        ! doubly linked list SFCB to see if the desired file is already on the
        ! list.  If it is already on the list, move it's Source File Control Block
        ! to the Most Recently Used position (which is pointed to by DBG$SRC_SFCB_PTR)
        ! and then return.
        !
        BAD_FILE = FALSE;
        LRUPTR = .DBG$SRC_SFCB_PTR[SFCB$L_BLINK];
        WHILE .LRUPTR NEQ .DBG$SRC_SFCB_PTR DO
            BEGIN
            IF (.LRUPTR[SFCB$B_KIND] NEQ SFCB$K_NOTUSED) AND
                (.LRUPTR[SFCB$L_DSTPTR] EQL .FILDSTPTR)
            THEN
                BEGIN


                ! The file already has a Source File Control Block.  Unlink it
                ! from the SFCB List.
                !
                BLINK = .LRUPTR[SFCB$L_BLINK];
                FLINK = .LRUPTR[SFCB$L_FLINK];
                BLINK[SFCB$L_FLINK] = .FLINK;
                FLINK[SFCB$L_BLINK] = .BLINK;


                ! Insert the SFCB at the Most Recently Used position.  Then return
                ! its address to the caller.
                !
                BLINK = .DBG$SRC_SFCB_PTR[SFCB$L_BLINK];
                LRUPTR[SFCB$L_FLINK] = .DBG$SRC_SFCB_PTR;
                LRUPTR[SFCB$L_BLINK] = .BLINK;
                BLINK[SFCB$L_FLINK] = .LRUPTR;
                DBG$SRC_SFCB_PTR[SFCB$L_BLINK] = .LRUPTR;
                DBG$SRC_SFCB_PTR = .LRUPTR;


                ! Check to see if this has been marked as unavailable file.
```

```
                !
                IF .DBG$SRC_SFCB_PTR[SFCB$B_KIND] EQL SFCB$K_FILE_UNAVAIL
                THEN
                    BEGIN
                    IF .STEP_FLAG
                    THEN
                        SIGNAL(DBG$_FILEUNAL)

                    ELSE
                        BEGIN
                        BAD_FILE = TRUE;
                        EXITLOOP;
                        END;

                    END

                ELSE
                    BEGIN
                    SFCBPTR[0] = .DBG$SRC_SFCB_PTR;
                    RETURN;
                    END;

                END;

            LRUPTR = .LRUPTR[SFCB$L_BLINK];
            END;

        ! We have circled the whole doubly linked list (the given file is not
        ! already on the list).  Advance the pointer to take the Least Recently
        ! Used SFCB and close the file using that control block (if any).
        !
        ! If we locate a bad file on the list, we want to force it to execute
        ! the code to get the bad status out to the user.  So this is the case
        ! we are by passing the IF.
        !
        IF NOT .BAD_FILE
        THEN
            BEGIN
            DBG$SRC_SFCB_PTR = .DBG$SRC_SFCB_PTR[SFCB$L_BLINK];
            IF .DBG$SRC_SFCB_PTR[SFCB$B_KIND] NEQ SFCB$K_NOTUSED
            THEN
                DBG$SRC_CLOSE_FILE(.DBG$SRC_SFCB_PTR);
            END;


        ! Determine whether the desired file is an RMS or LBR file.
        !
        LIBMOD_PTR =
            FILDSTPTR[DST$A_SRC_DF_FILENAME] + .FILDSTPTR[DST$B_SRC_DF_FILENAME];
        IF .LIBMOD_PTR[DST$B_SRC_DF_LIBMODNAME] EQL 0
        THEN
            SFCB_KIND = SFCB$K_RMSFILE
        ELSE
            SFCB_KIND = SFCB$K_LBRFILE;

```

```
! Initialize RMS source file control block.
!
$FAB_INIT(FAB = .DBG$SRC_SFCB_PTR[SFCB$L_FABPTR],
          FAC = <GET>,
          FOP = <NAM>,
          NAM = .DBG$SRC_SFCB_PTR[SFCB$L_NAMPTR],
          XAB = .DBG$SRC_SFCB_PTR[SFCB$L_XABDATPTR]);
$RAB_INIT(RAB = .DBG$SRC_SFCB_PTR[SFCB$L_RABPTR],
          FAB = .DBG$SRC_SFCB_PTR[SFCB$L_FABPTR],
          UBF = DBG$SRC_REC_BUF[0],
          USZ = MIN(SRCBUFSIZE,255));
$NAM_INIT(NAM = .DBG$SRC_SFCB_PTR[SFCB$L_NAMPTR],
          ESA = .DBG$SRC_SFCB_PTR[SFCB$L_NAMBUFFER],
          ESS = MIN(NAMBUFSIZE,255),
          RSA = .DBG$SRC_SFCB_PTR[SFCB$L_NAMBUFFER] + NAMBUFSIZE,
          RSS = MIN(NAMBUFSIZE,255));
$XABDAT_INIT(XAB = .DBG$SRC_SFCB_PTR[SFCB$L_XABDATPTR],
          NXT = .DBG$SRC_SFCB_PTR[SFCB$L_XABFHCPTR]);
$XABFHC_INIT(XAB = .DBG$SRC_SFCB_PTR[SFCB$L_XABFHCPTR]);
SRCFAB = .DBG$SRC_SFCB_PTR[SFCB$L_FABPTR];
SRCNAM = .DBG$SRC_SFCB_PTR[SFCB$L_NAMPTR];
PROPER_FLAG = 0;


! Set up default file name to be the same as that in the Source File
! Correlation DST Record except that we change the version number to '*'.
!
DEFAULT[0] = .FILDSTPTR[DST$B_SRC_DF_FILENAME];
CH$MOVE(.DEFAULT[0],FILDSTPTR[DST$A_SRC_DF_FILENAME],DEFAULT[1]);
DECR I FROM .DEFAULT[0] TO 1 DO
    BEGIN
    IF .DEFAULT[.I] EQL ';'
    THEN
        BEGIN
        DEFAULT[.I + 1] = '*';
        DEFAULT[0] = .I + 1;
        EXITLOOP;
        END;
    END;

SRCFAB[FAB$L_DNA] = DEFAULT[1];
SRCFAB[FAB$B_DNS] = .DEFAULT[0];


! Search the Source Directory Search List so we can open the given file in
! the proper directory.  Search all the Header Blocks for the Module RST
! Entry pointer for the current module.  If the desired module is not found,
! the Header Block with the zero Module RST Entry pointer is used.
!
SDSL_DIRPTR = 0;
SDSL_PTR = .DBG$SRC_DIR_LIST;
WHILE .SDSL_PTR NEQ 0 DO
    BEGIN
    IF .SDSL_PTR[SDSL$L_MODPTR] EQL .DBG$SRC_MODRSTPTR
    THEN
        BEGIN
        SDSL_DIRPTR = .SDSL_PTR[SDSL$L_LIST_PTR];
```

```
                IF .SDSL_DIRPTR EQL 0
                THEN
                    $DBG_ERROR('DBGSOURCE\DBG$SRC_OPEN_FILE 10');

                EXITLOOP;
                END;

            IF .SDSL_PTR[SDSL$L_MODPTR] EQL 0
            THEN
                BEGIN
                SDSL_DIRPTR = .SDSL_PTR[SDSL$L_LIST_PTR];
                IF .SDSL_DIRPTR EQL 0
                THEN
                    $DBG_ERROR('DBGSOURCE\DBG$SRC_OPEN_FILE 20');

                END;

            SDSL_PTR = .SDSL_PTR[SDSL$L_FLINK];
            END;

        ! If no Source Directory Search List entry is present, the file name from
        ! the Declare Source File DST command is used as is.  We thus open the
        ! file and if this succeeds, return to the caller.
        !
        IF .SDSL_DIRPTR EQL 0
        THEN
            BEGIN


            ! Set up the exact DST file name in the FAB.
            !
            SRCFAB[FAB$L_DNA] = 0;
            SRCFAB[FAB$B_DNS] = 0;
            SRCFAB[FAB$L_FNA] = FILDSTPTR[DST$A_SRC_DF_FILENAME];
            SRCFAB[FAB$B_FNS] = .FILDSTPTR[DST$B_SRC_DF_FILENAME];


            ! Do the $PARSE to initiate the $SEARCH loop.
            !
            STATUS = $PARSE(FAB = .SRCFAB);
            IF NOT .STATUS
            THEN
                BEGIN
                IF .STEP_FLAG
                THEN
                    BEGIN
                    DBG$SRC_SFCB_PTR[SFCB$B_KIND] = SFCB$K_FILE_UNAVAIL;
                    DBG$SRC_SFCB_PTR[SFCB$L_DSTPTR] = .FILDSTPTR;
                    DBG$SRC_SFCB_PTR[SFCB$L_CURRECNUM] = 1;
                    DBG$SRC_SFCB_PTR[SFCB$W_RFASPACING] = INIT_RFATBL_SPACING;
                    DBG$SRC_SFCB_PTR[SFCB$W_RFACURLEN] = 0;
                    END;

                SIGNAL(DBG$_UNAOPNSRC,2,.SRCNAM[NAM$B_ESL],.SRCNAM[NAM$L_ESA],.STATUS);

                END;
```

```

        ! Do $SEARCH to initate the NAME block.
        !
        STATUS = $SEARCH(FAB = .SRCFAB);
        IF NOT .STATUS
        THEN
            BEGIN
            IF .STEP_FLAG
            THEN
                BEGIN
                DBG$SRC_SFCB_PTR[SFCB$B_KIND] = SFCB$K_FILE_UNAVAIL;
                DBG$SRC_SFCB_PTR[SFCB$L_DSTPTR] = .FILDSTPTR;
                DBG$SRC_SFCB_PTR[SFCB$L_CURRECNUM] = 1;
                DBG$SRC_SFCB_PTR[SFCB$W_RFASPACING] = INIT_RFATBL_SPACING;
                DBG$SRC_SFCB_PTR[SFCB$W_RFACURLEN] = 0;
                END;

            SIGNAL(DBG$_UNAOPNSRC,2,.SRCNAM[NAM$B_ESL],.SRCNAM[NAM$L_ESA],.STATUS);

            END;


        ! Open the file and check the status.
        !
        STATUS = DBG$SRC_OPEN(.SFCB_KIND,.FILDSTPTR,PROPER_FLAG);
        IF NOT .STATUS
        THEN
            SIGNAL(DBG$_UNAOPNSRC,2,.SRCNAM[NAM$B_ESL],.SRCNAM[NAM$L_ESA],.STATUS);


        ! Check to see if we got the exact version of the desired file.  If not,
        ! issue the warning message.
        !
        IF (.PROPER_FLAG NEQ 0)
        THEN
            SIGNAL(DBG$_NOTORIGSRC,2,.SRCNAM[NAM$B_RSL],.SRCNAM[NAM$L_RSA]);


        ! The open succeeded.  Fill in the source file control block and return
        ! to the caller.
        !
        DBG$SRC_SFCB_PTR[SFCB$B_KIND] = .SFCB_KIND;
        DBG$SRC_SFCB_PTR[SFCB$L_DSTPTR] = .FILDSTPTR;
        DBG$SRC_SFCB_PTR[SFCB$L_CURRECNUM] = 1;
        DBG$SRC_SFCB_PTR[SFCB$W_RFASPACING] = INIT_RFATBL_SPACING;
        DBG$SRC_SFCB_PTR[SFCB$W_RFACURLEN] = 0;
        SFCBPTR[0] = .DBG$SRC_SFCB_PTR;
        RETURN;
        END;


    ! The found Header Block pointer then points to a list of directories.
    ! Search this list to find the desired file in one of the directories on
    ! that list.
    !
    FIRST_TIME = TRUE;
```

```
WHILE .SDSL_DIRPTR NEQ 0 DO
    BEGIN


    ! Make the current directory name the primary file name in the FAB.
    ! Then do the $PARSE to initiate the $SEARCH loop.
    !
    SRCFAB[FAB$L_FNA] = SDSL_DIRPTR[SDSL$A_ENT_DIRNAME];
    SRCFAB[FAB$B_FNS] = .SDSL_DIRPTR[SDSL$B_ENT_DIRLEN];
    STATUS = $PARSE(FAB = .SRCFAB);
    IF NOT ((.STATUS EQL RMS$_DNF) OR (.STATUS))
    THEN
        SIGNAL(DBG$_UNAOPNSRC,2,.SRCNAM[NAM$B_ESL],.SRCNAM[NAM$L_ESA],.STATUS);

    IF .STATUS
    THEN
        BEGIN


        ! Do wild-card processing by going through a $SEARCH loop.  For
        ! each file found by $SEARCH, we see if it is the file we want.
        !
        WHILE TRUE DO
            BEGIN


            ! Get the next file from $SEARCH.
            !
            STATUS = $SEARCH(FAB = .SRCFAB);
            IF (.STATUS EQL RMS$_FNF) OR (.STATUS EQL RMS$_NMF)
            THEN
                EXITLOOP;
            IF NOT .STATUS
            THEN
                SIGNAL(DBG$_UNAOPNSRC,2,.SRCNAM[NAM$B_ESL],.SRCNAM[NAM$L_ESA],
                        .STATUS);


            ! Try to open that file and see if it is the exact file we
            ! want.  If we find the exact file we want, exit the search
            ! loop.
            !
            STATUS = DBG$SRC_OPEN(.SFCB_KIND,.FILDSTPTR,PROPER_FLAG);
            IF NOT .STATUS
            THEN
                SIGNAL(DBG$_UNAOPNSRC,2,.SRCNAM[NAM$B_ESL],.SRCNAM[NAM$L_ESA],
                        .STATUS);

            IF .PROPER_FLAG EQL 0 THEN EXITLOOP;


            ! The first time we find any file with the right name, we remember
            ! that file's name even if it file characteristics are wrong.
            !
            IF .FIRST_TIME
            THEN
                BEGIN
```

```
                    FIRST_FILE[0] = .SRCFAB[FAB$B_FNS];
                    CH$MOVE(.FIRST_FILE[0],.SRCFAB[FAB$L_FNA],FIRST_FILE[1]);
                    FIRST_TIME = FALSE;
                    END;


                ! It is not the right file - close it and loop for another file.
                !
                DBG$SRC_SFCB_PTR[SFCB$B_KIND] = .SFCB_KIND;
                DBG$SRC_CLOSE_FILE(.DBG$SRC_SFCB_PTR);

                END;                ! End of $SEARCH WHILE loop.

            END;                    ! End of if on good $PARSE status.


        ! If we have not yot found the desired file, get the next Source
        ! Directory Search List Entry and loop to try it.
        !
        IF .STATUS AND (.PROPER_FLAG EQL 0) THEN EXITLOOP;
        SDSL_DIRPTR = .SDSL_DIRPTR[SDSL$L_ENT_FLINK];
        END;


    ! If no proper file is found, the first file for the first directory
    ! on the list is used, so try to open it again.
    !
    IF (.PROPER_FLAG NEQ 0) AND (NOT .FIRST_TIME)
    THEN
        BEGIN
        SRCFAB[FAB$L_FNA] = FIRST_FILE[1];
        SRCFAB[FAB$B_FNS] = .FIRST_FILE[0];
        STATUS = $PARSE(FAB = .SRCFAB);
        IF NOT .STATUS THEN SIGNAL(.STATUS);
        STATUS = $SEARCH(FAB = .SRCFAB);
        IF NOT .STATUS THEN SIGNAL(.STATUS);
        STATUS = DBG$SRC_OPEN(.SFCB_KIND,.FILDSTPTR,PROPER_FLAG);
        IF NOT .STATUS THEN SIGNAL(.STATUS);
        END;


    ! If the status returned from RMS is not 'no more files' or good status,
    ! signal the bad status.
    !
    IF NOT ((.STATUS EQL RMS$_NMF) OR (.STATUS))
    THEN
        BEGIN
        IF .STEP_FLAG
        THEN
            BEGIN
            DBG$SRC_SFCB_PTR[SFCB$B_KIND] = SFCB$K_FILE_UNAVAIL;
            DBG$SRC_SFCB_PTR[SFCB$L_DSTPTR] = .FILDSTPTR;
            DBG$SRC_SFCB_PTR[SFCB$L_CURRECNUM] = 1;
            END;

        SIGNAL(DBG$_UNAOPNSRC,2,.SRCNAM[NAM$B_ESL],.SRCNAM[NAM$L_ESA],.STATUS);
        END;
```

```
2
2
2          ! Issue the warning message, if the exact file is not found.
2          !
3          IF (.PROPER_FLAG NEQ 0)
2          THEN
2              SIGNAL(DBG$_NOTORIGSRC,2,.SRCNAM[NAM$B_RSL],.SRCNAM[NAM$L_RSA]);
2
2
2          ! We have now successfully opened a source file in the search directory.
2          ! Fill in the Source File Control Block and return.
2          !
2          DBG$SRC_SFCB_PTR[SFCB$B_KIND] = .SFCB_KIND;
2          DBG$SRC_SFCB_PTR[SFCB$L_DSTPTR] = .FI[DSTPTR;
2          DBG$SRC_SFCB_PTR[SFCB$L_CURRECNUM] = 1;
2          SFCBPTR[0] = .DBG$SRC_SFCB_PTR;
2          RETURN;
2
1          END;


                                                              .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

24 47 42 44 5C 45 43 52 55 4F 53 47 42 44 1E 000FF P.AAN:   .ASCII  <30>\DBGSOURCE\<92>\DBG$SRC_OPEN_FILE \   ;
   20 45 4C 49 46 5F 4E 45 50 4F 5F 43 52 53 0010E                                                       ;
                                           30 31 0011C      .ASCII  \10\                                 ;
24 47 42 44 5C 45 43 52 55 4F 53 47 42 44 1E 0011E P.AAO:   .ASCII  <30>\DBGSOURCE\<92>\DBG$SRC_OPEN_FILE \   ;
   20 45 4C 49 46 5F 4E 45 50 4F 5F 43 52 53 0012D                                                       ;
                                           30 32 0013B      .ASCII  \20\                                 ;

                                                              .EXTRN  SYS$PARSE, SYS$SEARCH

                                                              .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                  0FFC 00000 DBG$SRC_OPEN_FILE:
                                                              .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11   ; 1482
                   5E       FDF4  CE 9E 00002                 MOVAB   -524(SP), SP
                              0C  AC DD 00007                 PUSHL   STEP_FLAG                              ; 1592
                   21             6E E9 0000A                 BLBC    (SP), 1$
                   50 00000000'   EF D0 0000D                 MOVL    DBG$SRC_SFCB_PTR, R0                   ; 1595
                   04         08  A0 91 00014                 CMPB    8(R0), #4                              ;
                                  14 12 00018                 BNEQ    1$                                     ;
              04   AC         14  A0 D1 0001A                 CMPL    20(R0), FILDSTPTR                      ; 1596
                                  0D 12 0001F                 BNEQ    1$                                     ;
                        00028D10  8F DD 00021                 PUSHL   #167184                                ; 1598
         00000000G 00             01 FB 00027                 CALLS   #1, LIB$SIGNAL                         ;
                   50 00000000'   EF D0 0002E 1$:             MOVL    DBG$SRC_SFCB_PTR, R0                   ; 1606
                   02         08  A0 91 00035                 CMPB    8(R0), #2                              ;
                                  06 13 00039                 BEQL    2$                                     ;
                   03         08  A0 91 0003B                 CMPB    8(R0), #3                              ; 1607
                                  0A 12 0003F                 BNEQ    4$                                     ;
              04   AC         14  A0 D1 00041 2$:             CMPL    20(R0), FILDSTPTR                      ; 1608
                                  03 12 00046                 BNEQ    4$                                     ;
                            0491  31 00048 3$:                BRW     43$                                    ;
                                  55 D4 0004B 4$:             CLRL    BAD_FILE                               ; 1622
                   52         04  A0 D0 0004D                 MOVL    4(R0), LRUPTR                          ; 1623
```

```
                          50 00000000'  EF  D0 00051 5$:    MOVL    DBG$SRC_SFCB_PTR, R0                     ; 1624
                          50            52  D1 00058        CMPL    LRUPTR, R0
                                        5A  13 0005B        BEQL    8$
                          01        08  A2  91 0005D        CMPB    8(LRUPTR), #1                            ; 1626
                                        4E  13 00061        BEQL    7$
                   04     AC        14  A2  D1 00063        CMPL    20(LRUPTR), FILDSTPTR                    ; 1627
                                        47  12 00068        BNEQ    7$
                          53            62  7D 0006A        MOVQ    (LRUPTR), FLINK                          ; 1636
                          64            53  D0 0006D        MOVL    FLINK, (BLINK)                           ; 1637
                   04     A3            54  D0 00070        MOVL    BLINK, 4(FLINK)                          ; 1638
                          54        04  A0  D0 00074        MOVL    4(R0), BLINK                             ; 1644
                          62            50  D0 00078        MOVL    R0, (LRUPTR)                             ; 1645
                   04     A2            54  D0 0007B        MOVL    BLINK, 4(LRUPTR)                         ; 1646
                          64            52  D0 0007F        MOVL    LRUPTR, (BLINK)                          ; 1647
                   04     A0            52  D0 00082        MOVL    LRUPTR, 4(R0)                            ; 1648
            00000000'     EF            52  D0 00086        MOVL    LRUPTR, DBG$SRC_SFCB_PTR                 ; 1649
                          50 00000000'  EF  D0 0008D        MOVL    DBG$SRC_SFCB_PTR, R0                     ; 1654
                          04        08  A0  91 00094        CMPB    8(R0), #4
                                        AE  12 00098        BNEQ    3$
                          0F            6E  E9 0009A        BLBC    (SP), 6$                                 ; 1657
                             00028D10   8F  DD 0009D        PUSHL   #167184                                  ; 1659
            00000000G     00            01  FB 000A3        CALLS   #1, LIB$SIGNAL
                                        05  11 000AA        BRB     7$
                          55            01  D0 000AC 6$:    MOVL    #1, BAD_FILE                             ; 1663
                                        06  11 000AF        BRB     8$                                       ; 1662
                          52        04  A2  D0 000B1 7$:    MOVL    4(LRUPTR), LRUPTR                        ; 1677
                                        9A  11 000B5        BRB     5$                                       ; 1624
                          23            55  E8 000B7 8$:    BLBS    BAD_FILE, 9$                             ; 1689
                          50 00000000'  EF  D0 000BA        MOVL    DBG$SRC_SFCB_PTR, R0                     ; 1692
            00000000'     EF        04  A0  D0 000C1        MOVL    4(R0), DBG$SRC_SFCB_PTR
                          50 00000000'  EF  D0 000C9        MOVL    DBG$SRC_SFCB_PTR, R0                     ; 1693
                          01        08  A0  91 000D0        CMPB    8(R0), #1
                                        07  13 000D4        BEQL    9$
                                        50  DD 000D6        PUSHL   R0                                       ; 1695
                 F774     CF            01  FB 000D8        CALLS   #1, DBG$SRC_CLOSE_FILE
                          5B        04  AC  D0 000DD 9$:    MOVL    FILDSTPTR, R11                           ; 1702
                   04     AE        14  AB  9A 000E1        MOVZBL  20(R11), 4(SP)
           51      04     AE            15  C1 000E6        ADDL3   #21, 4(SP), R1
           50             5B            51  C1 000EB        ADDL3   R1, R11, LIBMOD_PTR
                                        60  95 000EF        TSTB    (LIBMOD_PTR)                             ; 1703
                                        06  12 000F1        BNEQ    10$
                   08     AE            02  D0 000F3        MOVL    #2, SFCB_KIND                            ; 1705
                                        04  11 000F7        BRB     11$
                   08     AE            03  D0 000F9 10$:   MOVL    #3, SFCB_KIND                            ; 1707
                          58 00000000'  EF  D0 000FD 11$:   MOVL    DBG$SRC_SFCB_PTR, R8                     ; 1716
                          59        18  A8  D0 00104        MOVL    24(R8), R9
0050   8F  00             6E            00  2C 00108        MOVC5   #0, (SP), #0, #80, (R9)
                                        69     0010F
                          69      5003  8F  B0 00110        MOVW    #20483, (R9)
                   04     A9  01000000  8F  D0 00115        MOVL    #16777216, 4(R9)
                   16     A9            02  90 0011D        MOVB    #2, 22(R9)
                   1F     A9            02  90 00121        MOVB    #2, 31(R9)
                          5A        24  A8  D0 00125        MOVL    36(R8), R10
                   24     A9            5A  D0 00129        MOVL    R10, 36(R9)
                          56        1C  A8  7D 0012D        MOVQ    28(R8), R6                               ; 1720
                   28     A9            57  D0 00131        MOVL    R7, 40(R9)                               ; 1716
0044   8F  00             6E            00  2C 00135        MOVC5   #0, (SP), #0, #68, (R6)                  ; 1720
```

```
                                     66       0013C
              66       4401          8F   B0  0013D        MOVW    #17409, (R6)
           20 A6         FF          8F   9B  00142        MOVZBW  #255, 32(R6)
           24 A6  00000000'          EF   9E  00147        MOVAB   DBG$SRC_REC_BUF, 36(R6)
           3C A6                     59   D0  0014F        MOVL    R9, 60(R6)
 0060 8F      00      6E             00   2C  00153        MOVC5   #0, (SP), #0, #96, (R7)                : 1725
                                     67       0015A
              67       6002          8F   B0  0015B        MOVW    #24578, (R7)
           02 A7                     01   8E  00160        MNEGB   #1, 2(R7)
     04 A7 2C A8  00000100           8F   C1  00164        ADDL3   #256, 44(R8), 4(R7)
           0A A7                     01   8E  0016E        MNEGB   #1, 10(R7)
           0C A7         2C          A8   D0  00172        MOVL    44(R8), 12(R7)
      2C      00      6E             00   2C  00177        MOVC5   #0, (SP), #0, #44, (R10)               : 1727
                                     6A       0017C
              6A       2C12          8F   B0  0017D        MOVW    #11282, (R10)
           04 AA         28          A8   D0  00182        MOVL    40(R8), 4(R10)
              58         28          A8   D0  00187        MOVL    40(R8), R8                             : 1728
      2C      00      6E             00   2C  0018B        MOVC5   #0, (SP), #0, #44, (R8)
                                     68       00190
              68       2C1D          8F   B0  00191        MOVW    #11293, (R8)
              5A                     57   D0  00196        MOVL    R7, SRCNAM                             : 1730
                         0C          AE   D4  00199        CLRL    PROPER_FLAG                            : 1731
         FF00 CD         04          AE   90  0019C        MOVB    4(SP), DEFAULT                         : 1737
              50       FF00          CD   9A  001A2        MOVZBL  DEFAULT, R0                            : 1738
 FF01 CD   15 AB                     50   28  001A7        MOVC3   R0, 21(R11), DEFAULT+1
              50       FF00          CD   9A  001AE        MOVZBL  DEFAULT, I                             : 1739
                                     50   D6  001B3        INCL    I
                                     16   11  001B5        BRB     13$
              3B       FF00 CD40     91       001B7  12$:  CMPB    DEFAULT[I], #59                        : 1741
                                     0E   12  001BD        BNEQ    13$
         FF01 CD40                   2A   90  001BF        MOVB    #42, DEFAULT+1[I]                      : 1744
 FF00 CD      50                     01   81  001C5        ADDB3   #1, I, DEFAULT                         : 1745
                                     03   11  001CB        BRB     14$                                    : 1743
              E7                     50   F5  001CD  13$:  SOBGTR  I, 12$                                 : 1739
           30 A9       FF01          CD   9E  001D0  14$:  MOVAB   DEFAULT+1, 48(SRCFAB)                  : 1750
           35 A9       FF00          CD   90  001D6        MOVB    DEFAULT, 53(SRCFAB)                    : 1751
                                     58   D4  001DC        CLRL    SDSL_DIRPTR                            : 1759
              52  00000000'          EF   D0  001DE        MOVL    DBG$SRC_DIR_LIST, SDSL_PTR             : 1760
                                     4C   13  001E5  15$:  BEQL    18$                                    : 1761
     00000000' EF            08      A2   D1  001E7        CMPL    8(SDSL_PTR), DBG$SRC_MODRSTPTR         : 1763
                                     1D   12  001EF        BNEQ    16$
              58             04      A2   D0  001F1        MOVL    4(SDSL_PTR), SDSL_DIRPTR               : 1766
                                     3C   12  001F5        BNEQ    18$                                    : 1767
                  00000000'          EF   9F  001F7        PUSHAB  P.AAN                                  : 1769
                                     01   DD  001FD        PUSHL   #1
                  00028362           8F   DD  001FF        PUSHL   #164706
     00000000G 00                    03   FB  00205        CALLS   #3, LIB$SIGNAL
                                     25   11  0020C        BRB     18$                                    : 1765
                             08      A2   D5  0020E  16$:  TSTL    8(SDSL_PTR)                            : 1774
                                     1B   12  00211        BNEQ    17$
              58             04      A2   D0  00213        MOVL    4(SDSL_PTR), SDSL_DIRPTR               : 1777
                                     15   12  00217        BNEQ    17$                                    : 1778
                  00000000'          EF   9F  00219        PUSHAB  P.AAO                                  : 1780
                                     01   DD  0021F        PUSHL   #1
                  00028362           8F   DD  00221        PUSHL   #164706
     00000000G 00                    03   FB  00227        CALLS   #3, LIB$SIGNAL
              52                     62   D0  0022E  17$:  MOVL    (SDSL_PTR), SDSL_PTR                   : 1784
```

```
                                  B2  11 00231       BRB      15$                                    ; 1761
                                  58  D5 00233 18$:  TSTL     SDSL_DIRPTR                            ; 1792
                                  03  13 00235       BEQL     19$                                    ;
                                00F0  31 00237       BRW      26$                                    ;
                          30      A9  D4 0023A 19$:  CLRL     48(SRCFAB)                             ; 1799
         2C      A9       15      AB  9E 0023D       MOVAB    21(R11), 44(SRCFAB)                    ; 1801
         34      A9       04      AE  9B 00242       MOVZBW   4(SP), 52(SRCFAB)                      ; 1802
                                  59  DD 00247       PUSHL    SRCFAB                                 ; 1807
  00000000G      00               01  FB 00249       CALLS    #1, SYS$PARSE                          ;
                 57               50  D0 00250       MOVL     R0, STATUS                             ;
                 32               57  E8 00253       BLBS     STATUS, 21$                            ; 1808
                 17               6E  E9 00256       BLBC     (SP), 20$                              ; 1811
                 50  00000000'    EF  D0 00259       MOVL     DBG$SRC_SFCB_PTR, R0                   ; 1814
         08      A0               04  90 00260       MOVB     #4, 8(R0)                              ;
         14      A0               5B  D0 00264       MOVL     R11, 20(R0)                            ; 1815
         30      A0               01  D0 00268       MOVL     #1, 48(R0)                             ; 1816
         0C      A0               05  D0 0026C       MOVL     #5, 12(R0)                             ; 1817
                                  57  DD 00270 20$:  PUSHL    STATUS                                 ; 1821
                          0C      AA  DD 00272       PUSHL    12(SRCNAM)                             ;
                 7E       0B      AA  9A 00275       MOVZBL   11(SRCNAM), -(SP)                      ;
                                  02  DD 00279       PUSHL    #2                                     ;
                    00028CF8      8F  DD 0027B       PUSHL    #167160                                ;
  00000000G      00               05  FB 00281       CALLS    #5, LIB$SIGNAL                         ;
                                  59  DD 00288 21$:  PUSHL    SRCFAB                                 ; 1828
  00000000G      00               01  FB 0028A       CALLS    #1, SYS$SEARCH                         ;
                 57               50  D0 00291       MOVL     R0, STATUS                             ;
                 32               57  E8 00294       BLBS     STATUS, 23$                            ; 1829
                 17               6E  E9 00297       BLBC     (SP), 22$                              ; 1832
                 50  00000000'    EF  D0 0029A       MOVL     DBG$SRC_SFCB_PTR, R0                   ; 1835
         08      A0               04  90 002A1       MOVB     #4, 8(R0)                              ;
         14      A0               5B  D0 002A5       MOVL     R11, 20(R0)                            ; 1836
         30      A0               01  D0 002A9       MOVL     #1, 48(R0)                             ; 1837
         0C      A0               05  D0 002AD       MOVL     #5, 12(R0)                             ; 1838
                                  57  DD 002B1 22$:  PUSHL    STATUS                                 ; 1842
                          0C      AA  DD 002B3       PUSHL    12(SRCNAM)                             ;
                 7E       0B      AA  9A 002B6       MOVZBL   11(SRCNAM), -(SP)                      ;
                                  02  DD 002BA       PUSHL    #2                                     ;
                    00028CF8      8F  DD 002BC       PUSHL    #167160                                ;
  00000000G      00               05  FB 002C2       CALLS    #5, LIB$SIGNAL                         ;
                          0C      AE  9F 002C9 23$:  PUSHAB   PROPER_FLAG                            ; 1849
                                  5B  DD 002CC       PUSHL    R11                                    ;
                          10      AE  DD 002CE       PUSHL    SFCB_KIND                              ;
       FBE6      CF               03  FB 002D1       CALLS    #3, DBG$SRC_OPEN                       ;
                 57               50  D0 002D6       MOVL     R0, STATUS                             ;
                 18               57  E8 002D9       BLBS     STATUS, 24$                            ; 1850
                                  57  DD 002DC       PUSHL    STATUS                                 ; 1852
                          0C      AA  DD 002DE       PUSHL    12(SRCNAM)                             ;
                 7E       0B      AA  9A 002E1       MOVZBL   11(SRCNAM), -(SP)                      ;
                                  02  DD 002E5       PUSHL    #2                                     ;
                    00028CF8      8F  DD 002E7       PUSHL    #167160                                ;
  00000000G      00               05  FB 002ED       CALLS    #5, LIB$SIGNAL                         ;
                          0C      AE  D5 002F4 24$:  TSTL     PROPER_FLAG                            ; 1858
                                  16  13 002F7       BEQL     25$                                    ;
                          04      AA  DD 002F9       PUSHL    4(SRCNAM)                              ; 1860
                 7E       03      AA  9A 002FC       MOVZBL   3(SRCNAM), -(SP)                       ;
                                  02  DD 00300       PUSHL    #2                                     ;
                    00028673      8F  DD 00302       PUSHL    #165491                                ;
```

```
00000000G  00                04  FB 00308          CALLS   #4, LIB$SIGNAL                              ;
           50  00000000'     EF  D0 0030F  25$:    MOVL    DBG$SRC_SFCB_PTR, R0                        ; 1866
       08  A0          08    AE  90 00316          MOVB    SFCB_KIND, 8(R0)                            ;
       14  A0                5B  D0 0031B          MOVL    R11, 20(R0)                                 ; 1867
       30  A0                01  D0 0031F          MOVL    #1, 48(R0)                                  ; 1868
       0C  A0                05  D0 00323          MOVL    #5, 12(R0)                                  ; 1869
                           01B2  31 00327          BRW     43$                                         ; 1871
           56                01  D0 0032A  26$:    MOVL    #1, FIRST_TIME                              ; 1880
                             58  D5 0032D  27$:    TSTL    SDSL_DIRPTR                                 ; 1881
                             03  12 0032F          BNEQ    28$                                         ;
                           00E1  31 00331          BRW     36$                                         ;
       2C  A9          05    A8  9E 00334  28$:    MOVAB   5(R8), 44(SRCFAB)                           ; 1888
       34  A9          04    A8  90 00339          MOVB    4(SDSL_DIRPTR), 52(SRCFAB)                  ; 1889
                             59  DD 0033E          PUSHL   SRCFAB                                      ; 1890
00000000G  00                01  FB 00340          CALLS   #1, SYS$PARSE                               ;
           57                50  D0 00347          MOVL    R0, STATUS                                  ;
0001C04A   8F                57  D1 0034A          CMPL    STATUS, #114762                             ; 1891
                             1B  13 00351          BEQL    29$                                         ;
           1E                57  E8 00353          BLBS    STATUS, 30$                                 ;
                             57  DD 00356          PUSHL   STATUS                                      ; 1893
                       0C    AA  DD 00358          PUSHL   12(SRCNAM)                                  ;
           7E          0B    AA  9A 0035B          MOVZBL  11(SRCNAM), -(SP)                           ;
                             02  DD 0035F          PUSHL   #2                                          ;
                00028CF8     8F  DD 00361          PUSHL   #167160                                     ;
00000000G  00                05  FB 00367          CALLS   #5, LIB$SIGNAL                              ;
           03                57  E8 0036E  29$:    BLBS    STATUS, 30$                                 ; 1895
                           009B  31 00371          BRW     35$                                         ;
                             59  DD 00374  30$:    PUSHL   SRCFAB                                      ; 1909
00000000G  00                01  FB 00376          CALLS   #1, SYS$SEARCH                              ;
           57                50  D0 0037D          MOVL    R0, STATUS                                  ;
00018292   8F                57  D1 00380          CMPL    STATUS, #98962                              ; 1910
                             7E  13 00387          BEQL    34$                                         ;
000182CA   8F                57  D1 00389          CMPL    STATUS, #99018                              ;
                             75  13 00390          BEQL    34$                                         ;
           18                57  E8 00392          BLBS    STATUS, 31$                                 ; 1913
                             57  DD 00395          PUSHL   STATUS                                      ; 1916
                       0C    AA  DD 00397          PUSHL   12(SRCNAM)                                  ; 1915
           7E          0B    AA  9A 0039A          MOVZBL  11(SRCNAM), -(SP)                           ;
                             02  DD 0039E          PUSHL   #2                                          ;
                00028CF8     8F  DD 003A0          PUSHL   #167160                                     ;
00000000G  00                05  FB 003A6          CALLS   #5, LIB$SIGNAL                              ;
                       0C    AE  9F 003AD  31$:    PUSHAB  PROPER_FLAG                                 ; 1923
                             5B  DD 003B0          PUSHL   R11                                         ;
                       10    AE  DD 003B2          PUSHL   SFCB_KIND                                   ;
     FB02  CF                03  FB 003B5          CALLS   #3, DBG$SRC_OPEN                            ;
           57                50  D0 003BA          MOVL    R0, STATUS                                  ;
           18                57  E8 003BD          BLBS    STATUS, 32$                                 ; 1924
                             57  DD 003C0          PUSHL   STATUS                                      ; 1927
                       0C    AA  DD 003C2          PUSHL   12(SRCNAM)                                  ; 1926
           7E          0B    AA  9A 003C5          MOVZBL  11(SRCNAM), -(SP)                           ;
                             02  DD 003C9          PUSHL   #2                                          ;
                00028CF8     8F  DD 003CB          PUSHL   #167160                                     ;
00000000G  00                05  FB 003D1          CALLS   #5, LIB$SIGNAL                              ;
                       0C    AE  D5 003D8  32$:    TSTL    PROPER_FLAG                                 ; 1929
                             2A  13 003DB          BEQL    34$                                         ;
           11                56  E9 003DD          BLBC    FIRST_TIME, 33$                             ; 1935
       10  AE          34    A9  90 003E0          MOVB    52(SRCFAB), FIRST_FILE                      ; 1938
```

```
                    50        10  AE  9A 003E5          MOVZBL  FIRST_FILE, R0                          ; 1939
  11  AE      2C    B9                50  28 003E9          MOVC3   R0, @44(SRCFAB), FIRST_FILE+1
                                      56  D4 003EF          CLRL    FIRST_TIME                              ; 1940
                    50  00000000'     EF  D0 003F1  33$:    MOVL    DBG$SRC_SFCB_PTR, R0                    ; 1946
              08    A0        08      AE  90 003F8          MOVB    SFCB_KIND, 8(R0)
                                      50  DD 003FD          PUSHL   R0                                      ; 1947
            F44D    CF                01  FB 003FF          CALLS   #1, DBG$SRC_CLOSE_FILE
                                 FF6D 31 00404              BRW     30$                                     ; 1903
                    05                57  E9 00407  34$:    BLBC    STATUS, 35$                             ; 1957
                              0C      AE  D5 0040A          TSTL    PROPER_FLAG
                                      06  13 0040D          BEQL    36$
                    58                68  D0 0040F  35$:    MOVL    (SDSL_DIRPTR), SDSL_DIRPTR              ; 1958
                                 FF18 31 00412              BRW     27$                                     ; 1881
                              0C      AE  D5 00415  36$:    TSTL    PROPER_FLAG                             ; 1965
                                      59  13 00418          BEQL    39$
                    56                56  E8 0041A          BLBS    FIRST_TIME, 39$
              2C    A9        11      AE  9E 0041D          MOVAB   FIRST_FILE+1, 44(SRCFAB)                ; 1968
              34    A9        10      AE  90 00422          MOVB    FIRST_FILE, 52(SRCFAB)                  ; 1969
                                      59  DD 00427          PUSHL   SRCFAB                                  ; 1970
       00000000G    00                01  FB 00429          CALLS   #1, SYS$PARSE
                    57                50  D0 00430          MOVL    R0, STATUS
                    09                57  E8 00433          BLBS    STATUS, 37$                             ; 1971
                                      57  DD 00436          PUSHL   STATUS
       00000000G    00                01  FB 00438          CALLS   #1, LIB$SIGNAL
                                      59  DD 0043F  37$:    PUSHL   SRCFAB                                  ; 1972
       00000000G    00                01  FB 00441          CALLS   #1, SYS$SEARCH
                    57                50  D0 00448          MOVL    R0, STATUS
                    09                57  E8 0044B          BLBS    STATUS, 38$                             ; 1973
                                      57  DD 0044E          PUSHL   STATUS
       00000000G    00                01  FB 00450          CALLS   #1, LIB$SIGNAL
                              0C      AE  9F 00457  38$:    PUSHAB  PROPER_FLAG                             ; 1974
                                      5B  DD 0045A          PUSHL   R11
                              10      AE  DD 0045C          PUSHL   SFCB_KIND
            FA58    CF                03  FB 0045F          CALLS   #3, DBG$SRC_OPEN
                    57                50  D0 00464          MOVL    R0, STATUS
                    09                57  E8 00467          BLBS    STATUS, 39$                             ; 1975
                                      57  DD 0046A          PUSHL   STATUS
       00000000G    00                01  FB 0046C          CALLS   #1, LIB$SIGNAL
       000182CA     8F                57  D1 00473  39$:    CMPL    STATUS, #99018                          ; 1982
                                      31  13 0047A          BEQL    41$
                    2E                57  E8 0047C          BLBS    STATUS, 41$
                    13                6E  E9 0047F          BLBC    (SP), 40$                               ; 1985
                    50  00000000'     EF  D0 00482          MOVL    DBG$SRC_SFCB_PTR, R0                    ; 1988
              08    A0                04  90 00489          MOVB    #4, 8(R0)
              14    A0                5B  D0 0048D          MOVL    R11, 20(R0)                             ; 1989
              30    A0                01  D0 00491          MOVL    #1, 48(R0)                              ; 1990
                                      57  DD 00495  40$:    PUSHL   STATUS                                  ; 1993
                              0C      AA  DD 00497          PUSHL   12(SRCNAM)
                    7E        0B      AA  9A 0049A          MOVZBL  11(SRCNAM), -(SP)
                                      02  DD 0049E          PUSHL   #2
                         00028CF8     8F  DD 004A0          PUSHL   #167160
       00000000G    00                05  FB 004A6          CALLS   #5, LIB$SIGNAL
                              0C      AE  D5 004AD  41$:    TSTL    PROPER_FLAG                             ; 1999
                                      16  13 004B0          BEQL    42$
                              04      AA  DD 004B2          PUSHL   4(SRCNAM)                               ; 2001
                    7E        03      AA  9A 004B5          MOVZBL  3(SRCNAM), -(SP)
                                      02  DD 004B9          PUSHL   #2
```

```
                  00028673   8F  DD 004BB          PUSHL   #165491                        ;
        00000000G  00        04  FB 004C1          CALLS   #4, LIB$SIGNAL                 ;
                  50 00000000'  EF  D0 004C8 42$:  MOVL    DBG$SRC_SFCB_PTR, R0           ; 2007
           08  A0       08   AE  90 004CF          MOVB    SFCB_KIND, 8(R0)               ;
           14  A0            5B  D0 004D4          MOVL    R11, 20(R0)                    ; 2008
           30  A0            01  D0 004D8          MOVL    #1, 48(R0)                     ; 2009
           08  BC            50  D0 004DC 43$:     MOVL    R0, @SFCBPTR                   ; 2010
                                 04 004E0          RET                                    ; 2013

; Routine Size:  1249 bytes,    Routine Base:  DBG$CODE + 082E


; 1889          2014  1
```

```
ROUTINE DBG$SRC_OUTPUT_LINE(LINE_NUM, STMT_NUM, BUFPTR, BUFLEN): NOVALUE =
!
! FUNCTION
!       This routine formats and outputs one source line.  The actual text of
!       the source line is passed in as a parameter.  The line's line number and
!       statement number are also passed in.  This routine then formats the line
!       number and statement number in front of the line's text and outputs the
!       line.  The statement number is omitted if it is zero (as it will be for
!       most languages).  The output for lines 12, 123, 1234, 123456, 1234567,
!       12345.6, and 123.4 would look as follows:
!
!                      12: TEXT OF LINE 12
!                     123: TEXT OF LINE 123
!                    1234: TEXT OF LINE 1234
!                  123456: TEXT OF LINE 123456
!                  1234567: TEXT OF LINE 1234567
!                  12345.6: TEXT OF LINE 12345, STATEMENT 6
!                   123.4: TEXT OF LINE 123, STATEMENT 4
!
!       Note that all lines line up unless the line number and statement number
!       require more that six characters.
!
!       This routine also folds long source lines when they would make the out-
!       put line exceed terminal width in length.  The following format is used:
!
!                     123: TEXT OF LINE 123, FIRST PART
!                      -: TEXT OF LINE 123, SECOND PART
!                      -: TEXT OF LINE 123, LAST PART
!
!       The string "-: " thus serves as a continuation marker.  Note that this
!       routine must expand tabs to spaces in the source text.
!
!       User can use the SET MARGIN Command to set the display window size of
!       the source text.  For example, if user gives SET MARGIN 10:40, user
!       will see the source text from the 10th character to the 40th character
!       displayed on the terminal.  However, the window size is adjused by the
!       the length of the source record (after the expension of TABs), and the
!       terminal width set by the SET TERM/WIDTH = n Command.
!
!       The final output lines are printed using DBG$PRINT in the normal way.
!
! INPUTS
!       LINE_NUM - The line number of the source line to be printed.
!
!       STMT_NUM - The statement number of the source line to be printed.
!                  If there is no statement number, STMT_NUM must be zero.
!
!       BUFPTR   - A pointer to a buffer which contains the ASCII text of the
!                  source line to be printed.
!
!       BUFLEN   - The length in characters of the source line pointed to by
!                  BUFPTR.
!
! OUTPUTS
!       NONE
!
```

```
2      BEGIN
2
2      MAP
2          BUFPTR: REF VECTOR[,BYTE];          ! Pointer to the ASCII image to print
2
2      BIND
2          SPACES = UPLIT BYTE (%ASCIC '        '): VECTOR[,BYTE];
2                                              ! A vector of 8 spaces
2
2      LOCAL
2          FST_LINE_FLAG,                      ! Flag set to indicate this output line
2                                              !         has a line number
2          BGNPTR: REF VECTOR[,BYTE],          ! Pointer to the begin search tab position
2                                              !         in source text buffer
2          FORMFEED_PTR: REF VECTOR[,BYTE],    ! Pointer to the formfeed character in
2                                              !         output line
2          GAP,                                ! Distance between the begin pointer and
2                                              !         the tab position
2          LENGTH,                             ! Search tab range length
2          LEFT_MARGIN,                        ! Source Text left column
2          RIGHT_MARGIN,                       ! Source Text right column
2          TABCNT,                             ! Number of tab characters
2          TABLEN,                             ! Spaces needed for the TAB character
2          TAB_POSITION,                       ! Point to the tab character
2          TMPBUF: VECTOR[SRCBUFSIZE,BYTE],    ! Expanded source text buffer
2          TMPBUFPTR,                          ! Pointer to the Expanded source text buffer
2          TERM_WIDTH,                         ! Terminal size - 8
2          WINDOW_WIDTH;                       ! The size of the source text to be displayed
2
2
2
2      ! Replace form feed character by space character.
2      !
2      BGNPTR = .BUFPTR;
2      LENGTH = .BUFLEN;
2      WHILE TRUE DO
3          BEGIN
3          FORMFEED_PTR = CH$FIND_CH(.LENGTH, .BGNPTR, 12);
3          IF .FORMFEED_PTR EQL 0 THEN EXITLOOP;
3          FORMFEED_PTR[0] = %C' ';
3          LENGTH = .LENGTH - (.FORMFEED_PTR - .BGNPTR + 1);
3          BGNPTR = .FORMFEED_PTR + 1;
2          END;
2
2
2      ! Expand the TAB character to spaces in the buffer which contains the
2      ! ASCII text of the source line.  Finally adjust the length of the original
2      ! buffer length.
2      !
2      TMPBUFPTR = TMPBUF;
2      BGNPTR = .BUFPTR;
2      LENGTH = .BUFLEN;
2      TABCNT = 0;
2      WHILE TRUE DO
3          BEGIN
3
```

```
        ! Locate the 'TAB' character in the buffer.  Pointed to by BGNPTR as
        ! the start search position, LENGTH is the given search range.
        !
        TAB_POSITION = CH$FIND_CH(.LENGTH,.BGNPTR,%C'	');


        ! There is no TAB character can be found, move the source text from
        ! BGNPTR position to TMPBUFPTR position.  Then exit the serach TAB loop.
        !
        IF .TAB_POSITION EQL 0
        THEN
            BEGIN
            CH$MOVE(.LENGTH, .BGNPTR, .TMPBUFPTR);
            EXITLOOP;
            END;


        ! Count the number of the TAB characters in the source text.
        !
        TABCNT = .TABCNT + 1;


        ! Move the source text before the TAB from BGNPTR postion to TMPBUFPTR
        ! position.
        !
        GAP = .TAB_POSITION - .BGNPTR;
        IF .GAP NEQ 0 THEN CH$MOVE(.GAP, .BGNPTR, .TMPBUFPTR);
        TMPBUFPTR = .TMPBUFPTR + .GAP;


        ! Calculate the number of spaces, this TAB character should expand to.
        ! Then move that number of spaces to the TMPBUFPTR position.
        !
        TABLEN = .SPACES[0] - ((.TMPBUFPTR - TMPBUF) MOD .SPACES[0]);
        CH$MOVE(.TABLEN , SPACES[1], .TMPBUFPTR);
        TMPBUFPTR = .TMPBUFPTR + .TABLEN;


        ! Update the source text record length by the number of spaces added.
        ! Adjust the search TAB range.  Skip the TAB charcter in the original
        ! source text buffer.
        !
        BUFLEN = .BUFLEN + .TABLEN;
        LENGTH = .LENGTH - .GAP - 1;
        IF .LENGTH EQL 0 THEN EXITLOOP;
        BGNPTR = .TAB_POSITION + 1;
        END;


    ! Expansion is all done.  Adjust the source record length by taking off the
    ! number of TABs encounted.
    !
    BUFLEN = .BUFLEN  - .TABCNT;


    ! Adjust the window margin by the length of the source text.
    !
```

```
LEFT_MARGIN = MAX(1, .DBG$SRC_LEFT_MARGIN);
RIGHT_MARGIN = MIN(.BUFLEN,.DBG$SRC_RIGHT_MARGIN);

! Check to see if the left margin is exceed the actual source text length.
!
IF .LEFT_MARGIN GTR .RIGHT_MARGIN
THEN
    WINDOW_WIDTH = 0
ELSE
    WINDOW_WIDTH = .RIGHT_MARGIN - .LEFT_MARGIN + 1;

TERM_WIDTH = .DBG$SRC_TERM_WIDTH - 8;

! Formats the line number and statement number, and outputs the line.
! Also folds long source lines when the output line exceed terminal width
! in length.
!
FST_LINE_FLAG = TRUE;
TMPBUFPTR = TMPBUF + .LEFT_MARGIN - 1;
WHILE TRUE DO
    BEGIN

    ! Format the Line's line number for the first line to be outputed.  For
    ! the line exceed terminal width in length, the second line starts with
    ! '        -:'.
    !
    IF .FST_LINE_FLAG
    THEN
        BEGIN
        IF .STMT_NUM EQL 0
        THEN
            BEGIN
            SELECTONE .LINE_NUM OF
                SET
                [1 TO 9]:
                    DBG$PRINT(UPLIT BYTE(%ASCIC '     '), 0);
                [10 TO 99]:
                    DBG$PRINT(UPLIT BYTE(%ASCIC '    '), 0);
                [100 TO 999]:
                    DBG$PRINT(UPLIT BYTE(%ASCIC '   '), 0);
                [1000 TO 9999]:
                    DBG$PRINT(UPLIT BYTE(%ASCIC '  '), 0);
                [10000 TO 99999]:
                    DBG$PRINT(UPLIT BYTE(%ASCIC ' '), 0);

                TES;

            END;

        DBG$PRINT(UPLIT BYTE(%ASCIC '!UL: '), .LINE_NUM);
        FST_LINE_FLAG = FALSE;
        END

    ELSE
```

```
; 2119        2243  3            DBG$PRINT(UPLIT BYTE(%ASCIC '      -: '), 0);
; 2120        2244  3
; 2121        2245  3        ! Output the corresponding source record.
; 2122        2246  3        !
; 2123        2247  3        LENGTH = MIN(.WINDOW_WIDTH, .TERM_WIDTH);
; 2124        2248  3        DBG$PRINT(UPLIT BYTE(%ASCIC '!AD'), .LENGTH, .TMPBUFPTR);
; 2125        2249  3        DBG$NEWLINE();
; 2126        2250  3
; 2127        2251  3
; 2128        2252  3        ! Check to see if any source text left to be displayed.
; 2129        2253  3        !
; 2130        2254  3        WINDOW_WIDTH = .WINDOW_WIDTH - .LENGTH;
; 2131        2255  3        IF .WINDOW_WIDTH LEQ 0 THEN EXITLOOP;
; 2132        2256  3        TMPBUFPTR = .TMPBUFPTR + .LENGTH;
; 2133        2257  3        END;
; 2134        2258  2
; 2135        2259  2    RETURN;
; 2136        2260  2
; 2137        2261  2
; 2138        2262  1    END;


                                                     .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

20  20  20  20  20  20  20  20  08  0013D P.AAP:     .ASCII  <8>\        \                  ;
            20  20  20  20  20  05  00146 P.AAQ:     .ASCII  <5>\     \                     ;
                20  20  20  20  04  0014C P.AAR:     .ASCII  <4>\    \                      ;
                    20  20  20  03  00151 P.AAS:     .ASCII  <3>\   \                       ;
                        20  20  02  00155 P.AAT:     .ASCII  <2>\  \                        ;
                            20  01  00158 P.AAU:     .ASCII  <1>\ \                         ;
            20  3A  4C  55  21  05  0015A P.AAV:     .ASCII  <5>\!UL: \                     ;
20  3A  2D  20  20  20  20  20  08  00160 P.AAW:     .ASCII  <8>\     -: \                  ;
                    44  41  21  03  00169 P.AAX:     .ASCII  <3>\!AD\                       ;

                                          SPACES=            P.AAP


                                                     .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                 OFFC 00000 DBG$SRC_OUTPUT_LINE:
                                                     .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11   ; 2015
            5E     FEFC   CE  9E 00002               MOVAB   -260(SP), SP                           ;
            57       0C   AC  D0 00007               MOVL    BUFPTR, BGNPTR                         ; 2105
            56       10   AC  D0 0000B               MOVL    BUFLEN, LENGTH                         ; 2106
    67      56            0C  3A 0000F 1$:           LOCC    #12, LENGTH, (BGNPTR)                  ; 2109
                          02  12 00013               BNEQ    2$                                     ;
                          51  D4 00015               CLRL    R1                                     ;
            52            51  D0 00017 2$:           MOVL    R1, FORMFEED_PTR                       ;
                          15  13 0001A               BEQL    3$                                     ; 2110
            62            20  90 0001C               MOVB    #32, (FORMFEED_PTR)                    ; 2111
    51      52            57  C3 0001F               SUBL3   BGNPTR, FORMFEED_PTR, R1               ; 2112
    50      56            51  C3 00023               SUBL3   R1, LENGTH, R0                         ;
            56       FF   A0  9E 00027               MOVAB   -1(R0), LENGTH                         ;
            57       01   A2  9E 0002B               MOVAB   1(R2), BGNPTR                          ; 2113
                          DE  11 0002F               BRB     1$                                     ; 2107
            5A       04   AE  9E 00031 3$:           MOVAB   TMPBUF, TMPBUFPTR                      ; 2121
```

```
                   57        0C   AC   D0 00035        MOVL    BUFPTR, BGNPTR                          ; 2122
                   56        10   AC   D0 00039        MOVL    BUFLEN, LENGTH                          ; 2123
                                  6E   D4 0003D        CLRL    TABCNT                                  ; 2124
           67      56             09   3A 0003F  4$:   LOCC    #9, LENGTH, (BGNPTR)                    ; 2132
                                  02   12 00043        BNEQ    5$                                      ;
                                  51   D4 00045        CLRL    R1                                      ;
                   5B             51   D0 00047  5$:   MOVL    R1, TAB_POSITION                        ;
                                  06   12 0004A        BNEQ    6$                                      ; 2138
           6A      67             56   28 0004C        MOVC3   LENGTH, (BGNPTR), (TMPBUFPTR)           ; 2141
                                  50   11 00050        BRB     8$                                      ; 2140
                                  6E   D6 00052  6$:   INCL    TABCNT                                  ; 2148
           58      5B             57   C3 00054        SUBL3   BGNPTR, TAB_POSITION, GAP               ; 2154
                                  04   13 00058        BEQL    7$                                      ; 2155
           6A      67             58   28 0005A        MOVC3   GAP, (BGNPTR), (TMPBUFPTR)              ;
                   5A             58   C0 0005E  7$:   ADDL2   GAP, TMPBUFPTR                          ; 2156
                   50        04   AE   9E 00061        MOVAB   TMPBUF, R0                              ; 2162
                   50             5A   C2 00065        SUBL2   TMPBUFPTR, R0                           ;
                   51 00000000'   EF   9A 00068        MOVZBL  SPACES, R1                              ;
    7E     00      50             01   7A 0006F        EMUL    #1, R0, #0, -(SP)                       ;
    50     50      8E             51   7B 00074        EDIV    R1, (SP)+, R0, R0                       ;
                   59 00000000'   EF   9A 00079        MOVZBL  SPACES, TABLEN                          ;
                   59             50   C0 00080        ADDL2   R0, TABLEN                              ;
           6A 00000000'   EF      59   28 00083        MOVC3   TABLEN, SPACES+1, (TMPBUFPTR)           ; 2163
                   5A             59   C0 0008B        ADDL2   TABLEN, TMPBUFPTR                       ; 2164
              10   AC             59   C0 0008E        ADDL2   TABLEN, BUFLEN                          ; 2171
           50      56             58   C3 00092        SUBL3   GAP, LENGTH, R0                         ; 2172
                   56        FF   A0   9E 00096        MOVAB   -1(R0), LENGTH                          ;
                                  06   13 0009A        BEQL    8$                                      ; 2173
                   57        01   AB   9E 0009C        MOVAB   1(R11), BGNPTR                          ; 2174
                                  9D   11 000A0        BRB     4$                                      ; 2125
              10   AC             6E   C2 000A2  8$:   SUBL2   TABCNT, BUFLEN                          ; 2181
                   50 00000000'   EF   D0 000A6        MOVL    DBG$SRC_LEFT_MARGIN, R0                 ; 2186
                                  03   14 000AD        BGTR    9$                                      ;
                                  50   01 D0 000AF     MOVL    #1, R0                                  ;
                                  51   50 D0 000B2 9$: MOVL    R0, LEFT_MARGIN                         ;
                   50        10   AC   D0 000B5        MOVL    BUFLEN, R0                              ; 2187
        00000000'  EF             50   D1 000B9        CMPL    R0, DBG$SRC_RIGHT_MARGIN                ;
                                  07   15 000C0        BLEQ    10$                                     ;
                   50 00000000'   EF   D0 000C2        MOVL    DBG$SRC_RIGHT_MARGIN, R0                ;
                   50             51   D1 000C9  10$:  CMPL    LEFT_MARGIN, RIGHT_MARGIN               ; 2192
                                  04   15 000CC        BLEQ    11$                                     ;
                                  52   D4 000CE        CLRL    WINDOW_WIDTH                            ; 2194
                                  07   11 000D0        BRB     12$                                     ;
                   50             51   C2 000D2  11$:  SUBL2   LEFT_MARGIN, R0                         ; 2196
                   52        01   A0   9E 000D5        MOVAB   1(R0), WINDOW_WIDTH                     ;
           54 00000000'   EF      08   C3 000D9  12$:  SUBL3   #8, DBG$SRC_TERM_WIDTH, TERM_WIDTH      ; 2198
                   55             01   D0 000E1        MOVL    #1, FST_LINE_FLAG                       ; 2205
                   5A        03 AE41   9E 000E4        MOVAB   TMPBUF-1[LEFT_MARGIN], TMPBUFPTR        ; 2206
                   03             55   E8 000E9  13$:  BLBS    FST_LINE_FLAG, 14$                      ; 2215
                                00A2   31 000EC        BRW     22$                                     ;
                             08   AC   D5 000EF  14$:  TSTL    STMT_NUM                                ; 2218
                                  03   13 000F2        BEQL    15$                                     ;
                                0086   31 000F4        BRW     21$                                     ;
                   53        04   AC   D0 000F7  15$:  MOVL    LINE_NUM, R3                            ; 2221
                                  0F   15 000FB        BLEQ    16$                                     ; 2223
                   09             53   D1 000FD        CMPL    R3, #9                                  ;
                                  0A   14 00100        BGTR    16$                                     ;
```

```
                                        7E  D4 00102          CLRL    -(SP)                          ; 2224
                          00000000'     EF  9F 00104          PUSHAB  P.AAQ                          ;
                                        6A  11 0010A          BRB     20$                            ;
                          0A            53  D1 0010C  16$:    CMPL    R3, #10                        ; 2225
                                        13  19 0010F          BLSS    17$                            ;
       00000063           8F            53  D1 00111          CMPL    R3, #99                        ;
                                        0A  14 00118          BGTR    17$                            ;
                                        7E  D4 0011A          CLRL    -(SP)                          ; 2226
                          00000000'     EF  9F 0011C          PUSHAB  P.AAR                          ;
                                        52  11 00122          BRB     20$                            ;
       00000064           8F            53  D1 00124  17$:    CMPL    R3, #100                       ; 2227
                                        13  19 0012B          BLSS    18$                            ;
       000003E7           8F            53  D1 0012D          CMPL    R3, #999                       ;
                                        0A  14 00134          BGTR    18$                            ;
                                        7E  D4 00136          CLRL    -(SP)                          ; 2228
                          00000000'     EF  9F 00138          PUSHAB  P.AAS                          ;
                                        36  11 0013E          BRB     20$                            ;
       000003E8           8F            53  D1 00140  18$:    CMPL    R3, #1000                      ; 2229
                                        13  19 00147          BLSS    19$                            ;
       0000270F           8F            53  D1 00149          CMPL    R3, #9999                      ;
                                        0A  14 00150          BGTR    19$                            ;
                                        7E  D4 00152          CLRL    -(SP)                          ; 2230
                          00000000'     EF  9F 00154          PUSHAB  P.AAT                          ;
                                        1A  11 0015A          BRB     20$                            ;
       00002710           8F            53  D1 0015C  19$:    CMPL    R3, #10000                     ; 2231
                                        18  19 00163          BLSS    21$                            ;
       0001869F           8F            53  D1 00165          CMPL    R3, #99999                     ;
                                        0F  14 0016C          BGTR    21$                            ;
                                        7E  D4 0016E          CLRL    -(SP)                          ; 2232
                          00000000'     EF  9F 00170          PUSHAB  P.AAU                          ;
       00000000G          00            02  FB 00176  20$:    CALLS   #2, DBG$PRINT                  ;
                                04      AC  DD 0017D  21$:    PUSHL   LINE_NUM                       ; 2238
                          00000000'     EF  9F 00180          PUSHAB  P.AAV                          ;
       00000000G          00            02  FB 00186          CALLS   #2, DBG$PRINT                  ;
                                        55  D4 0018D          CLRL    FST_LINE_FLAG                  ; 2239
                                        0F  11 0018F          BRB     23$                            ; 2215
                                        7E  D4 00191  22$:    CLRL    -(SP)                          ; 2243
                          00000000'     EF  9F 00193          PUSHAB  P.AAW                          ;
       00000000G          00            02  FB 00199          CALLS   #2, DBG$PRINT                  ;
                          50            52  D0 001A0  23$:    MOVL    WINDOW_WIDTH, R0               ; 2248
                          54            50  D1 001A3          CMPL    R0, TERM_WIDTH                 ;
                                        03  15 001A6          BLEQ    24$                            ;
                          50            54  D0 001A8          MOVL    TERM_WIDTH, R0                 ;
                          56            50  D0 001AB  24$:    MOVL    R0, LENGTH                     ;
                                0440    8F  BB 001AE          PUSHR   #^M<R6,R10>                    ; 2249
                          00000000'     EF  9F 001B2          PUSHAB  P.AAX                          ;
       00000000G          00            03  FB 001B8          CALLS   #3, DBG$PRINT                  ;
       00000000G          00            00  FB 001BF          CALLS   #0, DBG$NEWLINE                ; 2250
                          52            56  C2 001C6          SUBL2   LENGTH, WINDOW_WIDTH           ; 2255
                                        06  15 001C9          BLEQ    25$                            ; 2256
                          5A            56  C0 001CB          ADDL2   LENGTH, TMPBUFPTR              ; 2257
                                FF18    31  001CE         BRW     13$                            ; 2207
                                        04  001D1  25$:       RET                                    ; 2262

; Routine Size:  466 bytes,    Routine Base:  DBG$CODE + 0D0F
```

```
1 ROUTINE DBG$SRC_READ_FILE(SFCBPTR, REC_NUM, BUFPTR, BUFLEN): NOVALUE =
1 !
1 ! FUNCTION
1 !       This routine reads a specified source record from a specified source
1 !       file and returns the address and length of the read record.  This is
1 !       thus the routine which actually reads source lines from source files.
1 !
1 !       This routine assumes that the source file has already been opened by
1 !       DBG$SRC_OPEN_FILE.  It reads the source file via RMS or the LIBRARIAN
1 !       until the desired record number is reached.  The image corresponding
1 !       to that record number is then returned to the caller.
1 !
1 !       This routine also builds the Record File Address (RFA) Table for the
1 !       current source file as it reads.  This table can then be used on later
1 !       calls to position the file pointer directly at the desired line or on
1 !       a line shortly before the desired line.  The RFA table is obviously
1 !       used whenever available to take us closer to the desired line.  That
1 !       table is also expanded whenever we have to read past the last RFA in
1 !       the table.  In other words, no source line is read just to build up the
1 !       RFA Table, but whenever a source line must be scanned anyway to find a
1 !       desired source record, its RFA is recovered for use in the RFA Table.
1 !
1 ! INPUTS
1 !       SFCBPTR - A pointer to the Source File Control Block for the file to be
1 !                 read.  It is assumed that this control block has already been
1 !                 set up by routine DBG$SRC_OPEN_FILE.
1 !
1 !       REC_NUM - The Record Number within the file of the source record to be
1 !                 read in.
1 !
1 !       BUFPTR  - A longword location to receive a pointer to the input buffer.
1 !
1 !       BUFLEN  - A longword location to receive the input line's length.
1 !
1 ! OUTPUTS
1 !       BUFPTR  - A pointer to the buffer which contains the just read input
1 !                 line is returned to BUFPTR.
1 !
1 !       BUFLEN  - The length in characters of the just read input line is
1 !                 returned to BUFLEN.
1 !
1
2   BEGIN
2
2   MAP
2       SFCBPTR: REF SFCB$BLOCK,         ! Pointer to Source File Control Block
2       BUFPTR: REF VECTOR[1],           ! Pointer to a longword to receive the
2                                        !       input line's buffer pointer
2       BUFLEN: REF VECTOR[1];           ! Pointer to a longword to receive the
2                                        !       input line's length
2
2   LOCAL
2       CLOSEST_TABLE_ENTRY,
2       INPTR: BLOCK[8,BYTE],            ! Length and string address for the user
2                                        !        -supplied buffer
2       OUTPTR: BLOCK[8,BYTE],           ! Pointer to a string descriptor for the
2                                        !         actual record returned from LBR
```

```
2           SRCNAM: REF $NAM_DECL,                  ! NAME Block for RMS file
2           SRCRAB: REF $RAB_DECL,                  ! RAB for RMS file
2           STARTING_REC,
2           STATUS,                                 ! RAB and FIND status
2           TXTRFA: VECTOR[2],                      ! Pointer to RFA
2           RFA_TABLE : REF RFATBL$BLOCKVECTOR(RFATBL_ENTRIES),
2                                                   ! The RFA table for the file
2           RFATBL_SPACING;                         ! Number of source records per RFA
2                                                   ! table entry.
2
2
2       ! Determine whether to read from an RMS file or a module in a source library.
2       !
2       SRCNAM = .SFCBPTR[SFCB$L_NAMPTR];
2       CASE .SFCBPTR[SFCB$B_KIND] FROM SFCB$K_RMSFILE TO SFCB$K_LBRFILE OF
2           SET
2
2
2           ! Read from an RMS source file.
2           !
2           [SFCB$K_RMSFILE]:
3               BEGIN
3
3               ! Set up the RAB pointer and RFA table.
3               !
3               SRCRAB = .SFCBPTR[SFCB$L_RABPTR];
3               RFA_TABLE = .SFCBPTR[SFCB$L_RFATBLPTR];
3               RFATBL_SPACING = .SFCBPTR[SFCB$W_RFASPACING];
3
3
3               ! Find the line in the RFA table which represents the
3               ! record which is closest to the one we are looking for.
3               !
3               CLOSEST_TABLE_ENTRY = MIN( .SFCBPTR[SFCB$W_RFACURLEN],
3                                          (.REC_NUM-1)/.RFATBL_SPACING);
3               STARTING_REC = .CLOSEST_TABLE_ENTRY*.RFATBL_SPACING;
3
3
3               ! If the Source File is newly opened, get the first records's file
3               ! address field and store it in RFA table.
3               !
3               IF .SFCBPTR[SFCB$L_CURRECNUM] EQL 1
3               THEN
4                   BEGIN
4                   STATUS = $FIND(RAB = .SRCRAB);
4                   IF NOT .STATUS THEN SIGNAL(.STATUS);
4                   RFA_TABLE[0,RFATBL$L_RFA0] = .SRCRAB[RAB$L_RFA0];
4                   RFA_TABLE[0,RFATBL$W_RFA4] = .SRCRAB[RAB$W_RFA4];
4                   CLOSEST_TABLE_ENTRY = 0;
3                   END;
3
3
3               ! Take the RFA table entry that is closest to the desired record,
3               ! and position the RAB so that we are pointing to the record
3               ! corresponding to this RFA.
3               ! (If we are already positioned at the desired record we
3               ! skip this)
```

```
!
IF .REC_NUM NEQ .SFCBPTR[SFCB$L_CURRECNUM]
THEN
    BEGIN
    SRCRAB[RAB$L_RFA0] = .RFA_TABLE[.CLOSEST_TABLE_ENTRY,
            RFATBL$L_RFA0];
    SRCRAB[RAB$W_RFA4] = .RFA_TABLE[.CLOSEST_TABLE_ENTRY,
            RFATBL$W_RFA4];
    SFCBPTR[SFCB$L_CURRECNUM] = 1 + (.CLOSEST_TABLE_ENTRY *
            .RFATBL_SPACING);
    SRCRAB[RAB$B_RAC] = RAB$C_RFA;
    STATUS = $FIND(RAB = .SRCRAB);
    IF NOT .STATUS THEN SIGNAL(.STATUS);
    END;


! Sequentially read from the current record to the desired record
! number.  Skip over records containing only <Form Feed>;
! do not count them toward the record count.
!
SRCRAB[RAB$B_RAC] = RAB$C_SEQ;
INCR I FROM .SFCBPTR[SFCB$L_CURRECNUM]-1 TO .REC_NUM-1 DO
    BEGIN
    WHILE TRUE DO
        BEGIN
        STATUS = $GET(RAB = .SRCRAB);
        IF NOT .STATUS
        THEN
            SIGNAL(DBG$_UNAREASRC,2,.SRCNAM[NAM$B_RSL],.SRCNAM[NAM$L_RSA],.STATUS);


        ! If this flag is set, that means we do not want to skip
        ! the formfeed record.
        !
        IF .DBG$SRC_FORMFEED_FLAG THEN EXITLOOP;
        IF NOT ((.DBG$SRC_REC_BUF[0] EQL 12) AND
                (.SRCRAB[RAB$W_RSZ] EQL 1))
        THEN
            EXITLOOP;

        END;


    ! Check for whether we need to update the RFA table.
    ! We do this every "n" records, where "n" is RFATBL_SPACING.
    ! As a minor optimization, we do not update the table
    ! for the very first record we read - this table entry
    ! is already filled in.
    !
    IF ((.I MOD .RFATBL_SPACING) EQL 0) AND
       (.I NEQ .STARTING_REC)
    THEN
        BEGIN ! update table

        ! First check whether the table is full.
        !
        IF (.I / .RFATBL_SPACING) GEQ RFATBL_ENTRIES
```

```
5                  THEN
6                      BEGIN
6
6                      ! Compact the table.
6                      !
6                      RFATBL_SPACING = 2 * .RFATBL_SPACING;
6                      SFCBPTR[SFCB$W_RFASPACING] = .RFATBL_SPACING;
6                      INCR J FROM 0 TO RFATBL_ENTRIES-2 BY 2 DO
7                          BEGIN
7                          RFA_TABLE[.J/2,RFATBL$L_RFA0] =
7                              .RFA_TABLE[.J,RFATB[$L_RFA0];
7                          RFA_TABLE[.J/2,RFATBL$W_RFA4] =
7                              .RFA_TABLE[.J,RFATB[$W_RFA4];
6                          END;
5                      END;
5
5                  ! Update the table length and fill in the new
5                  ! table entry.
5                  !
5                  SFCBPTR[SFCB$W_RFACURLEN] = .I/.RFATBL_SPACING;
5                  IF .I MOD .RFATBL_SPACING EQL 0
5                  THEN
6                      BEGIN
6                      RFA_TABLE[.I/.RFATBL_SPACING,RFATBL$L_RFA0] =
6                          .SRCRAB[RAB$L_RFA0];
6                      RFA_TABLE[.I/.RFATBL_SPACING,RFATBL$W_RFA4] =
6                          .SRCRAB[RAB$W_RFA4];
5                      END;
5
4                  END; ! update the table
3
3              END; ! incr loop
3
3          ! Update the current record number field in SFCB. Set up the buffer
3          ! pointer and buffer length.
3          !
3          SFCBPTR[SFCB$L_CURRECNUM] = .REC_NUM + 1;
3          BUFPTR[0] = .SRCRAB[RAB$L_RBF];
3          BUFLEN[0] = .SRCRAB[RAB$W_RSZ];
3
2          END; ! RMS file case alternative
2
2
2      ! Read from a module in a Source Library File.
2      !
2      [SFCB$K_LBRFILE]:
3          BEGIN
3
3
3          ! If the current record number is beyond the desired record number,
3          ! reset the current record to 1 and set the record file address to the
3          ! beginning of the module.
3          !
3          IF .SFCBPTR[SFCB$L_CURRECNUM] GTR .REC_NUM
3          THEN
4              BEGIN
4              TXTRFA[0] = .SFCBPTR[SFCB$L_CUR_RFA0];
```

```
                        TXTRFA[1] = .SFCBPTR[SFCB$W_CUR_RFA4];
                        SFCBPTR[SFCB$L_CURRECNUM] = 1;
                        STATUS = LBR$FIND(SFCBPTR[SFCB$L_LBRINDEX],TXTRFA);
                        IF NOT .STATUS THEN SIGNAL(.STATUS);
                        END;


                    ! Sequentially read from the current record to the desired record
                    ! number.
                    !
                    INPTR[DSC$W_LENGTH] = SRCBUFSIZE;
                    INPTR[DSC$A_POINTER] = DBG$SRC_REC_BUF[0];
                    OUTPTR[DSC$A_POINTER] = DBG$SRC_REC_BUF[0];
                    INCR I FROM .SFCBPTR[SFCB$L_CURRECNUM] TO .REC_NUM DO
                        BEGIN
                        WHILE TRUE DO
                            BEGIN
                            STATUS = LBR$GET_RECORD(SFCBPTR[SFCB$L_LBRINDEX],INPTR,OUTPTR);
                            IF NOT .STATUS
                            THEN
                                SIGNAL(DBG$_UNAREASRC,2,.SRCNAM[NAM$B_RSL],.SRCNAM[NAM$L_RSA],.STATUS);

                            IF .DBG$SRC_FORMFEED_FLAG THEN EXITLOOP;
                            IF NOT ((.DBG$SRC_REC_BUF[0] EQL 12) AND
                                    (.OUTPTR[DSC$W_LENGTH] EQL 1))
                            THEN
                                EXITLOOP;

                            END;

                        END;


                    ! Update the current record number field in SFCB.  Set up buffer
                    ! pointer and buffer length.
                    !
                    SFCBPTR[SFCB$L_CURRECNUM] = .REC_NUM + 1;
                    BUFPTR[0] = .OUTPTR[DSC$A_POINTER];
                    BUFLEN[0] = .OUTPTR[DSC$W_LENGTH];

                    END;

                [INRANGE, OUTRANGE]:
                    $DBG_ERROR('DBGSOURCE\DBG$SRC_READ_FILE');

                TES;

            RETURN;

            END;


                                                            .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

24 47 42 44 5C 45 43 52 55 4F 53 47 42 44 1B 0016D P.AAY:   .ASCII  <27>\DBGSOURCE\<92>\DBG$SRC_READ_FILE\    ;
      45 4C 49 46 5F 44 41 45 52 5F 43 52 53 0017C                                                         ;
```

```
                                                        .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                  0FFC 00000 DBG$SRC_READ_FILE:
                                             .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11      ; 2263
                   5E        18    C2 00002  SUBL2    #24, SP                                   ;
                   53    04  AC    D0 00005  MOVL     SFCBPTR, R3                               ; 2333
                   57    20  A3    D0 00009  MOVL     32(R3), SRCNAM                            ;
  01               02    08  A3    8F 0000D  CASEB    8(R3), #2, #1                             ; 2334
                 017A      001A       00012 1$:  .WORD    2$-1$,-                               ;
                                                       13$-1$                                   ;
            00000000'        EF    9F 00016  PUSHAB   P.AAY                                     ; 2534
                             01    DD 0001C  PUSHL    #1                                        ;
            00028362         8F    DD 0001E  PUSHL    #164706                                   ;
       00000000G  00         03    FB 00024  CALLS    #3, LIB$SIGNAL                            ;
                             04    0002B     RET                                                ;
                   55    1C  A3    D0 0002C 2$:  MOVL     28(R3), SRCRAB                        ; 2345
                   54    10  A3    D0 00030  MOVL     16(R3), RFA_TABLE                         ; 2346
                   58    0C  A3    3C 00034  MOVZWL   12(R3), RFATBL_SPACING                    ; 2347
  5A          08   AC        01    C3 00038  SUBL3    #1, REC_NUM, R10                          ; 2354
  51               5A        58    C7 0003D  DIVL3    RFATBL_SPACING, R10, R1                   ;
                   50    0E  A3    3C 00041  MOVZWL   14(R3), R0                                ;
                   51        50    D1 00045  CMPL     R0, R1                                    ;
                             03    15 00048  BLEQ     3$                                        ;
                   50        51    D0 0004A  MOVL     R1, R0                                    ;
                   52        50    D0 0004D 3$:  MOVL     R0, CLOSEST_TABLE_ENTRY               ; 2353
  5B               52        58    C5 00050  MULL3    RFATBL_SPACING, CLOSEST_TABLE_ENTRY, -    ; 2355
                                                      STARTING_REC                              ;
              01         30  A3    D1 00054  CMPL     48(R3), #1                                ; 2361
                             23    12 00058  BNEQ     5$                                        ;
                             55    DD 0005A  PUSHL    SRCRAB                                    ; 2364
       00000000G  00         01    FB 0005C  CALLS    #1, SYS$FIND                              ;
                  59         50    D0 00063  MOVL     R0, STATUS                                ;
                  09         59    E8 00066  BLBS     STATUS, 4$                                ; 2365
                             59    DD 00069  PUSHL    STATUS                                    ;
       00000000G  00         01    FB 0006B  CALLS    #1, LIB$SIGNAL                            ;
                  64     10  A5    D0 00072 4$:  MOVL     16(SRCRAB), (RFA_TABLE)               ; 2366
             04   A4     14  A5    B0 00076  MOVW     20(SRCRAB), 4(RFA_TABLE)                  ; 2367
                             52    D4 0007B  CLRL     CLOSEST_TABLE_ENTRY                       ; 2368
          30  A3         08  AC    D1 0007D 5$:  CMPL     REC_NUM, 48(R3)                       ; 2378
                             37    13 00082  BEQL     6$                                        ;
  50              52         06    C5 00084  MULL3    #6, CLOSEST_TABLE_ENTRY, R0               ; 2382
                       6044  9F    00088     PUSHAB   (R0)[RFA_TABLE]                           ; 2381
          10  A5             9E    D0 0008B  MOVL     @(SP)+, 16(SRCRAB)                        ;
                     04 A044 9F    0008F     PUSHAB   4(R0)[RFA_TABLE]                          ; 2383
          14  A5             9E    B0 00093  MOVW     @(SP)+, 20(SRCRAB)                        ;
                  52         58    C4 00097  MULL2    RFATBL_SPACING, R2                        ; 2386
          30  A3         01  A2    9E 0009A  MOVAB    1(R2), 48(R3)                             ; 2385
          1E  A5             02    90 0009F  MOVB     #2, 30(SRCRAB)                            ; 2387
                             55    DD 000A3  PUSHL    SRCRAB                                    ; 2388
       00000000G  00         01    FB 000A5  CALLS    #1, SYS$FIND                              ;
                  59         50    D0 000AC  MOVL     R0, STATUS                                ;
                  09         59    E8 000AF  BLBS     STATUS, 6$                                ; 2389
                             59    DD 000B2  PUSHL    STATUS                                    ;
       00000000G  00         01    FB 000B4  CALLS    #1, LIB$SIGNAL                            ;
```

```
                                      1E    A5   94 000BB 6$:    CLRB    30(SRCRAB)                                  ; 2397
             52        30       A3          02   C3 000BE        SUBL3   #2, 48(R3), I                               ; 2398
                                          00AF   31 000C3        BRW     12$                                         ;
                                            55   DD 000C6 7$:    PUSHL   SRCRAB                                      ; 2402
                  00000000G     00          01   FB 000C8        CALLS   #1, SYS$GET                                 ;
                                59          50   D0 000CF        MOVL    R0, STATUS                                  ;
                                18          59   E8 000D2        BLBS    STATUS, 8$                                  ; 2403
                                            59   DD 000D5        PUSHL   STATUS                                      ; 2405
                                      04    A7   DD 000D7        PUSHL   4(SRCNAM)                                   ;
                                7E    03    A7   9A 000DA        MOVZBL  3(SRCNAM), -(SP)                            ;
                                            02   DD 000DE        PUSHL   #2                                          ;
                                00028D00    8F   DD 000E0        PUSHL   #167168                                     ;
                  00000000G     00          05   FB 000E6        CALLS   #5, LIB$SIGNAL                              ;
                                0F 00000000'  EF   E8 000ED 8$:  BLBS    DBG$SRC_FORMFEED_FLAG, 9$                   ; 2411
                                0C 00000000'  EF   91 000F4      CMPB    DBG$SRC_REC_BUF, #12                        ; 2412
                                            06   12 000FB        BNEQ    9$                                          ;
                                01    22    A5   B1 000FD        CMPW    34(SRCRAB), #1                              ; 2413
                                            C3   13 00101        BEQL    7$                                          ;
     7E          00             52          01   7A 00103 9$:    EMUL    #1, I, #0, -(SP)                            ; 2426
     50          50             8E          58   7B 00108        EDIV    RFATBL_SPACING, (SP)+, R0, R0               ;
                                            50   D5 0010D        TSTL    R0                                          ;
                                            64   12 0010F        BNEQ    12$                                         ;
                                5B          52   D1 00111        CMPL    I, STARTING_REC                             ; 2427
                                            5F   13 00114        BEQL    12$                                         ;
                 50             52          58   C7 00116        DIVL3   RFATBL_SPACING, I, R0                       ; 2433
                                32          50   D1 0011A        CMPL    R0, #50                                     ;
                                2E          19   0011D           BLSS    11$                                         ;
                                58          02   C4 0011F        MULL2   #2, RFATBL_SPACING                          ; 2439
                       0C       A3          58   B0 00122        MOVW    RFATBL_SPACING, 12(R3)                      ; 2440
                                            50   D4 00126        CLRL    J                                           ; 2443
                 51             50          02   C7 00128 10$:   DIVL3   #2, J, R1                                   ;
                                51          06   C4 0012C        MULL2   #6, R1                                      ;
                 56             50          06   C5 0012F        MULL3   #6, J, R6                                   ; 2444
                                          6144   9F 00133        PUSHAB  (R1)[RFA_TABLE]                             ;
                                          6644   9F 00136        PUSHAB  (R6)[RFA_TABLE]                             ;
                                9E          9E   D0 00139        MOVL    @(SP)+, @(SP)+                              ;
                                       04 A144   9F 0013C        PUSHAB  4(R1)[RFA_TABLE]                            ; 2446
                                       04 A644   9F 00140        PUSHAB  4(R6)[RFA_TABLE]                            ;
                                9E          9E   B0 00144        MOVW    @(SP)+, @(SP)+                              ;
   FFDB          50             02          30   F1 00147        ACBL    #48, #2, J, 10$                             ; 2441
                 50             52          58   C7 0014D 11$:   DIVL3   RFATBL_SPACING, I, R0                       ; 2453
                       0E       A3          50   B0 00151        MOVW    R0, 14(R3)                                  ;
     7E          00             52          01   7A 00155        EMUL    #1, I, #0, -(SP)                            ; 2454
     51          51             8E          58   7B 0015A        EDIV    RFATBL_SPACING, (SP)+, R1, R1               ;
                                            51   D5 0015F        TSTL    R1                                          ;
                                            12   12 00161        BNEQ    12$                                         ;
                                50          06   C4 00163        MULL2   #6, R0                                      ; 2457
                                          6044   9F 00166        PUSHAB  (R0)[RFA_TABLE]                             ; 2458
                                9E    10    A5   D0 00169        MOVL    16(SRCRAB), @(SP)+                          ;
                                       04 A044   9F 0016D        PUSHAB  4(R0)[RFA_TABLE]                            ; 2460
                                9E    14    A5   B0 00171        MOVW    20(SRCRAB), @(SP)+                          ;
   FF4B          52             01          5A   F1 00175 12$:   ACBL    R10, #1, I, 7$                              ; 2398
          30     A3      08     AC          01   C1 0017B        ADDL3   #1, REC_NUM, 48(R3)                         ; 2470
                         0C     BC    28    A5   D0 00181        MOVL    40(SRCRAB), @BUFPTR                         ; 2471
                         10     BC    22    A5   3C 00186        MOVZWL  34(SRCRAB), @BUFLEN                         ; 2472
                                            04   0018B           RET                                                 ; 2334
                         08     AC    30    A3   D1 0018C 13$:   CMPL    48(R3), REC_NUM                             ; 2487
```

```
                                     28  15 00191         BLEQ    14$                               ;
                    6E         34    A3  D0 00193         MOVL    52(R3), TXTRFA                    ; 2490
            04      AE         38    A3  3C 00197         MOVZWL  56(R3), TXTRFA+4                  ; 2491
            30      A3               01  D0 0019C         MOVL    #1, 48(R3)                        ; 2492
                                     5E  DD 001A0         PUSHL   SP                                ; 2493
                               3C    A3  9F 001A2         PUSHAB  60(R3)                            ;
     00000000G      00               02  FB 001A5         CALLS   #2, LBR$FIND                      ;
                    59               50  D0 001AC         MOVL    R0, STATUS                        ;
                    09               59  E8 001AF         BLBS    STATUS, 14$                       ; 2494
                                     59  DD 001B2         PUSHL   STATUS                            ;
     00000000G      00               01  FB 001B4         CALLS   #1, LIB$SIGNAL                    ;
            10      AE       0100    8F  B0 001BB  14$:   MOVW    #256, INPTR                       ; 2501
            14      AE  00000000'    EF  9E 001C1         MOVAB   DBG$SRC_REC_BUF, INPTR+4          ; 2502
            0C      AE  00000000'    EF  9E 001C9         MOVAB   DBG$SRC_REC_BUF, OUTPTR+4         ; 2503
                    54         3C    A3  9E 001D1         MOVAB   60(R3), R4                        ; 2508
     52     30      A3               01  C3 001D5         SUBL3   #1, 48(R3), I                     ;
                                     43  11 001DA         BRB     17$                               ;
                               08    AE  9F 001DC  15$:   PUSHAB  OUTPTR                            ;
                               14    AE  9F 001DF         PUSHAB  INPTR                             ;
                                     54  DD 001E2         PUSHL   R4                                ;
     00000000G      00               03  FB 001E4         CALLS   #3, LBR$GET_RECORD                ;
                    59               50  D0 001EB         MOVL    R0, STATUS                        ;
                    18               59  E8 001EE         BLBS    STATUS, 16$                       ; 2509
                                     59  DD 001F1         PUSHL   STATUS                            ; 2511
                               04    A7  DD 001F3         PUSHL   4(SRCNAM)                         ;
                    7E         03    A7  9A 001F6         MOVZBL  3(SRCNAM), -(SP)                  ;
                                     02  DD 001FA         PUSHL   #2                                ;
                         00028D00    8F  DD 001FC         PUSHL   #167168                           ;
     00000000G      00               05  FB 00202         CALLS   #5, LIB$SIGNAL                    ;
                    0F  00000000'    EF  E8 00209  16$:   BLBS    DBG$SRC_FORMFEED_FLAG, 17$        ; 2513
                    0C  00000000'    EF  91 00210         CMPB    DBG$SRC_REC_BUF, #12              ; 2514
                                     06  12 00217         BNEQ    17$                               ;
                    01         08    AE  B1 00219         CMPW    OUTPTR, #1                        ; 2515
                                     BD  13 0021D         BEQL    15$                               ;
            B8      52         08    AC  F3 0021F  17$:   AOBLEQ  REC_NUM, I, 15$                   ; 2504
     30     A3      08    AC         01  C1 00224         ADDL3   #1, REC_NUM, 48(R3)               ; 2527
            0C      BC         0C    AE  D0 0022A         MOVL    OUTPTR+4, @BUFPTR                 ; 2528
            10      BC         08    AE  3C 0022F         MOVZWL  OUTPTR, @BUFLEN                   ; 2529
                                         04 00234         RET                                       ; 2540

; Routine Size:  565 bytes,    Routine Base:  DBG$CODE + 0EE1
```

```
1 ROUTINE DBG$SRC_SEARCH(BUFPTR, BUFLEN): =
1 !
1 ! FUNCTION
1 !       This routine actually performs the search.  This routine locates the
1 !       desired search string in the input buffer.  Lowercase letters are
1 !       translated to uppercase letter before the search is performed.
1 !       If there is a match, this routine returns TRUE, otherwise FALSE is
1 !       returned.  There are two ways of searching for the desired string.
1 !       One is to search the string that has the match, ie. SEARCH for "ABC",
1 !       then "ABCXXX", or "XXXABC", or "XXABCXX" would be found.  The other
1 !       is to search the string that has the exact match, i.e. the characters
1 !       to the left and the right of the search string "ABC" can not be a valid
1 !       character in identifiers in the currently set language.
1 !
1 ! INPUT
1 !       BUFPTR  - A pointer to a buffer which contains the ASCII text of the
1 !                 source line to be printed.
1 !
1 !       BUFLEN  - The length in characters of the source line pointed to by
1 !                 BUFPTR.
1 !
1 ! OUTPUT
1 !       This routine returns TRUE of the search string was found in the current
1 !                 line and FALSE if it was not.
1 !
2     BEGIN
2
2     MAP
2       BUFPTR: REF VECTOR[,BYTE];        ! Pointer to the ASCII image to print
2
2     LOCAL
2       CHAR,                             ! Serve as an index character index
2                                         !       into the Translate Table
2       DES_STR: BLOCK[8,BYTE],           ! Address of destination string
2                                         !       descriptor
2       SRC_STR: BLOCK[8,BYTE],           ! Address of source string descriptor
2       STATUS,                           ! Return status from System routines
2       TMPBUF: VECTOR[SRCBUFSIZE,BYTE],! Tempory buffer which holds all the
2                                         !       letters in uppercase
2       TMPLEN,                           ! Remaining size of tempory buffer
2       TMPPTR;                           ! Pointer to the tempory buffer
2
2
2
2     ! Translate all lowercase letters to uppercase via STR$UPCASE.  Before
2     ! the call, build string descriptors for STR$UPCASE as passing parameters.
2     ! Array descriptor, character-coded text.
2     !
2     DES_STR[DSC$B_CLASS] = DSC$K_CLASS_S;
2     DES_STR[DSC$B_DTYPE] = DSC$K_DTYPE_T;
2     DES_STR[DSC$W_LENGTH] = .BUFLEN;
2     DES_STR[DSC$A_POINTER] = TMPBUF;
2     SRC_STR[DSC$B_CLASS] = DSC$K_CLASS_S;
2     SRC_STR[DSC$B_DTYPE] = DSC$K_DTYPE_T;
2     SRC_STR[DSC$W_LENGTH] = .BUFLEN;
2     SRC_STR[DSC$A_POINTER] = .BUFPTR;
2     STATUS = STR$UPCASE(DES_STR,SRC_STR);
```

```

        ! Search for the desired string in the tempory buffer.  If the desired
        ! string is not found, return FALSE.  If the desired string is found
        ! and search qualifier is SEARCH, return TRUE.  If the desired
        ! string is found and search qualifier is IDENTIFIER, then the left
        ! and right characters of the desired string are checked.  If the
        ! desired string is identifier, return TRUE.  Otherwise search goes
        ! on till reaching to end of the buffer.
        !
        TMPLEN = .BUFLEN;
        TMPPTR = TMPBUF;
        STATUS = TRUE;
        WHILE TRUE DO
            BEGIN
            TMPPTR = CH$FIND_SUB(.TMPLEN, .TMPPTR, .DBG$SRC_SEARCH_STRING[0],
                        DBG$SRC_SEARCH_STRING[1]);


            ! Check to see if the matched string is found.  If such string is not
            ! found, return the status.
            !
            IF .TMPPTR EQL 0
            THEN
                BEGIN
                STATUS = FALSE;
                EXITLOOP;
                END;


            ! Check to see if the exact match option /IDENT is used.  If it is, goes on
            ! with more checking.  Otherwise, we are all done.
            !
            IF (.DBG$SRC_SEARCH_FLAG EQL SEARCH_STRING_ALL) OR
               (.DBG$SRC_SEARCH_FLAG EQL SEARCH_STRING_NEXT)
            THEN
                BEGIN
                STATUS = TRUE;
                EXITLOOP;
                END;


            ! Check to see if the characters on each side of the pattern string are
            ! valid in identifiers for the current language.  This check is done by
            ! looking up each character's attributes in the DBGPARSER Character
            ! Table for the currently SET language.  First we check the character
            ! to the left of the found string, and then the character on the right.
            ! STATUS is set to FALSE if the string appears to be part of a larger
            ! identifier.
            !
            IF .TMPPTR NEQ TMPBUF
            THEN
                BEGIN
                CHAR = (.TMPPTR - 1) - TMPBUF;
                STATUS = TRUE;
                IF .DBG$GL_CHARTBL[.TMPBUF[.CHAR], CHRTBL$V_IDENT_START]  OR
                    .DBG$GL_CHARTBL[.TMPBUF[.CHAR], CHRTBL$V_IDENT_MIDDLE] OR
```

```
                .DBG$GL_CHARTBL[.TMPBUF[.CHAR], CHRTBL$V_IDENT_END]
            THEN
                STATUS = FALSE;

            END;


        ! Check the right hand side character.  First make sure there is a
        ! right hand side character.  Again, we set STATUS to FALSE if the
        ! string appears to be part of a larger identifier.
        !
        TMPPTR = .TMPPTR + .DBG$SRC_SEARCH_STRING[0];
        IF .TMPPTR GEQ (TMPBUF + .BUFLEN) THEN EXITLOOP;
        IF (.TMPPTR LEQ (TMPBUF + .BUFLEN)) AND .STATUS
        THEN
            BEGIN
            CHAR = .TMPPTR - TMPBUF;
            STATUS = TRUE;
            IF .DBG$GL_CHARTBL[.TMPBUF[.CHAR], CHRTBL$V_IDENT_START]  OR
                .DBG$GL_CHARTBL[.TMPBUF[.CHAR], CHRTBL$V_IDENT_MIDDLE] OR
                .DBG$GL_CHARTBL[.TMPBUF[.CHAR], CHRTBL$V_IDENT_END]
            THEN
                STATUS = FALSE;

            END;


        ! We have found the desired match, return good status to the caller.
        !
        IF .STATUS THEN EXITLOOP;
        TMPLEN = TMPBUF + .BUFLEN - .TMPPTR;
        IF .TMPLEN LEQ 0 THEN EXITLOOP;

        END;            ! End of WHILE loop.

    RETURN .STATUS;

    END;



                              0FFC 00000 DBG$SRC_SEARCH:
                                               .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 2541
          5B 0000000G    00   9E 00002         MOVAB    DBG$GL_CHARTBL, R11                           ;
          5A 00000000'   EF   9E 00009         MOVAB    DBG$SRC_SEARCH_FLAG, R10                      ;
          59 00000000'   EF   9E 00010         MOVAB    DBG$SRC_SEARCH_STRING, R9                     ;
          5E       FEF0  CE   9E 00017         MOVAB    -272(SP), SP                                  ;
     FA   AD       010E  8F   B0 0001C         MOVW     #270, DES_STR+2                               ; 2590
     F8   AD         08  AC   B0 00022         MOVW     BUFLEN, DES_STR                               ; 2591
     FC   AD              6E  9E 00027         MOVAB    TMPBUF, DES_STR+4                             ; 2592
     F2   AD       010E  8F   B0 0002B         MOVW     #270, SRC_STR+2                               ; 2594
     F0   AD         08  AC   B0 00031         MOVW     BUFLEN, SRC_STR                               ; 2595
     F4   AD         04  AC   D0 00036         MOVL     BUFPTR, SRC_STR+4                             ; 2596
                     F0  AD   9F 0003B         PUSHAB   SRC_STR                                       ; 2597
                     F8  AD   9F 0003E         PUSHAB   DES_STR                                       ;
```

```
                    00000000G  00        02  FB 00041       CALLS   #2, STR$UPCASE                                    ;
                               56        50  D0 00048       MOVL    R0, STATUS                                        ;
                               58   08   AC  D0 0004B       MOVL    BUFLEN, TMPLEN                                    ; 2608
                               52        6E  9E 0004F       MOVAB   TMPBUF, TMPPTR                                    ; 2609
                               56        01  D0 00052       MOVL    #1, STATUS                                        ; 2610
                               57        6E  9E 00055       MOVAB   TMPBUF, R7                                        ; 2667
                               57   08   AC  C0 00058       ADDL2   BUFLEN, R7                                        ;
                               55        69  9A 0005C 1$:   MOVZBL  DBG$SRC_SEARCH_STRING, R5                         ; 2613
       62          58    01    A9        55  39 0005F       MATCHC  R5, DBG$SRC_SEARCH_STRING+1, TMPLEN, -            ; 2614
                                                                    (TMPPTR)
                                         03  13 00065       BEQL    2$                                                ;
                               53        55  D0 00067       MOVL    R5, R3                                            ;
                               53        55  C2 0006A 2$:   SUBL2   R5, R3                                            ;
                               52        53  D0 0006D       MOVL    R3, TMPPTR                                        ;
                                         04  12 00070       BNEQ    3$                                                ; 2620
                                         56  D4 00072       CLRL    STATUS                                            ; 2623
                                         0D  11 00074       BRB     5$                                                ; 2622
                               03        6A  D1 00076 3$:   CMPL    DBG$SRC_SEARCH_FLAG, #3                           ; 2631
                                         05  13 00079       BEQL    4$                                                ;
                               04        6A  D1 0007B       CMPL    DBG$SRC_SEARCH_FLAG, #4                           ; 2632
                                         05  12 0007E       BNEQ    6$                                                ;
                               56        01  D0 00080 4$:   MOVL    #1, STATUS                                        ; 2635
                                         72  11 00083 5$:   BRB     12$                                               ; 2634
                               50        6E  9E 00085 6$:   MOVAB   TMPBUF, R0                                        ; 2648
                               50        52  D1 00088       CMPL    TMPPTR, R0                                        ;
                                         29  13 0008B       BEQL    8$                                                ;
                               50        6E  9E 0008D       MOVAB   TMPBUF, R0                                        ; 2651
                   53          52        50  C3 00090       SUBL3   R0, TMPPTR, R3                                    ;
                               54   FF   A3  9E 00094       MOVAB   -1(R3), CHAR                                      ;
                               56        01  D0 00098       MOVL    #1, STATUS                                        ; 2652
                               51        6B  D0 0009B       MOVL    DBG$GL_CHARTBL, R1                                ; 2653
                               50      6E44  9A 0009E       MOVZBL  TMPBUF[CHAR], R0                                  ;
                               0E      6140  E8 000A2       BLBS    (R1)[R0], 7$                                      ;
                                       6140  DF 000A6       PUSHAL  (R1)[R0]                                          ; 2654
                   07          9E        01  E0 000A9       BBS     #1, @(SP)+, 7$                                    ;
                                       6140  DF 000AD       PUSHAL  (R1)[R0]                                          ; 2655
                   02          9E        02  E1 000B0       BBC     #2, @(SP)+, 8$                                    ;
                                         56  D4 000B4 7$:   CLRL    STATUS                                            ; 2657
                               50        69  9A 000B6 8$:   MOVZBL  DBG$SRC_SEARCH_STRING, R0                         ; 2666
                               52        50  C0 000B9       ADDL2   R0, TMPPTR                                        ;
                               57        52  D1 000BC       CMPL    TMPPTR, R7                                        ; 2667
                                         36  18 000BF       BGEQ    12$                                               ;
                                         28  14 000C1       BGTR    10$                                               ; 2668
                               28        56  E9 000C3       BLBC    STATUS, 11$                                       ;
                               50        6E  9E 000C6       MOVAB   TMPBUF, R0                                        ; 2671
                   54          52        50  C3 000C9       SUBL3   R0, TMPPTR, CHAR                                  ;
                               56        01  D0 000CD       MOVL    #1, STATUS                                        ; 2672
                               51        6B  D0 000D0       MOVL    DBG$GL_CHARTBL, R1                                ; 2673
                               50      6E44  9A 000D3       MOVZBL  TMPBUF[CHAR], R0                                  ;
                               0E      6140  E8 000D7       BLBS    (R1)[R0], 9$                                      ;
                                       6140  DF 000DB       PUSHAL  (R1)[R0]                                          ; 2674
                   07          9E        01  E0 000DE       BBS     #1, @(SP)+, 9$                                    ;
                                       6140  DF 000E2       PUSHAL  (R1)[R0]                                          ; 2675
                   02          9E        02  E1 000E5       BBC     #2, @(SP)+, 10$                                   ;
                                         56  D4 000E9 9$:   CLRL    STATUS                                            ; 2677
                               09        56  E8 000EB 10$:  BLBS    STATUS, 12$                                       ; 2684
                   58          57        52  C3 000EE 11$:  SUBL3   TMPPTR, R7, TMPLEN                                ; 2685
```

```
                          03  15 000F2          BLEQ    12$                  ; 2686
                        FF65  31 000F4          BRW     1$                   ;
        50                56  D0 000F7 12$:     MOVL    STATUS, R0           ; 2690
                              04 000FA          RET                          ; 2692

; Routine Size:  251 bytes,    Routine Base:  DBG$CODE + 1116
```

```
1 GLOBAL ROUTINE DBG$SRC_SEARCH_CMD(MODRSTPTR, LOW_LNUM, LOW_STMT,
1                                   HIGH_LNUM, HIGH_STMT,
1                                   SEARCH_BY_STRING_FLAG, SEARCH_NEXT_FLAG): NOVALUE =
1 !
1 ! FUNCTION
1 !       This routine accepts a Module RST Entry pointer, a line number range,
1 !       and flags set to indicate the options are used in SEARCH command.
1 !       The options for the SEARCH command are SEARCH by IDENTIFIER or SEARCH
1 !       by STRING; find NEXT matched string or ALL matched string in the range.
1 !       This routine then calls DBG$SRC_TYPE_LNUM_SOURCE to type out source
1 !       lines in that range from that module which contains matched search
1 !       string.
1 !
1 ! INPUTS
1 !       MODRSTPTR - A pointer to the Module RST Entry of the module in which
1 !                   the source lines are to be looked up.
1 !
1 !       LOW_LNUM - The line number of the first source line to be searched.
1 !                  This is assumed to be a line number in the MODRSTPTR module.
1 !
1 !       LOW_STMT - The statement number, if any, associated with the first
1 !                  source line to be searched.
1 !
1 !       HIGH_LNUM - The line number of the last source line to be searched.
1 !                   This too is assumed to be a line in the MODRSTPTR module.
1 !
1 !       HIGH_STMT - The statement number, if any, associated with the last
1 !                   source line to be searched.
1 !
1 !       SEARCH_BY_STRING_FLAG - Flag set to TRUE if /STRING is used, this
1 !                               means search for any string that matches the
1 !                               given search string.  Otherwise flag set to
1 !                               FALSE if /IDENTIFIER is used, this means
1 !                               search for the exact match for the given string.
1 !
1 !       SEARCH_NEXT_FLAG - Flag set to TRUE if /NEXT is used, this means
1 !                          search for the first match string in the given range.
1 !                          Otherwise flag set to FALSE if /ALL is used, this
1 !                          means search for all the match stings in the range.
1 !
1 ! IMPLICIT INPUT
1 !       DBG$SRC_SEARCH_STRING - Buffer contains the desired search string.
1 !
1 ! OUTPUTS
1 !       NONE
1 !
1
2     BEGIN
2
2     LOCAL
2         SEARCH_FLAG;                        ! Flag set to non-zero value to indicate
2                                             !       the kind of SEARCH option is
2                                             !       active
2
2     ! If no search string string is given, signal the warning message.
2     !
```

```
; 2629      2750  2    IF .DBG$SRC_SEARCH_STRING[0] EQL 0 THEN SIGNAL(DBG$_NOSRCHSTR);
; 2630      2751  2
; 2631      2752  2    ! Set up the SEARCH command option.  The value set to the SEARCH FLAG
; 2632      2753  2    ! is 1, or 2, or 3, or 4.  1 -- SEARCH/IDENT/ALL.  2 -- SEARCH/IDENT/NEXT.
; 2633      2754  2    ! 3 -- SEARCH/STRING/ALL.  4 -- SEARCH/STRING/NEXT.
; 2634      2755  2    !
; 2635      2756  2    SEARCH_FLAG = .SEARCH_BY_STRING_FLAG * 2 + .SEARCH_NEXT_FLAG + 1;
; 2636      2757  2
; 2637      2758  2
; 2638      2759  2    ! Perform the search.
; 2639      2760  2    !
; 2640      2761  2    DBG$SRC_TYPE_LNUM_SOURCE(.MODRSTPTR, .LOW_LNUM, .LOW_STMT, .HIGH_LNUM,
; 2641      2762  2                             .HIGH_STMT, .SEARCH_FLAG, FALSE);
; 2642      2763  2
; 2643      2764  2    RETURN;
; 2644      2765  2
; 2645      2766  2    END;
; 2646      2767  1


                              0000 00000        .ENTRY  DBG$SRC_SEARCH_CMD, Save nothing          ; 2693
           00000000'   EF      95 00002         TSTB    DBG$SRC_SEARCH_STRING                     ; 2750
                       0D      12 00008         BNEQ    1$                                        ;
           00028E28    8F      DD 0000A         PUSHL   #167464                                   ;
 00000000G  00                 01 FB 00010      CALLS   #1, LIB$SIGNAL                            ;
      50         18    AC      D0 00017  1$:    MOVL    SEARCH_BY_STRING_FLAG, R0                 ; 2757
      50         1C BC40       3E 0001B         MOVAW   @SEARCH_NEXT_FLAG[R0], SEARCH_FLAG        ;
                       50      D6 00020         INCL    SEARCH_FLAG                               ;
                       7E      D4 00022         CLRL    -(SP)                                     ; 2762
                       50      DD 00024         PUSHL   SEARCH_FLAG                               ; 2763
      7E         10    AC      7D 00026         MOVQ    HIGH_LNUM, -(SP)                          ; 2762
      7E         08    AC      7D 0002A         MOVQ    LOW_LNUM, -(SP)                           ;
                 04    AC      DD 0002E         PUSHL   MODRSTPTR                                 ;
   0000V  CF                   07 FB 00031      CALLS   #7, DBG$SRC_TYPE_LNUM_SOURCE              ;
                               04 00036         RET                                               ; 2767

; Routine Size:  55 bytes,    Routine Base:  DBG$CODE + 1211
```

```
; 2648     2768 1 GLOBAL ROUTINE DBG$SRC_SET_MAX_FILES(MAX_FILES): NOVALUE =
; 2649     2769 1 !
; 2650     2770 1 ! FUNCTION
; 2651     2771 1 !      This routine sets the maximum number of source files DEBUG will keep
; 2652     2772 1 !      open at any one time.  It is called during the processing of the
; 2653     2773 1 !      SET MAX_SOURCE_FILES command.  The purpose of this command is to give
; 2654     2774 1 !      the user some control over the number of source files DEBUG will keep
; 2655     2775 1 !      open.  The reason for this is that DEBUG can interfere with the correct
; 2656     2776 1 !      operation of the user program if the program itself has many files open
; 2657     2777 1 !      and the extra DEBUG files cause the program to exceed its open channel
; 2658     2778 1 !      quota.
; 2659     2779 1 !
; 2660     2780 1 ! INPUTS
; 2661     2781 1 !      MAX_FILES - The maximum number of source files DEBUG is to keep open
; 2662     2782 1 !                  at any one time.
; 2663     2783 1 !
; 2664     2784 1 ! OUTPUTS
; 2665     2785 1 !      NONE
; 2666     2786 1 !
; 2667     2787 1
; 2668     2788 2     BEGIN
; 2669     2789 2
; 2670     2790 2
; 2671     2791 2
; 2672     2792 2     ! Make sure the number of files the user specified is reasonable; otherwise
; 2673     2793 2     ! signal an error.
; 2674     2794 2     !
; 2675     2795 3     IF (.MAX_FILES LSS 1) OR (.MAX_FILES GTR MAX_MAX_FILES)
; 2676     2796 2     THEN
; 2677     2797 2         SIGNAL(DBG$_INVNUMSRC);
; 2678     2798 2
; 2679     2799 2
; 2680     2800 2     ! Set the maximum number of open source files in DBG$SRC_MAX_FILES and then
; 2681     2801 2     ! call DBG$SRC_INIT to initialize that number of Source File Control Blocks.
; 2682     2802 2     !
; 2683     2803 2     DBG$SRC_MAX_FILES = .MAX_FILES;
; 2684     2804 2     DBG$SRC_INIT();
; 2685     2805 2     RETURN;
; 2686     2806 2
; 2687     2807 1     END;


                               0000 00000      .ENTRY  DBG$SRC_SET_MAX_FILES, Save nothing       ; 2768
                    04     AC    D5 00002      TSTL    MAX_FILES                                 ; 2795
                           06    15 00005      BLEQ    1$                                        ;
           14       04     AC    D1 00007      CMPL    MAX_FILES, #20                            ;
                           0D    15 0000B      BLEQ    2$                                        ;
               00028CC8    8F    DD 0000D 1$:  PUSHL   #167112                                   ; 2797
  00000000G  00            01    FB 00013      CALLS   #1, LIB$SIGNAL                            ;
  00000000'  EF     04     AC    D0 0001A 2$:  MOVL    MAX_FILES, DBG$SRC_MAX_FILES              ; 2803
       EE60  CF            00    FB 00022      CALLS   #0, DBG$SRC_INIT                          ; 2804
                                 04 00027      RET                                               ; 2807

; Routine Size:  40 bytes,    Routine Base:  DBG$CODE + 1248
```

```
GLOBAL ROUTINE DBG$SRC_SET_SOURCE(MODRSTPTR, DIRLIST): NOVALUE =
!
! FUNCTION
!       This routine handles the semantic processing of the SET SOURCE and SET
!       SOURCE/MODULE=xxx commands.  It accepts a Module RST Entry pointer if
!       /MODULE was specified on the command and it accepts a list of directory
!       names, each given by a Source Directory Search List Entry.  It then
!       builds a Source Directory Search List Header Block for the specified
!       module and enters this block on the linked list pointed to by variable
!       DBG$SRC_DIR_LIST.  It also throws away any directory search list super-
!       ceded by the new list.
!
!       This routine verfies each directory name on the list, using the $PARSE
!       system service. If any are invalid, an error is signalled.
!
!       The source directory search lists are eventually used by the routine
!       DBG$SRC_OPEN_FILE.
!
! INPUTS
!       MODRSTPTR - A pointer to the Module RST Entry of the module specified
!                   on the SET SOURCE/MODULE command.  If the /MODULE qualifier
!                   was not present on the command, MODRSTPTR must be zero.
!
!       DIRLIST - A pointer to a source directory search list.  More precisely,
!                 DIRLIST points to the first of a list of singly linked Source
!                 Directory Search List (SDSL) Entries.  Each SDSL Entry con-
!                 tains a directory name (or even a full file name) in counted
!                 ASCII.  DIRLIST may not be zero--an empty directory list on
!                 the SET SOURCE command is a syntax error.
!
! OUTPUTS
!       NONE
!

    BEGIN

    MAP
        MODRSTPTR: REF RST$ENTRY,           ! Pointer to Module RST Entry or zero
        DIRLIST: REF SDSL$ENTRY;            ! Pointer to directory list

    LOCAL
        BLANK_COUNT,                        ! Holds a count of the number of
                                            !     leading blanks in the directory
                                            !     name as the user entered it.
        CHAR,                               ! Used to hold a single character
                                            !     during blank-stripping.
        CHAR_PTR,                           ! Pointer that is used to step through
                                            !     a directory name
                                            !     character by character, during
                                            !     blank-stripping.
        COUNT,                              ! Holds a count of the number of
                                            !     non-blank characters in a
                                            !     directory name that is being
                                            !     verified.
        DIRPTR: REF SDSL$HEADER,            ! Pointer that is used to walk through
                                            !     the linked list of headers.
        DUMMY_BUF: VECTOR [132, BYTE],      ! Output buffer for call to $PARSE.
```

```
                                       !       (this is not actually used since
                                       !       we are only interested in the
                                       !       status code returned by $PARSE)
    ENTRYPTR: REF SDSL$ENTRY,          ! Pointer that is used to walk through
                                       !       a linked list of directory names.
    LAST_DIRPTR: REF SDSL$HEADER,      ! Points to the last header in the list.
    NEW_DIRPTR: REF SDSL$HEADER,       ! Points to the new header record that
                                       !       is allocated.
    STATUS,                            ! Return code from $PARSE
    STRING_BUF: VECTOR [132, BYTE],    ! Input buffer for call to $PARSE
    STRING_BUF_PTR,                    ! Pointer to an output buffer where a
                                       !       directory name will be copied
                                       !          during validation of that
                                       !          directory name.
    TSTFAB: $FAB_DECL,                 ! FAB block for $PARSE
    TSTNAM: $NAM_DECL;                 ! NAM block for $PARSE



! Reinitialize all Source File Control Blocks. This forces presently
! open files to be reopened using the new source directories we are
! about to set.
!
DBG$SRC_INIT ();


! Initialize the fields of the FAB and the NAM blocks
!
$FAB_INIT ( FAB = TSTFAB,
            DNA = STRING_BUF,
            DNS = 132);
$NAM_INIT ( NAM = TSTNAM,
            ESA = DUMMY_BUF,
            ESS = 132,
            RSA = DUMMY_BUF,
            RSS = 132);


! We first verify the list that is given to us.  Walk through the list of
! directory names, verfifying that each one is valid.
!
ENTRYPTR = .DIRLIST;
WHILE .ENTRYPTR NEQ 0 DO
    BEGIN

    ! Initialize char_ptr to the start of the directory name
    !
    CHAR_PTR = ENTRYPTR[SDSL$A_ENT_DIRNAME];


    ! Inintialize STRING_BUF_PTR to the address of the string buffer
    ! that is declared in the FAB.
    !
    STRING_BUF_PTR = STRING_BUF;
```

```
        ! Initialize blank_count and count

        BLANK_COUNT = 0;
        COUNT = 0;


        ! Loop through the leading blanks to strip them off.
        !
        WHILE .BLANK_COUNT LSS .ENTRYPTR[SDSL$B_ENT_DIRLEN] DO
            BEGIN
            CHAR = CH$RCHAR_A (CHAR_PTR);
            IF .CHAR NEQ DBG$K_BLANK THEN EXITLOOP;
            BLANK_COUNT = .BLANK_COUNT + 1;
            END;


        ! If the name is all blanks, signal an error. This is caused by
        ! a pair of commas with only blanks between them.
        !
        IF .BLANK_COUNT EQL .ENTRYPTR[SDSL$B_ENT_DIRLEN]
        THEN
            SIGNAL(DBG$_INVDIRNAM, 1, UPLIT BYTE (%ASCIC ' '));


        ! Now loop through the non-blank characters. Move each character
        ! to the STRING_BUF_PTR buffer temporarily and translate to
        ! upper case.
        !
        WHILE .COUNT + .BLANK_COUNT LSS .ENTRYPTR[SDSL$B_ENT_DIRLEN] DO
            BEGIN
            COUNT = .COUNT + 1;
            IF .CHAR LEQ 'z' AND .CHAR GEQ 'a'
            THEN
                CHAR = .CHAR - ('a' - 'A');

            CH$WCHAR_A (.CHAR, STRING_BUF_PTR);
            CHAR = CH$RCHAR_A (CHAR_PTR);
            END;


        ! Copy the blank_stripped and translated directory name back
        ! into the original entry.
        !
        CH$MOVE (.COUNT, STRING_BUF, ENTRYPTR[SDSL$A_ENT_DIRNAME]);
        ENTRYPTR[SDSL$B_ENT_DIRLEN] = .COUNT;


        ! Now check the directory name for syntactic validity. This is done
        ! by $PARSE so that RMS does the checking.
        !
        TSTFAB[FAB$B_DNS] = .COUNT;
        STATUS = $PARSE (FAB = TSTFAB);
        IF NOT .STATUS
        THEN
            SIGNAL(DBG$_INVDIRNAM, 1, ENTRYPTR[SDSL$B_ENT_DIRLEN], .STATUS);

        ENTRYPTR = .ENTRYPTR[SDSL$L_ENT_FLINK];
```

```
                END;


             ! If we get this far we have a valid directory list.  Attempt to find a
             ! directory list for the same module in the current Source Directory Search
             ! List.  If we find one, we remove it from the list.
             !
             DIRPTR = .DBG$SRC_DIR_LIST;
             WHILE .DIRPTR NEQ 0 DO
                 BEGIN
                 IF .DIRPTR[SDSL$L_MODPTR] EQL .MODRSTPTR
                 THEN
                     BEGIN
                     DBG$SRC_CANCEL_SOURCE (.DIRPTR[SDSL$L_MODPTR]);
                     EXITLOOP;
                     END;

                 DIRPTR = .DIRPTR[SDSL$L_FLINK];
                 END;


             ! Allocate and build a new Source Directory List Header Block.
             ! Insert it at the end of the Source Directory Search List.
             !
             NEW_DIRPTR = DBG$GET_MEMORY (SDSL$K_HDR_SIZE);
             NEW_DIRPTR[SDSL$L_LIST_PTR] = .DIRLIST;
             NEW_DIRPTR[SDSL$L_MODPTR] = .MODRSTPTR;


             ! Walk to the end of the Source Directory Search List.
             !
             DIRPTR = DBG$SRC_DIR_LIST;
             WHILE .DIRPTR[SDSL$L_FLINK] NEQ 0 DO
                 DIRPTR = .DIRPTR[SDSL$L_FLINK];

             DIRPTR[SDSL$L_FLINK] = .NEW_DIRPTR;
             RETURN;

             END;


                                                         .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

                                 20  01  00189 P.AAZ:    .ASCII  <1>\ \                                  ;


                                                         .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                     0FFC 00000          .ENTRY  DBG$SRC_SET_SOURCE, Save R2,R3,R4,R5,R6,R7,-; 2808
                                                                 R8,R9,R10,R11                          ;
                      5E     FE44   CE   9E 00002        MOVAB   -444(SP), SP                           ;
                EE53  CF            00   FB 00007        CALLS   #0, DBG$SRC_INIT                       ; 2888
  0050    8F       00  6E           00   2C 0000C        MOVC5   #0, (SP), #0, #80, $RMS_PTR            ; 2895
                             64     AE      00013                                                       ;
                64    AE     5003   8F   B0 00015        MOVW    #20483, $RMS_PTR                       ;
```

```
                        7A   AE              02  90 0001B        MOVB    #2, $RMS_PTR+22                         ;
                      0083   CE              02  90 0001F        MOVB    #2, $RMS_PTR+31                         ;
                      0094   CE      00B4    CE  9E 00024        MOVAB   STRING_BUF, $RMS_PTR+48                 ;
                      0099   CE        84    8F  90 0002B        MOVB    #-124, $RMS_PTR+53                      ;
    0060  8F       00        6E              00  2C 00031        MOVC5   #0, (SP), #0, #96, $RMS_PTR             ; 2900
                                       04    AE     00038                                                        ;
                        04   AE      6002    8F  B0 0003A        MOVW    #24578, $RMS_PTR                        ;
                        06   AE        84    8F  90 00040        MOVB    #-124, $RMS_PTR+2                       ;
                        08   AE      FF7C    CD  9E 00045        MOVAB   DUMMY_BUF, $RMS_PTR+4                   ;
                        0E   AE        84    8F  90 0004B        MOVB    #-124, $RMS_PTR+10                      ;
                        10   AE      FF7C    CD  9E 00050        MOVAB   DUMMY_BUF, $RMS_PTR+12                  ;
                             57        08    AC  D0 00056        MOVL    DIRLIST, ENTRYPTR                       ; 2906
                                             03  12 0005A 1$:    BNEQ    2$                                      ; 2907
                                       00A3  31 0005C            BRW     9$                                      ;
                             5B        05    A7  9E 0005F 2$:    MOVAB   5(ENTRYPTR), CHAR_PTR                   ; 2913
                             5A      00B4    CE  9E 00063        MOVAB   STRING_BUF, STRING_BUF_PTR              ; 2919
                                             59  D4 00068        CLRL    BLANK_COUNT                             ; 2924
                                             56  D4 0006A        CLRL    COUNT                                   ; 2925
    59        04   A7        08              00  ED 0006C 3$:    CMPZV   #0, #8, 4(ENTRYPTR), BLANK_COUNT        ; 2930
                                             0C  15 00072        BLEQ    4$                                      ;
                             58              8B  9A 00074        MOVZBL  (CHAR_PTR)+, CHAR                       ; 2932
                             20              58  D1 00077        CMPL    CHAR, #32                               ; 2933
                                             04  12 0007A        BNEQ    4$                                      ;
                                             59  D6 0007C        INCL    BLANK_COUNT                             ; 2934
                                             EC  11 0007E        BRB     3$                                      ; 2930
    59        04   A7        08              00  ED 00080 4$:    CMPZV   #0, #8, 4(ENTRYPTR), BLANK_COUNT        ; 2941
                                             15  12 00086        BNEQ    5$                                      ;
                                00000000'    EF  9F 00088        PUSHAB  P.AAZ                                   ; 2943
                                             01  DD 0008E        PUSHL   #1                                      ;
                                0002897A     8F  DD 00090        PUSHL   #166266                                 ;
                   00000000G  00             03  FB 00096        CALLS   #3, LIB$SIGNAL                          ;
                        50    56             59  C1 0009D 5$:    ADDL3   BLANK_COUNT, COUNT, R0                  ; 2950
    50        04   A7         08             00  ED 000A1        CMPZV   #0, #8, 4(ENTRYPTR), R0                 ;
                                             1F  15 000A7        BLEQ    7$                                      ;
                                             56  D6 000A9        INCL    COUNT                                   ; 2952
                   0000007A   8F             58  D1 000AB        CMPL    CHAR, #122                              ; 2953
                                             0C  14 000B2        BGTR    6$                                      ;
                   00000061   8F             58  D1 000B4        CMPL    CHAR, #97                               ;
                                             03  19 000BB        BLSS    6$                                      ;
                              58             20  C2 000BD        SUBL2   #32, CHAR                               ; 2955
                              8A             58  90 000C0 6$:    MOVB    CHAR, (STRING_BUF_PTR)+                 ; 2957
                              58             8B  9A 000C3        MOVZBL  (CHAR_PTR)+, CHAR                       ; 2958
                                             D5  11 000C6        BRB     5$                                      ; 2950
              05   A7       00B4   CE        56  28 000C8 7$:    MOVC3   COUNT, STRING_BUF, 5(ENTRYPTR)          ; 2965
                              04   A7        56  90 000CF        MOVB    COUNT, 4(ENTRYPTR)                      ; 2966
                            0099   CE        56  90 000D3        MOVB    COUNT, TSTFAB+53                        ; 2972
                                       64    AE  9F 000D8        PUSHAB  TSTFAB                                  ; 2973
                   00000000G  00             01  FB 000DB        CALLS   #1, SYS$PARSE                           ;
                              6E             50  D0 000E2        MOVL    R0, STATUS                              ;
                              14             6E  E8 000E5        BLBS    STATUS, 8$                              ; 2974
                                             6E  DD 000E8        PUSHL   STATUS                                  ; 2976
                                       04    A7  9F 000EA        PUSHAB  4(ENTRYPTR)                             ;
                                             01  DD 000ED        PUSHL   #1                                      ;
                                0002897A     8F  DD 000EF        PUSHL   #166266                                 ;
                   00000000G  00             04  FB 000F5        CALLS   #4, LIB$SIGNAL                          ;
                              57             67  D0 000FC 8$:    MOVL    (ENTRYPTR), ENTRYPTR                    ; 2978
                                       FF58  31 000FF            BRW     1$                                      ; 2907
```

```
            52 00000000'  EF  D0 00102 9$:   MOVL    DBG$SRC_DIR_LIST, DIRPTR          ; 2986
                          16  13 00109 10$:  BEQL    12$                               ; 2987
      04    AC        08  A2  D1 0010B       CMPL    8(DIRPTR), MODRSTPTR              ; 2989
                          0A  12 00110       BNEQ    11$                               ;
                      08  A2  DD 00112       PUSHL   8(DIRPTR)                         ; 2992
    EC76    CF            01  FB 00115       CALLS   #1, DBG$SRC_CANCEL_SOURCE         ;
                          05  11 0011A       BRB     12$                               ; 2991
            52            62  D0 0011C 11$:  MOVL    (DIRPTR), DIRPTR                  ; 2996
                          E8  11 0011F       BRB     10$                               ; 2987
                          03  DD 00121 12$:  PUSHL   #3                                ; 3003
00000000G   00            01  FB 00123       CALLS   #1, DBG$GET_MEMORY                ;
      04    A0        08  AC  D0 0012A       MOVL    DIRLIST, 4(NEW_DIRPTR)            ; 3004
      08    A0        04  AC  D0 0012F       MOVL    MODRSTPTR, 8(NEW_DIRPTR)          ; 3005
            52 00000000'  EF  9E 00134       MOVAB   DBG$SRC_DIR_LIST, DIRPTR          ; 3010
                          62  D5 0013B 13$:  TSTL    (DIRPTR)                          ; 3011
                          05  13 0013D       BEQL    14$                               ;
            52            62  D0 0013F       MOVL    (DIRPTR), DIRPTR                  ; 3012
                          F7  11 00142       BRB     13$                               ;
            62            50  D0 00144 14$:  MOVL    NEW_DIRPTR, (DIRPTR)              ; 3014
                              04 00147       RET                                       ; 3017

; Routine Size:  328 bytes,    Routine Base:  DBG$CODE + 1270
```

```
GLOBAL ROUTINE DBG$SRC_SHOW_SOURCE: NOVALUE =
!
! FUNCTION
!       This routine handles the semantics of the SHOW SOURCE command.  It thus
!       formats and prints the contents of the directory search lists pointed to
!       by DBG$SRC_DIR_LIST.  If one or more SET SOURCE/MODULE commands are in
!       effect, the output has this general format:
!
!                       source directory search list for MODNAME1:
!                               [FIRST.DIR]
!                               LOGNAME$
!                       source directory search list for MODNAME2:
!                               [DIR1]
!                               [DIR2.SUBDIR]
!                               DB1:[DIR3.DIR31.DIR32]
!                       source directory search list for all other modules:
!                               [.SUBDIR]
!
!       If a SET SOURCE command is in effect but no SET SOURCE/MODULE command is
!       in effect, this format is used:
!
!                       source directory search list for all modules:
!                               [DIRNAME1]
!                               [DIRNAME2]
!
!       And if no SET SOURCE command is in effect at all, this is output:
!
!                       no source directory search list in effect
!
!       All directory names are output exactly as the user originally entered
!       them.  There is no attempt to translate logical names or determine
!       defaults until the directory name is actually used to open a file.
!       Hence no such translations are done for the SHOW SOURCE display.
!
! INPUTS
!       NONE
!
! OUTPUTS
!       NONE
!

    BEGIN

    LOCAL
        DEF_DIRPTR : REF SDSL$HEADER,   ! Points to the default
                                        !       Source Directory Search List,
                                        !       if one exists.
        DIRPTR : REF SDSL$HEADER,       ! A pointer used to walk through the
                                        !       linked list of Source Directory
                                        !       Search List header blocks.
        MODNAMEPTR;                     ! Points to a module name.

    ! If there is no Source Directory Search List in effect, print a message
    ! to that effect and return.
    !
    IF .DBG$SRC_DIR_LIST EQL 0
    THEN
```

```
3                 BEGIN
3                 DBG$PRINT(UPLIT BYTE(%ASCIC 'no directory search list in effect'));
3                 DBG$NEWLINE();
3                 RETURN;
2                 END;
2
2
2         ! If SET SOURCE is in effect but no SET SOURCE/MODULE= is in effect,
2         ! print the "for all modules" header message and the corresponding
2         ! list of directory names. Then return.
2         !
2         DIRPTR = .DBG$SRC_DIR_LIST;
2         IF   .DIRPTR[SDSL$L_MODPTR] EQL 0
2             AND
2             .DIRPTR[SDSL$L_FLINK] EQL 0
2         THEN
3             BEGIN
3             DBG$PRINT (UPLIT BYTE(
3                     %ASCIC 'source directory search list for all modules:'));
3             DBG$NEWLINE();
3             OUTPUT_DIR_LIST(.DIRPTR[SDSL$L_LIST_PTR]);
3             RETURN;
2             END;
2
2
2         ! If we reach this point, there is at least one directory list for a
2         ! specific module. Loop through all the Source Directory Search List
2         ! Header Blocks and print the directory list for each corresponding
2         ! module.
2         !
2         DEF_DIRPTR = 0;
2         WHILE .DIRPTR NEQ 0 DO
3             BEGIN
3
3
3             ! Check for default directory list (SET SOURCE with no
3             ! module specified). Store this away in DEF_DIRPTR since
3             ! it will be printed last.
3             !
3             IF .DIRPTR[SDSL$L_MODPTR] EQL 0
3             THEN
3                 DEF_DIRPTR = .DIRPTR
3
3
3             ! Else this SDSL Header Block is for a specific module. Print a
3             ! header message with the module name and print the corresponding
3             ! directory names.
3             !
3             ELSE
4                 BEGIN
4                 DBG$STA_SYMNAME (.DIRPTR[SDSL$L_MODPTR], MODNAMEPTR);
4                 DBG$PRINT (UPLIT BYTE(
4                             %ASCIC 'source directory search list for !AC:'),
4                             .MODNAMEPTR);
4                 DBG$NEWLINE();
4                 OUTPUT_DIR_LIST (.DIRPTR[SDSL$L_LIST_PTR]);
3                 END;
```

```
; 3014          3132  3
; 3015          3133  3
; 3016          3134  3           ! Link to the next SDSL Header Block and loop
; 3017          3135  3           !
; 3018          3136  3           DIRPTR = .DIRPTR[SDSL$L_FLINK];
; 3019          3137  2           END; ! While loop
; 3020          3138  2
; 3021          3139  2       ! Now output the default directory list used for all other modules
; 3022          3140  2       ! if such a list is present. Then return.
; 3023          3141  2       !
; 3024          3142  2       IF .DEF_DIRPTR NEQ 0
; 3025          3143  2       THEN
; 3026          3144  3           BEGIN
; 3027          3145  3           DBG$PRINT (UPLIT BYTE(
; 3028          3146  3                       %ASCIC 'source directory list for all other modules:'));
; 3029          3147  3           DBG$NEWLINE();
; 3030          3148  3           OUTPUT_DIR_LIST (.DEF_DIRPTR[SDSL$L_LIST_PTR]);
; 3031          3149  2           END;
; 3032          3150  2
; 3033          3151  2       RETURN;
; 3034          3152  2
; 3035          3153  1       END;


                                                                   .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

73 20 79 72 6F 74 63 65 72 69 64 20 6F 6E 22 0018B P.ABA:  .ASCII  \"no directory search list in effect\        ;
65 20 6E 69 20 74 73 69 6C 20 68 63 72 61 65 0019A                                                                 ;
                                    74 63 65 66 66 001A9                                                                 ;
6F 74 63 65 72 69 64 20 65 63 72 75 6F 73 2D 001AE P.ABB:  .ASCII  \-source directory search list for all mo\   ;
20 74 73 69 6C 20 68 63 72 61 65 73 20 79 72 001BD                                                                 ;
               6F 6D 20 6C 6C 61 20 72 6F 66 001CC                                                                 ;
                              3A 73 65 6C 75 64 001D6          .ASCII  \dules:\                                    ;
6F 74 63 65 72 69 64 20 65 63 72 75 6F 73 25 001DC P.ABC:  .ASCII  \%source directory search list for !AC:\     ;
20 74 73 69 6C 20 68 63 72 61 65 73 20 79 72 001EB                                                                 ;
                        3A 43 41 21 20 72 6F 66 001FA                                                                 ;
6F 74 63 65 72 69 64 20 65 63 72 75 6F 73 2C 00202 P.ABD:  .ASCII  \,source directory list for all other mod\   ;
6C 6C 61 20 72 6F 66 20 74 73 69 6C 20 79 72 00211                                                                 ;
               64 6F 6D 20 72 65 68 74 6F 20 00220                                                                 ;
                              3A 73 65 6C 75 0022A          .ASCII  \ules:\                                     ;


                                                                   .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                      00FC 00000         .ENTRY  DBG$SRC_SHOW_SOURCE, Save R2,R3,R4,R5,R6,R7 ; 3018
                 57 00000000'  EF   9E 00002         MOVAB   DBG$SRC_DIR_LIST, R7                               ;
                 56 00000000G  00   9E 00009         MOVAB   DBG$NEWLINE, R6                                    ;
                 55 00000000G  00   9E 00010         MOVAB   DBG$PRINT, R5                                      ;
                 54 00000000'  EF   9E 00017         MOVAB   P.ABA, R4                                          ;
                 5E            04   C2 0001E         SUBL2   #4, SP                                             ;
                               67   D5 00021         TSTL    DBG$SRC_DIR_LIST                                   ; 3073
                               09   12 00023         BNEQ    1$                                                 ;
                               54   DD 00025         PUSHL   R4                                                 ; 3076
                 65            01   FB 00027         CALLS   #1, DBG$PRINT                                      ;
                 66            00   FB 0002A         CALLS   #0, DBG$NEWLINE                                    ; 3077
```

```
                                   04 0002D         RET                                         ; 3075
           52              67      D0 0002E 1$:     MOVL    DBG$SRC_DIR_LIST, DIRPTR            ; 3086
                    08     A2      D5 00031         TSTL    8(DIRPTR)                           ; 3087
                           12      12 00034         BNEQ    2$                                  ;
                           62      D5 00036         TSTL    (DIRPTR)                            ; 3089
                           0E      12 00038         BNEQ    2$                                  ;
                    23     A4      9F 0003A         PUSHAB  P.ABB                               ; 3092
           65              01      FB 0003D         CALLS   #1, DBG$PRINT                       ;
           66              00      FB 00040         CALLS   #0, DBG$NEWLINE                     ; 3094
                    04     A2      DD 00043         PUSHL   4(DIRPTR)                           ; 3095
                           44      11 00046         BRB     7$                                  ;
                           53      D4 00048 2$:     CLRL    DEF_DIRPTR                          ; 3105
                           52      D5 0004A 3$:     TSTL    DIRPTR                              ; 3106
                           2E      13 0004C         BEQL    6$                                  ;
                    08     A2      D5 0004E         TSTL    8(DIRPTR)                           ; 3114
                           05      12 00051         BNEQ    4$                                  ;
           53              52      D0 00053         MOVL    DIRPTR, DEF_DIRPTR                  ; 3116
                           1F      11 00056         BRB     5$                                  ;
                           5E      DD 00058 4$:     PUSHL   SP                                  ; 3125
                    08     A2      DD 0005A         PUSHL   8(DIRPTR)                           ;
 00000000G  00             02      FB 0005D         CALLS   #2, DBG$STA_SYMNAME                 ;
                           6E      DD 00064         PUSHL   MODNAMEPTR                          ; 3128
                    51     A4      9F 00066         PUSHAB  P.ABC                               ; 3126
           65              02      FB 00069         CALLS   #2, DBG$PRINT                       ;
           66              00      FB 0006C         CALLS   #0, DBG$NEWLINE                     ; 3129
                    04     A2      DD 0006F         PUSHL   4(DIRPTR)                           ; 3130
     0000V  CF             01      FB 00072         CALLS   #1, OUTPUT_DIR_LIST                 ;
           52              62      D0 00077 5$:     MOVL    (DIRPTR), DIRPTR                    ; 3136
                           CE      11 0007A         BRB     3$                                  ; 3106
                           53      D5 0007C 6$:     TSTL    DEF_DIRPTR                          ; 3142
                           11      13 0007E         BEQL    8$                                  ;
                    77     A4      9F 00080         PUSHAB  P.ABD                               ; 3145
           65              01      FB 00083         CALLS   #1, DBG$PRINT                       ;
           66              00      FB 00086         CALLS   #0, DBG$NEWLINE                     ; 3147
                    04     A3      DD 00089         PUSHL   4(DEF_DIRPTR)                       ; 3148
     0000V  CF             01      FB 0008C 7$:     CALLS   #1, OUTPUT_DIR_LIST                 ;
                                   04 00091 8$:     RET                                         ; 3153

; Routine Size:  146 bytes,    Routine Base:  DBG$CODE + 13B8
```

```
ROUTINE DBG$SRC_TYPE_LINE(LINE_NUM, STMT_NUM, SRC_FILE, REC_NUM, STEP_FLAG): =
!
! FUNCTION
!       This routine accepts the line number and statement number of a source
!       line, a pointer to the Declare Source File DST command for the corre-
!       sponding source file, and the record number of the line within that
!       file.  It then reads in the text of the source line and causes it to be
!       typed out to the user.  It is called by DBG$SRC_TYPE_LNUM_SOURCE each
!       time that routine finds a desired line number in the current module's
!       Source Line Correlation DST Record.
!
!       This routine calls DBG$SRC_OPEN_FILE with the pointer to the Declare
!       Source File DST command (where the full file declaration is found).
!       DBG$SRC_OPEN_FILE opens the file (unless it is already opened), and
!       returns a pointer to the file's Source File Control Block (SFCB).  This
!       pointer is then passed to DBG$SRC_READ_FILE to read in the actual source
!       image.  If SEARCH command is active then calls SRC$SRC_SEARCH to locate
!       the desired string.  If matched string is not found, exit from the
!       routine and return status back to caller.  Finally, DBG$SRC_OUTPUT_LINE is
!       is called to print the source line.
!
! INPUTS
!       LINE_NUM - The line number of the line to be read and printed.  If the
!                  line is unnumbered (has no line number as is true for comments
!                  in some languages), LINE_NUM should be negative.
!
!       STMT_NUM - The statement number of the line to be read and printed.  If
!                  a statement number is not applicable (which is the case in
!                  most languages), STMT_NUM should be zero.
!
!       SRC_FILE - A pointer to the Declare Source File command for the desired
!                  source file in the Source Line Correlation DST Record.  This
!                  pointer defines which file the desired source line should be
!                  read from.
!
!       REC_NUM - The record number in the SRC_FILE source file of the source
!                  line to be read and printed.  This record number also comes
!                  from the Source Line Correlation DST Record.
!
!       STEP_FLAG - Flag set to TRUE to indicate this is called from STEP.
!
! OUTPUTS
!       NONE
!

    BEGIN

    MAP
        SRC_FILE: REF DST$SRC_COMMAND;      ! Pointer to Define Source File command
                                            !       in Source Correlation DST Record

    LOCAL
        BUFLEN,                             ! The length in characters of the input
                                            !       line
        BUFPTR: REF VECTOR[,BYTE],          ! Pointer to the input buffer
        SEARCH_STATUS,                      ! Flag set to TRUE if the search string
                                            !       is found, for other conditions,
```

```
2               flag set to FALSE
2       SRC_FILE_CNTRL_BLK:                     ! Pointer to the Source File
2                           REF SFCB$BLOCK;     !     Control Block
2
2
2     ! Signal an error if SRC_FILE is not declared or set in Source File
2     ! Correlation DST Record.
2     !
2     IF .SRC_FILE EQL 0 THEN SIGNAL(DBG$_INVDSTREC);
2
2
2     ! Abort the source output if the user interrupted it with Control-Y DEBUG.
2     !
2     $ABORT_ON_CONTROL_Y;
2
2
2     ! Initialize SEARCH_STATUS to FALSE for all non-search commands, and default
2     ! to no matches for SEARCH command.
2     !
2     SEARCH_STATUS = FALSE;
2
2
2     ! Calls OPEN LINE routine to open the Source File pointed by Declare Source
2     ! File DST Command in the given directory.
2     !
2     DBG$SRC_OPEN_FILE(.SRC_FILE,SRC_FILE_CNTRL_BLK,.STEP_FLAG);
2
2
2     ! Calls READ SOURCE FILE routine to read in Source File Records.
2     !
2     DBG$SRC_READ_FILE(.SRC_FILE_CNTRL_BLK,.REC_NUM,BUFPTR,BUFLEN);
2
2
2     ! Call SEARCH routine to search the desried string if SEARCH command is in
2     ! action.  If we did not find the matched string, return the status to
2     ! TYPE LNUM SOURCE, otherwise call OUTPUT LINE routine to output the line.
2     !
2     IF .DBG$SRC_SEARCH_FLAG NEQ 0
2     THEN
3         BEGIN
3         SEARCH_STATUS = DBG$SRC_SEARCH(.BUFPTR,.BUFLEN);
3         IF NOT .SEARCH_STATUS THEN RETURN .SEARCH_STATUS;
2         END;
2
2
2     ! Call the OUTPUT_LINE routine to print the source line.  However, if a
2     ! screen display for source lines is active, we call DBG$SCR_SOURCE_LINE
2     ! instead as the output routine.
2     !
2     IF .SOURCE_TO_SCREEN_FLAG
2     THEN
2         DBG$SCR_SOURCE_LINE(.LINE_NUM,
2                 .SRC_FILE[DST$W_SRC_DF_FILEID], .REC_NUM, .BUFLEN, .BUFPTR)
2
2     ELSE
2         DBG$SRC_OUTPUT_LINE(.LINE_NUM,.STMT_NUM,.BUFPTR,.BUFLEN);
```

```
: 3151      3268  2
: 3152      3269  2     IF .LINE_NUM NEQ -1 THEN DBG$SRC_NEXT_LNUM = .LINE_NUM + 1;
: 3153      3270  2     IF .STMT_NUM NEQ 0 THEN DBG$SRC_NEXT_STMT = .STMT_NUM + 1;
: 3154      3271  2     DBG$SRC_NEXT_MODRSTPTR = .DBG$SRC_MODRSTPTR;
: 3155      3272  2
: 3156      3273  2     RETURN .SEARCH_STATUS;
: 3157      3274  2
: 3158      3275  1     END;


                                                       .EXTRN  DBG$GV_CONTROL

                                   007C 00000 DBG$SRC_TYPE_LINE:
                                                       .WORD   Save R2,R3,R4,R5,R6                        ; 3154
              56 00000000G    00   9E 00002            MOVAB   LIB$SIGNAL, R6
              55 00000000'    EF   9E 00009            MOVAB   DBG$SRC_NEXT_LNUM, R5
              54 00000000'    EF   9E 00010            MOVAB   DBG$SRC_SEARCH_FLAG, R4
              5E              0C   C2 00017            SUBL2   #12, SP
              52         0C   AC   D0 0001A            MOVL    SRC_FILE, R2                               ; 3220
                              09   12 0001E            BNEQ    1$
                 0002832A     8F   DD 00020            PUSHL   #164650
              66              01   FB 00026            CALLS   #1, LIB$SIGNAL
      09 00000000G    00      01   E1 00029 1$:        BBC     #1, DBG$GV_CONTROL+1, 2$
                 000280E8     8F   DD 00031            PUSHL   #164072
              66              01   FB 00037            CALLS   #1, LIB$SIGNAL
                              53   D4 0003A 2$:        CLRL    SEARCH_STATUS                              ; 3231
                         14   AC   DD 0003C            PUSHL   STEP_FLAG                                  ; 3237
                         04   AE   9F 0003F            PUSHAB  SRC_FILE_CNTRL_BLK
                              52   DD 00042            PUSHL   R2
         F39B  CF             03   FB 00044            CALLS   #3, DBG$SRC_OPEN_FILE
                         04   AE   9F 00049            PUSHAB  BUFLEN                                     ; 3242
                         0C   AE   9F 0004C            PUSHAB  BUFPTR
                         10   AC   DD 0004F            PUSHL   REC_NUM
                         0C   AE   DD 00052            PUSHL   SRC_FILE_CNTRL_BLK
         FA3D  CF             04   FB 00055            CALLS   #4, DBG$SRC_READ_FILE
                              64   D5 0005A            TSTL    DBG$SRC_SEARCH_FLAG                        ; 3249
                              11   13 0005C            BEQL    3$
                         04   AE   DD 0005E            PUSHL   BUFLEN                                     ; 3252
                         0C   AE   DD 00061            PUSHL   BUFPTR
         FC63  CF             02   FB 00064            CALLS   #2, DBG$SRC_SEARCH
               53             50   D0 00069            MOVL    R0, SEARCH_STATUS
               4C             53   E9 0006C            BLBC    SEARCH_STATUS, 8$                          ; 3253
               19      0108   C4   E9 0006F 3$:        BLBC    SOURCE_TO_SCREEN_FLAG, 4$                  ; 3261
                         08   AE   DD 00074            PUSHL   BUFPTR                                     ; 3264
                         08   AE   DD 00077            PUSHL   BUFLEN
                         10   AC   DD 0007A            PUSHL   REC_NUM
               7E        03   A2   3C 0007D            MOVZWL  3(R2), -(SP)
                         04   AC   DD 00081            PUSHL   LINE_NUM                                   ; 3263
      00000000G   00          05   FB 00084            CALLS   #5, DBG$SCR_SOURCE_LINE
                              0F   11 0008B            BRB     5$
                         04   AE   DD 0008D 4$:        PUSHL   BUFLEN                                     ; 3267
                         0C   AE   DD 00090            PUSHL   BUFPTR
               7E        04   AC   7D 00093            MOVQ    LINE_NUM, -(SP)
         F829  CF             04   FB 00097            CALLS   #4, DBG$SRC_OUTPUT_LINE
     FFFFFFFF  8F        04   AC   D1 0009C 5$:        CMPL    LINE_NUM, #-1                              ; 3269
                              05   13 000A4            BEQL    6$
```

```
         65       04   AC            01  C1 000A6        ADDL3   #1, LINE_NUM, DBG$SRC_NEXT_LNUM              ;
                            08       AC  D5 000AB 6$:    TSTL    STMT_NUM                                     ; 3270
                                     06  13 000AE        BEQL    7$                                           ;
     08  A5       08   AC            01  C1 000B0        ADDL3   #1, STMT_NUM, DBG$SRC_NEXT_STMT              ;
                  04   A5   FC       A4  D0 000B6 7$:    MOVL    DBG$SRC_MODRSTPTR, DBG$SRC_NEXT_MODRSTPTR    ; 3271
                       50            53  D0 000BB 8$:    MOVL    SEARCH_STATUS, R0                            ; 3273
                                         04 000BE        RET                                                  ; 3275

; Routine Size:  191 bytes,    Routine Base:  DBG$CODE + 144A
```

```
GLOBAL ROUTINE DBG$SRC_TYPE_LNUM_SOURCE(MODRSTPTR, LOWLNUM, LOWSTMT,
                                        HIGHLNUM, HIGHSTMT,
                                        SEARCH_FLAG, STEP_FLAG): NOVALUE =
!
! FUNCTION
!       This routine accepts a Module RST Entry pointer and a line number range
!       and types out all source lines, or, ALL matched strings or NEXT matched
!       string for SEARCH command in that range from that module.  It is
!       called to handle source line display in response to the TYPE command,
!       the EXAMINE/SOURCE command, the SEARCH command, breakpoints, watchpoints,
!       and stepping by line.  In fact, all source line displays go through
!       this routine.
!
!       More specifically, this routine scans through all the Source Line Corre-
!       lation DST Records for the specified module to find the desired line
!       numbers.  It executes all the commands in these DST records as it goes
!       along, and is thus able to determine the appropriate File ID and record
!       number within the file for each desired line number.  For each such line
!       it then calls DBG$SRC_TYPE_LINE to actually retrieve, or search, and/or
!       display the text of the line.
!
!       It handles Declare Source File commands in the DST records by building a
!       File ID Table for the current module.  This table consists of a singly
!       linked list where each entry corresponds to one source file contributing
!       to the current module.  The file's File ID and a pointer to its Declare
!       Source File command block are stored in each entry.  SRC_FILEID_TBL
!       points to this linked list.  DBG$SRC_MODRSTPTR points to the current
!       module's Module RST Entry, this variable is set by this routine and
!       is later used by DBG$SRC_TYPE_LINE.
!
! INPUTS
!       MODRSTPTR - A pointer to the Module RST Entry of the module in which
!                   the source lines are to be looked up.
!
!       LOWLNUM - The line number of the first source line to be typed out.
!                 This is assumed to be a line number in the MODRSTPTR module.
!
!       LOWSTMT - The statement number, if any, associated with the first
!                 source line to be typed out.  If no statement number should
!                 be associated with the LOWLNUM line, LOWSTMT must be zero.
!
!       HIGHLNUM - The line number of the last source line to be type out.
!                  This too is assumed to be a line in the MODRSTPTR module.
!                  HIGHLNUM must be greater than or equal to LOWLNUM.
!
!       HIGHSTMT - The statement number, if any, associated with the last
!                  source line to be typed out.  If no statement number should
!                  be associated with the HIGHLNUM line, HIGHSTMT must be zero.
!                  If HIGHLNUM equals LOWLNUM, HIGHSTMT must be greater than
!                  or equal to LOWSTMT.
!
!       SEARCH_FLAG - Flag set to non-zero value to indicate the SEARCH command
!                     is active, set to zero to indicate other commands are active.
!
!       STEP_FLAG - Flag set to TRUE to indicate the call is from STEP command.
!                   If the source file is not available we only give the message
!                   once, and suppress it the rest of the time.
```

```
!
! OUTPUTS
!       NONE
!

    BEGIN

    MAP
        MODRSTPTR: REF RST$ENTRY;           ! Pointer to Module RST Entry of module
                                            !       containing desired source lines

    LOCAL
        ALL_DONE_FLAG,                      ! Flag set to TRUE if source operation
                                            !       is all done in screen display
        BUFFER: VECTOR[140,BYTE],           ! Output buffer for signaling
        BUF_DESC: VECTOR[2,LONG],           ! Output buffer string descriptor
        DSTPTR: REF DST$SRC_COMMAND,        ! Pointer to the Source File Correlation
                                            !       Command
        DSTREC_PTR: REF DST$RECORD,         ! Pointer to the Source File Correlation
                                            !       DST record
        HIGH_LINE_NUM,                      ! Place holder for Ending Line Number
        HIGH_LNUM,                          ! Highest line number of range
        HIGH_STMT,                          ! Statement number of of highest line
        LENGTH,                             ! Length of current Source File Correlation
                                            !       DST Record
        LINE_NUM,                           ! The current listing line number
        LOW_LNUM,                           ! Lowest line number of range
        LOW_STMT,                           ! Statement number of lowest line
        MODNAMEPTR: REF VECTOR[,BYTE],      ! Pointer to counted ASCII name of current
                                            !       module
        MODSRCTBL: REF VECTOR[,LONG],       ! Pointer to a list of Source File
                                            !       Correlation DST Record pointers
                                            !       for the current module
        SEARCH_FND,                         ! Used for SEARCH command only.  Flag
                                            !       set to TRUE if the search string
                                            !       is found
        SFIT_PTR: REF SFIT$ENTRY,           ! Pointer to the Source File ID Table
        SRC_FILEID_TBL,                     ! Pointer to a linked list of File ID
                                            !       Table entries for current module
        SRC_DSTPTR,                         ! Pointer to Declare Source File Command
                                            !       in DST Record
        SRC_REC,                            ! The record number in the current source
                                            !       file of the current source line
        STATEMENT_MODE,                     ! Flag set to TRUE if we are currently
                                            !       in Statement Mode
        STMT_NUM,                           ! The current statement number
        TYPE_FLAG;                          ! Flag set to TRUE if TYPE LINE routine
                                            !       has been called at least once



    ! Make sure Module RST pointer is not null.
    !
    IF .MODRSTPTR EQL 0
    THEN
        $DBG_ERROR('DBGSOURCE\DBG$SRC_TYPE_LNUM_SOURCE 10');
```

```
2
2       ! Start by doing some consistency checks to ensure that the line number
2       ! of the last source line is greater than or equal to the line number
2       ! of the first source line.  If no statement number should be associated
2       ! with line number, statement number must be zero.  If HIGHLNUM equals
2       ! LOWLNUM, HIGHSTMT must be greater than or equal to LOWSTMT.  If
2       ! either is wrong, signal an error and give up.
2       !
2       IF (.LOWLNUM GTR .HIGHLNUM) THEN SIGNAL(DBG$_INVSRCLIN);
3       IF (.LOWLNUM EQL .HIGHLNUM) AND (.LOWSTMT GTR .HIGHSTMT)
2       THEN
2           SIGNAL(DBG$_INVSRCLIN);
2
2
2       ! Determine whether the source output from this call should be redirected
2       ! to a screen display.
2       !
2       SOURCE_TO_SCREEN_FLAG = FALSE;
2       IF (.DBG$GL_SCREEN_SOURCE NEQ 0) AND
2          (.SEARCH_FLAG EQL 0)          AND
3          (NOT .STEP_FLAG)
2       THEN
2           SOURCE_TO_SCREEN_FLAG = TRUE;
2
2
2       ! Copy the line number range into local variables.  Then, if this is part
2       ! of a screen output operation, initialize the screen output and adjust
2       ! the line number range according to the size of the screen display to
2       ! receive the source output.
2       !
2       LOW_LNUM = .LOWLNUM;
2       LOW_STMT = .LOWSTMT;
2       HIGH_LNUM = .HIGHLNUM;
2       HIGH_STMT = .HIGHSTMT;
2       IF .SOURCE_TO_SCREEN_FLAG
2       THEN
3           BEGIN
3           DBG$SCR_SOURCE_BEGIN(.LOWLNUM, .HIGHLNUM,
3                           LOW_LNUM, HIGH_LNUM, .MODRSTPTR, ALL_DONE_FLAG);
3           IF .ALL_DONE_FLAG THEN RETURN;
2           END;
2
2
2       ! Initialization.  Assume that Line Number is set to the beginning of the
2       ! listing, Source File Record Number is set to the beginning of the
2       ! source file, no Source File ID is found, Statement Mode is not set,
2       ! no line is yet typed out, and statement number is set to 0.
2       !
2       IF .SEARCH_FLAG EQL 0 THEN DBG$SRC_NEXT_MODRSTPTR = 0;
2       DBG$SRC_SEARCH_FLAG = .SEARCH_FLAG;
2       DBG$SRC_FORMFEED_FLAG = FALSE;
2       LINE_NUM = 1;
2       STMT_NUM = 0;
2       SEARCH_FLAG = FALSE;
2       SEARCH_FND = FALSE;
2       SFIT_PTR = 0;
2       SRC_DSTPTR = 0;
```

```
SRC_REC = 1;
STATEMENT_MODE = FALSE;
SRC_FILEID_TBL = 0;
TYPE_FLAG = FALSE;


! Points to the current modules's Module RST Entry, this is later used
! by DBG$SRC_TYPE_LINE routine.  And also get the Module Name.
!
DBG$SRC_MODRSTPTR = .MODRSTPTR;
DBG$STA_SYMNAME(.MODRSTPTR,MODNAMEPTR);


! Make sure Module Source Table pointer in module's Module RST Entry
! points to a list of Source File DST Record Pointers.
!
IF .MODRSTPTR[RST$L_MODSRCTBL] EQL 0
THEN
    BEGIN
    SIGNAL(DBG$_SRCLINNOT,1,.MODNAMEPTR);
    RETURN;
    END;


! Loop thru Module's Source Line Correlation DST Records.
!
MODSRCTBL = .MODRSTPTR[RST$L_MODSRCTBL];
INCR COUNT FROM 1 TO .MODSRCTBL[0] DO
    BEGIN
    IF (.LINE_NUM GTR .HIGH_LNUM) AND (.STMT_NUM GEQ .HIGH_STMT)
    THEN
        EXITLOOP;


    ! We have found the 1st match string for the SEARCH/IDENT/NEXT or
    ! SEARCH/STRING/NEXT command.  Exit the loop.
    !
    IF .SEARCH_FND AND
       ((.DBG$SRC_SEARCH_FLAG EQL SEARCH_IDENT_NEXT) OR
        (.DBG$SRC_SEARCH_FLAG EQL SEARCH_STRING_NEXT))
    THEN
        EXITLOOP;


    ! Get Source Line Correlation DST Record pointer.
    !
    DSTREC_PTR = .MODSRCTBL[.COUNT];
    IF .DSTREC_PTR[DST$B_TYPE] NEQ DST$K_SOURCE
    THEN
        $DBG_ERROR('DBGSOURCE\DBG$SRC_TYPE_LNUM_SOURCE 20');


    ! Scans thru Source File DST Record to decode the Source File Correlation
    ! Commands.  Pick up the length of this DST record.  Set pointer to the
    ! first command entry.
    !
    LENGTH = .DSTREC_PTR[DST$B_LENGTH];
```

```
DSTPTR = DSTREC_PTR[DST$A_SRC_FIRST_CMD];
WHILE (.DSTPTR = .DSTREC_PTR = 1) NEQ .LENGTH DO
    BEGIN
    IF (.LINE_NUM GTR .HIGH_LNUM) AND (.STMT_NUM GEQ .HIGH_STMT)
    THEN
        EXITLOOP;


    ! We have found the 1st match string for SEARCH/IDENT/NEXT or
    ! SEARCH/STRING/NEXT command.  Exit the loop.
    !
    IF .SEARCH_FND AND
       ((.DBG$SRC_SEARCH_FLAG EQL SEARCH_IDENT_NEXT) OR
        (.DBG$SRC_SEARCH_FLAG EQL SEARCH_STRING_NEXT))
    THEN
        EXITLOOP;


    ! Pick up the command code defined in Source File DST Record
    ! and use it as a CASE index so each significant command code
    ! can be interpreted individually.
    !
    CASE .DSTPTR[DST$B_SRC_COMMAND] FROM DST$K_SRC_MIN_CMD TO
                                         DST$K_SRC_MAX_CMD OF
        SET


        ! Declare Source File Command.  Allocate some space to build
        ! the File ID Table.
        !
        [DST$K_SRC_DECLFILE]:
            BEGIN
            IF .SFIT_PTR NEQ 0 THEN SFIT_PTR[SFIT$L_CURRECNUM] = .SRC_REC;
            SRC_REC = 1;
            SFIT_PTR = DBG$GET_TEMPMEM(SFIT$K_SIZE);
            SFIT_PTR[SFIT$L_FLINK] = .SRC_FILEID_TBL;
            SFIT_PTR[SFIT$L_FILE_ID] = .DSTPTR[DST$W_SRC_DF_FILEID];
            SFIT_PTR[SFIT$L_DSTPTR] = .DSTPTR;
            SFIT_PTR[SFIT$L_CURRECNUM] = .SRC_REC;
            SRC_FILEID_TBL = .SFIT_PTR;
            DSTPTR = .DSTPTR + .DSTPTR[DST$B_SRC_DF_LENGTH] + 2;
            END;


        ! Set Source File Command.  This command sets the current
        ! source file to the file denoted by the File ID given in
        ! the command.  This File ID must be pre-defined by the
        ! Declare Source File Command.
        !
        [DST$K_SRC_SETFILE]:
            BEGIN
            IF .SFIT_PTR NEQ 0 THEN SFIT_PTR[SFIT$L_CURRECNUM] = .SRC_REC;
            SRC_DSTPTR = 0;
            SFIT_PTR = .SRC_FILEID_TBL;
            WHILE .SFIT_PTR NEQ 0 DO
                BEGIN
                IF .DSTPTR[DST$W_SRC_UNSWORD] EQL .SFIT_PTR[SFIT$L_FILE_ID]
```

```
                THEN
                    BEGIN
                    SRC_DSTPTR = .SFIT_PTR[SFIT$L_DSTPTR];
                    SRC_REC = .SFIT_PTR[SFIT$L_CURRECNUM];
                    EXITLOOP;
                    END;

                SFIT_PTR = .SFIT_PTR[SFIT$L_FLINK];
                END;


            ! If this File ID has not yet been declared, signal an
            ! invalid DST Record.
            !
            IF .SRC_DSTPTR EQL 0 THEN SIGNAL(DBG$_INVDSTREC);
            DSTPTR = .DSTPTR + 3;
            END;


        ! Set Source Record Number Long Command.  Set the current source
        ! file record number.  Advance DSTPTR.
        !
        [DST$K_SRC_SETREC_L]:
            BEGIN
            SRC_REC = .DSTPTR[DST$L_SRC_UNSLONG];
            DSTPTR = .DSTPTR + 5;
            END;


        ! Set Source Record Number Word Command. (More compact form).
        !
        [DST$K_SRC_SETREC_W]:
            BEGIN
            SRC_REC = .DSTPTR[DST$W_SRC_UNSWORD];
            DSTPTR = .DSTPTR + 3;
            END;


        ! Set Line Number Long Command.  Set the current listing line
        ! number.  Advance DSTPTR.
        !
        [DST$K_SRC_SETLNUM_L]:
            BEGIN
            LINE_NUM = .DSTPTR[DST$L_SRC_UNSLONG];
            DSTPTR = .DSTPTR + 5;
            END;


        ! Set Line Number Word Command. (More compact form).
        !
        [DST$K_SRC_SETLNUM_W]:
            BEGIN
            LINE_NUM = .DSTPTR[DST$W_SRC_UNSWORD];
            DSTPTR = .DSTPTR + 3;
            END;
```

```
        ! Increment Line Number Byte Command.  Increment the current
        ! listing line number by a byte value given in this command.
        ! Advance DSTPTR.
        !
        [DST$K_SRC_INCRLNUM_B]:
            BEGIN
            LINE_NUM = .LINE_NUM + .DSTPTR[DST$B_SRC_UNSBYTE];
            DSTPTR = .DSTPTR + 2;
            END;


        ! Count Form-Feeds as Source Records Command.  This command says
        ! that form-feeds in the source should be counted as separate
        ! source records; the absence of this command says that such
        ! records should be ignored as control information and should
        ! not be assigned line nubmers.
        !
        [DST$K_SRC_FORMFEED]:
            BEGIN
            DBG$SRC_FORMFEED_FLAG = TRUE;
            DSTPTR = .DSTPTR + 1;
            END;


        ! Define N Separate Lines Word Command, and Define N Separate
        ! Lines Byte Command (More Compact Form).  If a Source File is not
        ! set, signal and give up.  Defines the Source File and Source
        ! Record Numbers for a specified number of Listing Line Numbers.
        ! Call TYPE LINE routine, if LOW_LNUM <= LINE_NUM <= HIGH_LNUM.
        !
        [DST$K_SRC_DEFLINES_W, DST$K_SRC_DEFLINES_B]:
            BEGIN
            IF .SRC_DSTPTR EQL 0 THEN SIGNAL(DBG$_INVDSTREC);


            ! Calculate the listing line number range.
            !
            IF .DSTPTR[DST$B_SRC_COMMAND] EQL DST$K_SRC_DEFLINES_W
            THEN
                HIGH_LINE_NUM = .LINE_NUM + .DSTPTR[DST$W_SRC_UNSWORD] - 1
            ELSE
                HIGH_LINE_NUM = .LINE_NUM + .DSTPTR[DST$B_SRC_UNSBYTE] - 1;


            ! Test to see if the listing line number exceeds the type
            ! line range.  If it is, increments the source record number
            ! listing line number and DST pointer, then returns.
            !
            IF (.LINE_NUM GTR .HIGH_LNUM) OR
               (.HIGH_LINE_NUM LSS .LOW_LNUM)
            THEN
                BEGIN
                IF .DSTPTR[DST$B_SRC_COMMAND] EQL DST$K_SRC_DEFLINES_W
                THEN
                    SRC_REC = .SRC_REC + .DSTPTR[DST$W_SRC_UNSWORD]
                ELSE
                    SRC_REC = .SRC_REC + .DSTPTR[DST$B_SRC_UNSBYTE];
```

```
                            END
                        ELSE
                            BEGIN
                            TYPE_FLAG = TRUE;

                            ! Synchronize the Source Record Number with the Listing Line
                            ! Number.
                            !
                            IF (.LOW_LNUM - .LINE_NUM) GTR 0
                            THEN
                                SRC_REC = .SRC_REC + (.LOW_LNUM - .LINE_NUM);

                            INCR I FROM MAX(.LOW_LNUM,.LINE_NUM) TO
                                        MIN(.HIGH_LNUM,.HIGH_LINE_NUM) DO
                                BEGIN
                                SEARCH_FND = DBG$SRC_TYPE_LINE(.I, .STMT_NUM,
                                            .SRC_DSTPTR, .SRC_REC, .STEP_FLAG);
                                IF .SEARCH_FND THEN SEARCH_FLAG = TRUE;
                                SRC_REC = .SRC_REC + 1;
                                IF .SEARCH_FND AND
                                   ((.DBG$SRC_SEARCH_FLAG EQL SEARCH_IDENT_NEXT) OR
                                    (.DBG$SRC_SEARCH_FLAG EQL SEARCH_STRING_NEXT))
                                THEN
                                    EXITLOOP;

                                END;

                            END;

                        LINE_NUM = .HIGH_LINE_NUM + 1;
                        IF .DSTPTR[DST$B_SRC_COMMAND] EQL DST$K_SRC_DEFLINES_W
                        THEN
                            DSTPTR = .DSTPTR + 3
                        ELSE
                            DSTPTR = .DSTPTR + 2;

                        END;


                    ! Any other command code constitutes an error in the DST.
                    !
                    [INRANGE, OUTRANGE]:
                        SIGNAL(DBG$_INVDSTREC);

                    TES;

                END;                            ! End of WHILE Loop through commands.

            END;                                ! End of INCR Loop through DST records.

        ! If we were outputting source lines to a screen display, close out that
        ! screen display properly.
        !
```

```
IF .SOURCE_TO_SCREEN_FLAG THEN DBG$SCR_SOURCE_END();


! Issue a warning message if no lines were typed out.  This situation
! occurs if the specified line numbers do not exist in this module.
! If it is in the search mode, signal no matches instead.
!
IF NOT .TYPE_FLAG
THEN
    IF .DBG$SRC_SEARCH_FLAG EQL 0
    THEN
        BEGIN
        BUF_DESC[0] = 139;
        BUF_DESC[1] = BUFFER[1];
        IF (.LOWLNUM EQL .HIGHLNUM) AND (.LOWSTMT EQL .HIGHSTMT)
        THEN
            DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC 'line '), 0)
        ELSE
            DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC 'lines '), 0);

        DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC '!UL'), .LOWLNUM);
        IF .LOWSTMT NEQ 0
        THEN
            DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC '.!UL'), .LOWSTMT);

        IF (.LOWLNUM NEQ .HIGHLNUM) OR (.LOWSTMT NEQ .HIGHSTMT)
        THEN
            DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC ':!UL'), .HIGHLNUM);

        IF .HIGHSTMT NEQ 0
        THEN
            DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC '.!UL'), .HIGHSTMT);

        IF (.LOWLNUM EQL .HIGHLNUM) AND (.LOWSTMT EQL .HIGHSTMT)
        THEN
            DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC ' does'), 0)
        ELSE
            DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC ' do'), 0);

        DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC ' not exist in module !AC'),
                           .MODNAMEPTR);
        BUFFER[0] = 139 - .BUF_DESC[0];
        SIGNAL(DBG$_NOLINXXX, 1, BUFFER);
        END


    ! If we were in a SEARCH command and output no lines, signal no matches.
    !
    ELSE
        SIGNAL(DBG$_NOMATCH);


! Check to see if any matches were found for the SEARCH command.
!
IF .TYPE_FLAG AND NOT .SEARCH_FLAG AND .DBG$SRC_SEARCH_FLAG NEQ 0
THEN
    SIGNAL(DBG$_NOMATCH);
```

```
; 3673          3789  2
; 3674          3790  2       RETURN;
; 3675          3791  2
; 3676          3792  1       END;


                                                                   .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

24 47 42 44 5C 45 43 52 55 4F 53 47 42 44 25 0022F P.ABE:  .ASCII  \%DBGSOURCE\<92>\DBG$SRC_TYPE_LNUM_SOURC\  ;
53 5F 4D 55 4E 4C 5F 45 50 59 54 5F 43 52 53 0023E                                                          ;
                                     43 52 55 4F 0024D                                                          ;
                                     30 31 20 45 00251         .ASCII  \E 10\                                 ;
24 47 42 44 5C 45 43 52 55 4F 53 47 42 44 25 00255 P.ABF:  .ASCII  \%DBGSOURCE\<92>\DBG$SRC_TYPE_LNUM_SOURC\  ;
53 5F 4D 55 4E 4C 5F 45 50 59 54 5F 43 52 53 00264                                                          ;
                                     43 52 55 4F 00273                                                          ;
                                     30 32 20 45 00277         .ASCII  \E 20\                                 ;
                            20 65 6E 69 6C 05 0027B P.ABG:  .ASCII  <5>\line \                             ;
                         20 73 65 6E 69 6C 06 00281 P.ABH:  .ASCII  <6>\lines \                            ;
                                     4C 55 21 03 00288 P.ABI:  .ASCII  <3>\!UL\                              ;
                                  4C 55 21 2E 04 0028C P.ABJ:  .ASCII  <4>\.!UL\                             ;
                                  4C 55 21 3A 04 00291 P.ABK:  .ASCII  <4>\:!UL\                             ;
                                  4C 55 21 2E 04 00296 P.ABL:  .ASCII  <4>\.!UL\                             ;
                               73 65 6F 64 20 05 0029B P.ABM:  .ASCII  <5>\ does\                            ;
                                     6F 64 20 03 002A1 P.ABN:  .ASCII  <3>\ do\                              ;
20 6E 69 20 74 73 69 78 65 20 74 6F 6E 20 18 002A5 P.ABO:  .ASCII  <24>\ not exist in module !AC\          ;
            43 41 21 20 65 6C 75 64 6F 6D 002B4                                                          ;


                                                                   .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                    0FFC 00000         .ENTRY  DBG$SRC_TYPE_LNUM_SOURCE, Save R2,R3,R4,R5,-; 3276
                                                               R6,R7,R8,R9,R10,R11
                     5E     FF30  CE  9E 00002         MOVAB   -208(SP), SP                                   ;
                     52       04  AC  D0 00007         MOVL    MODRSTPTR, R2                                  ; 3386
                                  15  12 0000B         BNEQ    1$                                             ;
                         00000000'  EF  9F 0000D         PUSHAB  P.ABE                                          ; 3388
                                  01  DD 00013         PUSHL   #1                                             ;
                         00028362  8F  DD 00015         PUSHL   #164706                                        ;
            00000000G  00          03  FB 0001B         CALLS   #3, LIB$SIGNAL                                 ;
                   10  AC     08  AC  D1 00022 1$:     CMPL    LOWLNUM, HIGHLNUM                              ; 3398
                                  0D  15 00027         BLEQ    2$                                             ;
                         00028CC0  8F  DD 00029         PUSHL   #167104                                        ;
            00000000G  00          01  FB 0002F         CALLS   #1, LIB$SIGNAL                                 ;
                              28  AE  D4 00036 2$:     CLRL    40(SP)                                         ; 3399
                   10  AC     08  AC  D1 00039         CMPL    LOWLNUM, HIGHLNUM                              ;
                                  17  12 0003E         BNEQ    3$                                             ;
                              28  AE  D6 00040         INCL    40(SP)                                         ;
                   14  AC     0C  AC  D1 00043         CMPL    LOWSTMT, HIGHSTMT                              ;
                                  0D  15 00048         BLEQ    3$                                             ;
                         00028CC0  8F  DD 0004A         PUSHL   #167104                                        ; 3401
            00000000G  00          01  FB 00050         CALLS   #1, LIB$SIGNAL                                 ;
                         00000000'  EF  D4 00057 3$:     CLRL    SOURCE_TO_SCREEN_FLAG                          ; 3407
                         00000000G  00  D5 0005D         TSTL    DBG$GL_SCREEN_SOURCE                           ; 3408
                                  10  13 00063         BEQL    4$                                             ;
                              18  AC  D5 00065         TSTL    SEARCH_FLAG                                    ; 3409
```

```
                                  0B  12 00068          BNEQ    4$                                              ;
                  07         1C   AC  E8 0006A          BLBS    STEP_FLAG, 4$                                   ; 3410
        00000000' EF              01  D0 0006E          MOVL    #1, SOURCE_TO_SCREEN_FLAG                       ; 3412
              34  AE         08   AC  D0 00075 4$:      MOVL    LOWLNUM, LOW_LNUM                               ; 3420
                  5B         0C   AC  D0 0007A          MOVL    LOWSTMT, R11                                    ; 3421
                  50              5B  D0 0007E          MOVL    R11, LOW_STMT                                   ;
              30  AE         10   AC  D0 00081          MOVL    HIGHLNUM, HIGH_LNUM                             ; 3422
                  5A         14   AC  D0 00086          MOVL    HIGHSTMT, R10                                   ; 3423
              20  AE              5A  D0 0008A          MOVL    R10, HIGH_STMT                                  ;
                  1D 00000000'    EF  E9 0008E          BLBC    SOURCE_TO_SCREEN_FLAG, 5$                       ; 3424
                             2C   AE  9F 00095          PUSHAB  ALL_DONE_FLAG                                   ; 3427
                                  52  DD 00098          PUSHL   R2                                              ; 3428
                             38   AE  9F 0009A          PUSHAB  HIGH_LNUM                                       ; 3427
                             40   AE  9F 0009D          PUSHAB  LOW_LNUM                                        ;
                             10   AC  DD 000A0          PUSHL   HIGHLNUM                                        ;
                             08   AC  DD 000A3          PUSHL   LOWLNUM                                         ;
        00000000G 00              06  FB 000A6          CALLS   #6, DBG$SCR_SOURCE_BEGIN                        ;
                  01         2C   AE  E9 000AD          BLBC    ALL_DONE_FLAG, 5$                               ; 3429
                                  04 000B1              RET                                                     ;
                             18   AC  D5 000B2 5$:      TSTL    SEARCH_FLAG                                     ; 3438
                                  06  12 000B5          BNEQ    6$                                              ;
                      00000000'   EF  D4 000B7          CLRL    DBG$SRC_NEXT_MODRSTPTR                          ;
        00000000' EF         18   AC  D0 000BD 6$:      MOVL    SEARCH_FLAG, DBG$SRC_SEARCH_FLAG                ; 3439
                      00000000'   EF  D4 000C5          CLRL    DBG$SRC_FORMFEED_FLAG                           ; 3440
                  54              01  D0 000CB          MOVL    #1, LINE_NUM                                    ; 3441
                             18   AC  D4 000CE          CLRL    SEARCH_FLAG                                     ; 3443
                             14   AE  7C 000D1          CLRQ    SEARCH_FND                                      ; 3444
                                  53  D4 000D4          CLRL    SFIT_PTR                                        ; 3445
                                  6E  D4 000D6          CLRL    SRC_DSTPTR                                      ; 3446
                  56              01  D0 000D8          MOVL    #1, SRC_REC                                     ; 3447
                                  50  D4 000DB          CLRL    STATEMENT_MODE                                  ; 3448
                             0C   AE  7C 000DD          CLRQ    STMT_NUM                                        ; 3442
        00000000' EF              52  D0 000E0          MOVL    R2, DBG$SRC_MODRSTPTR                           ; 3456
                             38   AE  9F 000E7          PUSHAB  MODNAMEPTR                                      ; 3457
                                  52  DD 000EA          PUSHL   R2                                              ;
        00000000G 00              02  FB 000EC          CALLS   #2, DBG$STA_SYMNAME                             ;
                             24   A2  D5 000F3          TSTL    36(R2)                                          ; 3463
                                  13  12 000F6          BNEQ    7$                                              ;
                             38   AE  DD 000F8          PUSHL   MODNAMEPTR                                      ; 3466
                                  01  DD 000FB          PUSHL   #1                                              ;
                      00028CB8    8F  DD 000FD          PUSHL   #167096                                         ;
        00000000G 00              03  FB 00103          CALLS   #3, LIB$SIGNAL                                  ;
                                  04 0010A              RET                                                     ; 3465
              1C  AE         24   A2  D0 0010B 7$:      MOVL    36(R2), MODSRCTBL                               ; 3473
                                  58  D4 00110          CLRL    COUNT                                           ; 3476
                                0223  31 00112 8$:      BRW     53$                                             ;
              30  AE              54  D1 00115 9$:      CMPL    LINE_NUM, HIGH_LNUM                             ;
                                  0A  15 00119          BLEQ    11$                                             ;
              20  AE         0C   AE  D1 0011B          CMPL    STMT_NUM, HIGH_STMT                             ;
                                  03  19 00120          BLSS    11$                                             ;
                                021A  31 00122 10$:     BRW     54$                                             ;
                  12         14   AE  E9 00125 11$:     BLBC    SEARCH_FND, 12$                                 ; 3484
                  02 00000000'    EF  D1 00129          CMPL    DBG$SRC_SEARCH_FLAG, #2                         ; 3485
                                  F0  13 00130          BEQL    10$                                             ;
                  04 00000000'    EF  D1 00132          CMPL    DBG$SRC_SEARCH_FLAG, #4                         ; 3486
                                  E7  13 00139          BEQL    10$                                             ;
                  59         1C BE48  D0 0013B 12$:     MOVL    @MODSRCTBL[COUNT], DSTREC_PTR                   ; 3493
```

```
                 9B   8F         01  A9  91 00140       CMPB    1(DSTREC_PTR), #155                 ; 3494
                                 15  13 00145           BEQL    13$                                 ;
                         00000000'   EF  9F 00147       PUSHAB  P.ABF                               ; 3496
                                 01  DD 0014D           PUSHL   #1                                  ;
                         00028362    8F  DD 0014F       PUSHL   #164706                             ;
          00000000G   00         03  FB 00155           CALLS   #3, LIB$SIGNAL                      ;
                 24   AE         69  9A 0015C  13$:     MOVZBL  (DSTREC_PTR), LENGTH                ; 3503
                      52     02  A9  9E 00160           MOVAB   2(R9), DSTPTR                       ; 3504
          50          52         59  C3 00164  14$:     SUBL3   DSTREC_PTR, DSTPTR, R0              ; 3505
                                 50  D7 00168           DECL    R0                                  ;
                 24   AE         50  D1 0016A           CMPL    R0, LENGTH                          ;
                                 A2  13 0016E           BEQL    8$                                  ;
                                 55  D4 00170           CLRL    R5                                  ; 3507
                 30   AE         54  D1 00172           CMPL    LINE_NUM, HIGH_LNUM                 ;
                                 09  15 00176           BLEQ    15$                                 ;
                                 55  D6 00178           INCL    R5                                  ;
                 20   AE     0C  AE  D1 0017A           CMPL    STMT_NUM, HIGH_STMT                 ;
                                 91  18 0017F           BGEQ    8$                                  ;
                      15     14  AE  E9 00181  15$:     BLBC    SEARCH_FND, 16$                     ; 3515
                      02 00000000'   EF  D1 00185       CMPL    DBG$SRC_SEARCH_FLAG, #2             ; 3516
                                 84  13 0018C           BEQL    8$                                  ;
                      04 00000000'   EF  D1 0018E       CMPL    DBG$SRC_SEARCH_FLAG, #4             ; 3517
                                 03  12 00195           BNEQ    16$                                 ;
                               019E  31 00197           BRW     53$                                 ;
               0F     01         62  8F 0019A  16$:     CASEB   (DSTPTR), #1, #15                   ; 3526
   00A7      00A1   0066       002F     0019E  17$:     .WORD   19$-17$,-                           ;
   0020      00BD   00B6       00AD     001A6                   21$-17$,-                           ;
   0020      00D3   00D3       0020     001AE                   26$-17$,-                           ;
   00C7      0020   0020       0020     001B6                   27$-17$,-                           ;
                                                                28$-17$,-                           ;
                                                                30$-17$,-                           ;
                                                                32$-17$,-                           ;
                                                                18$-17$,-                           ;
                                                                18$-17$,-                           ;
                                                                35$-17$,-                           ;
                                                                35$-17$,-                           ;
                                                                18$-17$,-                           ;
                                                                18$-17$,-                           ;
                                                                18$-17$,-                           ;
                                                                18$-17$,-                           ;
                                                                33$-17$                             ;
                         0002832A    8F  DD 001BE  18$:     PUSHL   #164650                         ; 3720
          00000000G   00         01  FB 001C4           CALLS   #1, LIB$SIGNAL                      ;
                                 97  11 001CB           BRB     14$                                 ;
                                 53  D5 001CD  19$:     TSTL    SFIT_PTR                            ; 3536
                                 04  13 001CF           BEQL    20$                                 ;
                 0C   A3         56  D0 001D1           MOVL    SRC_REC, 12(SFIT_PTR)               ;
                      56         01  D0 001D5  20$:     MOVL    #1, SRC_REC                         ; 3537
                                 04  DD 001D8           PUSHL   #4                                  ; 3538
          00000000G   00         01  FB 001DA           CALLS   #1, DBG$GET_TEMPMEM                 ;
                      53         50  D0 001E1           MOVL    R0, SFIT_PTR                        ;
                      63     10  AE  D0 001E4           MOVL    SRC_FILEID_TBL, (SFIT_PTR)          ; 3539
                 04   A3     03  A2  3C 001E8           MOVZWL  3(DSTPTR), 4(SFIT_PTR)              ; 3540
                 08   A3         52  D0 001ED           MOVL    DSTPTR, 8(SFIT_PTR)                 ; 3541
                 0C   A3         56  D0 001F1           MOVL    SRC_REC, 12(SFIT_PTR)               ; 3542
                 10   AE         53  D0 001F5           MOVL    SFIT_PTR, SRC_FILEID_TBL            ; 3543
                      50     01  A2  9A 001F9           MOVZBL  1(DSTPTR), R0                       ; 3544
```

```
                    52       02 A042  9E 001FD        MOVAB   2(R0)[DSTPTR], DSTPTR                    ;
                                 6A  11 00202        BRB     34$                                      ; 3526
                                 53  D5 00204 21$:   TSTL    SFIT_PTR                                 ; 3555
                                 04  13 00206        BEQL    22$                                      ;
              0C    A3           56  D0 00208        MOVL    SRC_REC, 12(SFIT_PTR)                    ;
                                 6E  D4 0020C 22$:   CLRL    SRC_DSTPTR                               ; 3556
                    53       10  AE  D0 0020E        MOVL    SRC_FILEID_TBL, SFIT_PTR                 ; 3557
                                 18  13 00212 23$:   BEQL    25$                                      ; 3558
  04   A3    01   A2         10  00  ED 00214        CMPZV   #0, #16, 1(DSTPTR), 4(SFIT_PTR)          ; 3560
                                 0A  12 0021B        BNEQ    24$                                      ;
                    6E       08  A3  D0 0021D        MOVL    8(SFIT_PTR), SRC_DSTPTR                  ; 3563
                    56       0C  A3  D0 00221        MOVL    12(SFIT_PTR), SRC_REC                    ; 3564
                                 05  11 00225        BRB     25$                                      ; 3562
                    53           63  D0 00227 24$:   MOVL    (SFIT_PTR), SFIT_PTR                     ; 3568
                                 E6  11 0022A        BRB     23$                                      ; 3558
                                 6E  D5 0022C 25$:   TSTL    SRC_DSTPTR                               ; 3575
                                 28  12 0022E        BNEQ    31$                                      ;
                        0002832A 8F  DD 00230        PUSHL   #164650                                  ;
           00000000G  00         01  FB 00236        CALLS   #1, LIB$SIGNAL                           ;
                                 19  11 0023D        BRB     31$                                      ; 3576
                    56       01  A2  D0 0023F 26$:   MOVL    1(DSTPTR), SRC_REC                       ; 3585
                                 0A  11 00243        BRB     29$                                      ; 3586
                    56       01  A2  3C 00245 27$:   MOVZWL  1(DSTPTR), SRC_REC                       ; 3594
                                 0D  11 00249        BRB     31$                                      ; 3595
                    54       01  A2  D0 0024B 28$:   MOVL    1(DSTPTR), LINE_NUM                      ; 3604
                    52           05  C0 0024F 29$:   ADDL2   #5, DSTPTR                               ; 3605
                                 1A  11 00252        BRB     34$                                      ; 3526
                    54       01  A2  3C 00254 30$:   MOVZWL  1(DSTPTR), LINE_NUM                      ; 3613
                               00D2  31 00258 31$:   BRW     50$                                      ; 3614
                    50       01  A2  9A 0025B 32$:   MOVZBL  1(DSTPTR), R0                            ; 3624
                    54           50  C0 0025F        ADDL2   R0, LINE_NUM                             ;
                               00CD  31 00262        BRW     51$                                      ; 3625
           00000000'  EF         01  D0 00265 33$:   MOVL    #1, DBG$SRC_FORMFEED_FLAG                ; 3637
                                 52  D6 0026C        INCL    DSTPTR                                   ; 3638
                               FEF3  31 0026E 34$:   BRW     14$                                      ; 3526
                                 6E  D5 00271 35$:   TSTL    SRC_DSTPTR                               ; 3650
                                 0D  12 00273        BNEQ    36$                                      ;
                        0002832A 8F  DD 00275        PUSHL   #164650                                  ;
           00000000G  00         01  FB 0027B        CALLS   #1, LIB$SIGNAL                           ;
                    51       01  A2  9E 00282 36$:   MOVAB   1(DSTPTR), R1                            ; 3657
                             08  AE  D4 00286        CLRL    8(SP)                                    ; 3655
                    0A           62  91 00289        CMPB    (DSTPTR), #10                            ;
                                 08  12 0028C        BNEQ    37$                                      ;
                             08  AE  D6 0028E        INCL    8(SP)                                    ;
                    50           61  3C 00291        MOVZWL  (R1), R0                                 ; 3657
                                 03  11 00294        BRB     38$                                      ;
                    50           61  9A 00296 37$:   MOVZBL  (R1), R0                                 ; 3659
              04    AE       FF A044  9E 00299 38$:   MOVAB   -1(R0)[LINE_NUM], HIGH_LINE_NUM          ;
                    07           55  E8 0029F        BLBS    R5, 39$                                  ; 3666
              34    AE       04  AE  D1 002A2        CMPL    HIGH_LINE_NUM, LOW_LNUM                  ; 3667
                                 11  18 002A7        BGEQ    42$                                      ;
                    05       08  AE  E9 002A9 39$:   BLBC    8(SP), 40$                               ; 3670
                    50           61  3C 002AD        MOVZWL  (R1), R0                                 ; 3672
                                 03  11 002B0        BRB     41$                                      ;
                    50           61  9A 002B2 40$:   MOVZBL  (R1), R0                                 ; 3674
                    56           50  C0 002B5 41$:   ADDL2   R0, SRC_REC                              ;
                                 6A  11 002B8        BRB     49$                                      ; 3666
```

```
              18  AE           01  D0 002BA 42$:   MOVL    #1, TYPE_FLAG                         ; 3680
                  54      34   AE  D1 002BE        CMPL    LOW_LNUM, LINE_NUM                    ; 3686
                               08  15 002C2        BLEQ    43$                                   ;
          50      34  AE       54  C3 002C4        SUBL3   LINE_NUM, LOW_LNUM, R0                ; 3688
                  56           50  C0 002C9        ADDL2   R0, SRC_REC                           ;
                  55      34   AE  D0 002CC 43$:   MOVL    LOW_LNUM, R5                          ; 3690
                  54           55  D1 002D0        CMPL    R5, LINE_NUM                          ;
                               03  18 002D3        BGEQ    44$                                   ;
                  55           54  D0 002D5        MOVL    LINE_NUM, R5                          ;
                  57      30   AE  D0 002D8 44$:   MOVL    HIGH_LNUM, R7                         ; 3691
              04  AE           57  D1 002DC        CMPL    R7, HIGH_LINE_NUM                     ;
                               04  15 002E0        BLEQ    45$                                   ;
              57  04           AE  D0 002E2        MOVL    HIGH_LINE_NUM, R7                     ;
                               55  D7 002E6 45$:   DECL    I                                     ; 3693
                               36  11 002E8        BRB     48$                                   ;
                          1C   AC  DD 002EA 46$:   PUSHL   STEP_FLAG                             ; 3694
                               56  DD 002ED        PUSHL   SRC_REC                               ;
                          08   AE  DD 002EF        PUSHL   SRC_DSTPTR                            ;
                          18   AE  DD 002F2        PUSHL   STMT_NUM                              ; 3693
                               55  DD 002F5        PUSHL   I                                     ;
            FC45  CF           05  FB 002F7        CALLS   #5, DBG$SRC_TYPE_LINE                 ;
              14  AE           50  D0 002FC        MOVL    R0, SEARCH_FND                        ;
                  04      14   AE  E9 00300        BLBC    SEARCH_FND, 47$                       ; 3695
              18  AC           01  D0 00304        MOVL    #1, SEARCH_FLAG                       ;
                               56  D6 00308 47$:   INCL    SRC_REC                               ; 3696
                  12      14   AE  E9 0030A        BLBC    SEARCH_FND, 48$                       ; 3697
                  02 00000000' EF  D1 0030E        CMPL    DBG$SRC_SEARCH_FLAG, #2               ; 3698
                               0D  13 00315        BEQL    49$                                   ;
                  04 00000000' EF  D1 00317        CMPL    DBG$SRC_SEARCH_FLAG, #4               ; 3699
                               04  13 0031E        BEQL    49$                                   ;
          C6      55           57  F3 00320 48$:   AOBLEQ  R7, I, 46$                            ; 3690
          54      04  AE       01  C1 00324 49$:   ADDL3   #1, HIGH_LINE_NUM, LINE_NUM           ; 3707
                  05      08   AE  E9 00329        BLBC    8(SP), 51$                            ; 3708
                  52           03  C0 0032D 50$:   ADDL2   #3, DSTPTR                            ; 3710
                               03  11 00330        BRB     52$                                   ;
                  52           02  C0 00332 51$:   ADDL2   #2, DSTPTR                            ; 3712
                             FE2C  31 00335 52$:   BRW     14$                                   ; 3526
    FDD6          58      01  1C   BE  F1 00338 53$:   ACBL    @MODSRCTBL, #1, COUNT, 9$         ; 3474
                  07 00000000' EF  E9 0033F 54$:   BLBC    SOURCE_TO_SCREEN_FLAG, 55$            ; 3732
       00000000G  00           00  FB 00346        CALLS   #0, DBG$SCR_SOURCE_END                ;
                  03      18   AE  E9 0034D 55$:   BLBC    TYPE_FLAG, 56$                        ; 3739
                             00FC  31 00351        BRW     68$                                   ;
                     00000000' EF  D5 00354 56$:   TSTL    DBG$SRC_SEARCH_FLAG                   ; 3741
                               03  13 0035A        BEQL    57$                                   ;
                             00E0  31 0035C        BRW     66$                                   ;
              3C  AE      8B   8F  9A 0035F 57$:   MOVZBL  #139, BUF_DESC                        ; 3744
              40  AE      45   AE  9E 00364        MOVAB   BUFFER+1, BUF_DESC+4                  ; 3745
                  0F      28   AE  E9 00369        BLBC    40(SP), 58$                           ; 3746
                  5A           5B  D1 0036D        CMPL    R11, R10                              ;
                               0A  12 00370        BNEQ    58$                                   ;
                               7E  D4 00372        CLRL    -(SP)                                 ; 3748
                     00000000' EF  9F 00374        PUSHAB  P.ABG                                 ;
                               08  11 0037A        BRB     59$                                   ;
                               7E  D4 0037C 58$:   CLRL    -(SP)                                 ; 3750
                     00000000' EF  9F 0037E        PUSHAB  P.ABH                                 ;
                          44   AE  9F 00384 59$:   PUSHAB  BUF_DESC                              ;
       00000000G  00           03  FB 00387        CALLS   #3, DBG$FORMAT_FAO_OUT                ;
```

```
                         08   AC   DD 0038E         PUSHL   LOWLNUM                         ; 3752
                   00000000'  EF   9F 00391         PUSHAB  P.ABI                           ;
                         44   AE   9F 00397         PUSHAB  BUF_DESC                        ;
        00000000G   00        03   FB 0039A         CALLS   #3, DBG$FORMAT_FAO_OUT          ;
                              5B   D5 003A1         TSTL    R11                             ; 3753
                              12   13 003A3         BEQL    60$                             ;
                              5B   DD 003A5         PUSHL   R11                             ; 3755
                   00000000'  EF   9F 003A7         PUSHAB  P.ABJ                           ;
                         44   AE   9F 003AD         PUSHAB  BUF_DESC                        ;
        00000000G   00        03   FB 003B0         CALLS   #3, DBG$FORMAT_FAO_OUT          ;
               10   AC   08   AC   D1 003B7 60$:    CMPL    LOWLNUM, HIGHLNUM               ; 3757
                              05   12 003BC         BNEQ    61$                             ;
                    5A        5B   D1 003BE         CMPL    R11, R10                        ;
                              13   13 003C1         BEQL    62$                             ;
                         10   AC   DD 003C3 61$:    PUSHL   HIGHLNUM                        ; 3759
                   00000000'  EF   9F 003C6         PUSHAB  P.ABK                           ;
                         44   AE   9F 003CC         PUSHAB  BUF_DESC                        ;
        00000000G   00        03   FB 003CF         CALLS   #3, DBG$FORMAT_FAO_OUT          ;
                              5A   D5 003D6 62$:    TSTL    R10                             ; 3761
                              12   13 003D8         BEQL    63$                             ;
                              5A   DD 003DA         PUSHL   R10                             ; 3763
                   00000000'  EF   9F 003DC         PUSHAB  P.ABL                           ;
                         44   AE   9F 003E2         PUSHAB  BUF_DESC                        ;
        00000000G   00        03   FB 003E5         CALLS   #3, DBG$FORMAT_FAO_OUT          ;
                    0F   28   AE   E9 003EC 63$:    BLBC    40(SP), 64$                     ; 3765
                    5A        5B   D1 003F0         CMPL    R11, R10                        ;
                              0A   12 003F3         BNEQ    64$                             ;
                              7E   D4 003F5         CLRL    -(SP)                           ; 3767
                   00000000'  EF   9F 003F7         PUSHAB  P.ABM                           ;
                              08   11 003FD         BRB     65$                             ;
                              7E   D4 003FF 64$:    CLRL    -(SP)                           ; 3769
                   00000000'  EF   9F 00401         PUSHAB  P.ABN                           ;
                         44   AE   9F 00407 65$:    PUSHAB  BUF_DESC                        ;
        00000000G   00        03   FB 0040A         CALLS   #3, DBG$FORMAT_FAO_OUT          ;
                         38   AE   DD 00411         PUSHL   MODNAMEPTR                      ; 3772
                   00000000'  EF   9F 00414         PUSHAB  P.ABO                           ; 3771
                         44   AE   9F 0041A         PUSHAB  BUF_DESC                        ;
        00000000G   00        03   FB 0041D         CALLS   #3, DBG$FORMAT_FAO_OUT          ;
   44   AE     8B   8F   3C   AE   83 00424         SUBB3   BUF_DESC, #139, BUFFER          ; 3773
                         44   AE   9F 0042B         PUSHAB  BUFFER                          ; 3774
                              01   DD 0042E         PUSHL   #1                              ;
                   00028E18   8F   DD 00430         PUSHL   #167448                         ;
        00000000G   00        03   FB 00436         CALLS   #3, LIB$SIGNAL                  ;
                              0D   11 0043D         BRB     67$                             ; 3741
                   00028E20   8F   DD 0043F 66$:    PUSHL   #167456                         ; 3781
        00000000G   00        01   FB 00445         CALLS   #1, LIB$SIGNAL                  ;
                    19   18   AE   E9 0044C 67$:    BLBC    TYPE_FLAG, 69$                  ; 3786
                    15   18   AC   E8 00450 68$:    BLBS    SEARCH_FLAG, 69$                ;
                   00000000'  EF   D5 00454         TSTL    DBG$SRC_SEARCH_FLAG             ;
                              0D   13 0045A         BEQL    69$                             ;
                   00028E20   8F   DD 0045C         PUSHL   #167456                         ; 3788
        00000000G   00        01   FB 00462         CALLS   #1, LIB$SIGNAL                  ;
                                   04 00469 69$:    RET                                     ; 3792

; Routine Size:  1130 bytes,    Routine Base:  DBG$CODE + 1509
```

```
1 GLOBAL ROUTINE DBG$SRC_TYPE_PC_SOURCE(LOW_PC, HIGH_PC, MODPRTFLG, STEP_FLAG): NOVALUE =
1 !
1 ! FUNCTION
1 !       This routine accepts a range of Program Counter values and prints all
1 !       source lines which cover that PC range.  This assumes that both PC
1 !       values are located in the same module and that a line number is associ-
1 !       ated with each end of the PC range.  It also assums that the lower PC
1 !       value is associated with the lower line number of the range.  This
1 !       routine is called in the processing of the EXAMINE/SOURCE command and
1 !       does the source display in response to breakpoints, watchpoints, and
1 !       stepping by lines.
1 !
1 !       The routine calls DBG$PC_TO_LINE_LOOKUP for the two PC values to get the corre-
1 !       sponding line numbers.  If both PC values are the same (which is always
1 !       the case on breakpoints or when stepping), DBG$PC_TO_LINE_LOOKUP is only called
1 !       once.  The routine then checks that both PC values are in the same mod-
1 !       ule and signals an error if they are not.  It also signals an error if
1 !       one or the other PC value does not correspond to a line number or if the
1 !       line numbers they correspond to are not in the proper order.  Routine
1 !       DBG$SRC_TYPE_LNUM_SOURCE is then called to retrieve and type the source
1 !       images.
1 !
1 ! INPUTS
1 !       LOW_PC  - The lower PC value of the address range whose source lines
1 !                 are to be displayed.
1 !
1 !       HIGH_PC - The higher PC value of the address range whose source lines
1 !                 are to be displayed.  HIGH_PC must be greater than or equal
1 !                 to LOW_PC.
1 !
1 !       MODPRTFLG - Flag set to FALSE when the routine is called in reponse to
1 !                   breakpoints, watchpoints, and stepping by lines, so that the
1 !                   module name will not be printed.
1 !
1 !       STEP_FLAG - Flag set to TRUE to indicate the call is from STEP.
1 !
1 ! OUTPUTS
1 !       NONE
1 !
1
2     BEGIN
2
2     ENABLE
2         TYPE_PC_SOURCE_HANDLER;
2
2     LOCAL
2         LLINE_END,                        ! Low End PC of the selected line/stmt
2         LLINE_NUM,                        ! Low line number
2         LLINE_START,                      ! Low Start PC of the selected line/stmt
2         LMODNAMEPTR,                      ! Low PC module name pointer
2         LOW_MODPTR : REF RST$ENTRY,       ! Low PC module pointer
2         LSTMT_NUM,                        ! Low stmt number
2         HLINE_END,                        ! High End PC of the selected line/stmt
2         HLINE_NUM,                        ! High line number
2         HLINE_START,                      ! High Start PC of the selected line/stmt
2         HMODNAMEPTR,                      ! High PC module name pointer
2         HIGH_MODPTR,                      ! High PC module pointer
```

```
2           HSTMT_NUM,                      ! High stmt number
2           STATUS;                         ! return status from PC to LINE translation
2
2
2
2       ! Check to see if the PC is an entry point, if it is then PC = PC + 2.
2       ! Separately handle the case where LOW_PC = HIGH_PC, so that we only
2       ! have to call IS_IT_ENTRY once.
2
2       !IF .HIGH_PC EQL .LOW_PC
2       !THEN
2       !    BEGIN
2       !     IF (DBG$IS_IT_ENTRY(.LOW_PC))
2       !    THEN
2       !        BEGIN
2       !        LOW_PC = .LOW_PC + 2;
2       !        HIGH_PC = .LOW_PC;
2       !        END;
2       !    END
2       !
2       !ELSE
2       !    BEGIN
2       !     IF (DBG$IS_IT_ENTRY(.LOW_PC)) THEN LOW_PC = .LOW_PC + 2;
2       !     IF (DBG$IS_IT_ENTRY(.HIGH_PC)) THEN HIGH_PC = .HIGH_PC + 2;
2       !    END;
2
2
2       ! Ensure that higher PC value is greater than or equal to lower PC value.
2       ! If it is wrong, signal an error and give up.
2       !
2       IF (.LOW_PC GTR .HIGH_PC) THEN SIGNAL(DBG$_EXARANGE);
2
2
2       ! Match an absolute PC address to a line number.
2       !
2       LOW_MODPTR = 0;
2       STATUS = DBG$PC_TO_LINE_LOOKUP(.LOW_PC, LLINE_NUM, LSTMT_NUM, LLINE_START,
2                                       LLINE_END, LOW_MODPTR);
2
2
2       ! Check to see if any error or if match cannot be made in PC to LINE
2       ! translation.
2       !
2       IF NOT .STATUS
2       THEN
3           BEGIN
3           IF .MODPRTFLG
3           THEN
3               SIGNAL(DBG$_NOSRCLIN,1,.LOW_PC)
3           ELSE
3               RETURN;
3
2           END;
2
2
2       ! Get Module Name for lower PC value.
2       !
```

```
DBG$STA_SYMNAME(.LOW_MODPTR,LMODNAMEPTR);


! If two PC values are not the same, calls PC to LINE routine for the
! higher PC value to get the corresponding line number.  If both PC values
! are the same, set up parameters for routine TYPE_LNUM_SOURCE.
!
IF .LOW_PC EQL .HIGH_PC
THEN
    BEGIN
    HLINE_NUM = .LLINE_NUM;
    HSTMT_NUM = .LSTMT_NUM;
    END
ELSE
    BEGIN
    HIGH_MODPTR = 0;
    STATUS = DBG$PC_TO_LINE_LOOKUP(.HIGH_PC,HLINE_NUM,HSTMT_NUM,HLINE_START,
                                   HLINE_END,HIGH_MODPTR);


    ! Check to see if any error or if match cannot be made in PC to LINE
    ! translation.
    !
    IF NOT .STATUS
    THEN
        BEGIN
        IF .MODPRTFLG
        THEN
            SIGNAL(DBG$_NOSRCLIN,1,.HIGH_PC)
        ELSE
            RETURN;

        END;


    ! Get the Module Name for higher PC value.
    !
    DBG$STA_SYMNAME(.HIGH_MODPTR,HMODNAMEPTR);


    ! Check that both PC values are in the same module.
    !
    IF .LOW_MODPTR NEQ .HIGH_MODPTR
    THEN
        BEGIN
        IF .MODPRTFLG
        THEN
            SIGNAL(DBG$_ADDRANCOV,4,.LOW_PC,.LMODNAMEPTR,.HIGH_PC,.HMODNAMEPTR)
        ELSE
            RETURN;

        END;

    END;            ! End of matching high PC value to a line number.


! Check for the entry point.
```

```
!
IF (.LLINE_NUM EQL 0) AND (.HLINE_NUM EQL 0)
THEN
    BEGIN
    IF .MODPRTFLG
    THEN
        SIGNAL(DBG$_NOSRCLIN,1,.LOW_PC)
    ELSE
        RETURN;

    END;


! Make sure Module Source Table pointer in module's Module RST Entry
! points to a list of Source File DST Record Pointers.
! If not, then check whether we have reached this point through a step
! or breakpoint. In this case we do not want to signal an error, so
! we just return.
!
IF (.LOW_MODPTR[RST$L_MODSRCTBL] EQL 0) AND (NOT .MODPRTFLG)
THEN
    RETURN;


! Print out Module name if the print-module-name flag is set to TRUE.
! Note that the message is suppressed if the source line output is
! being directed to a source screen display.
!
IF .MODPRTFLG AND (.DBG$GL_SCREEN_SOURCE EQL 0)
THEN
    BEGIN
    DBG$PRINT(UPLIT BYTE(%ASCIC 'module !AC'), .LMODNAMEPTR);
    DBG$NEWLINE();
    END;


! Call TYPE_LNUM_SOURCE to type out the given source line number ranges.
!
DBG$SRC_TYPE_LNUM_SOURCE(.LOW_MODPTR, .LLINE_NUM, .LSTMT_NUM,
                         .HLINE_NUM, .HSTMT_NUM, 0, .STEP_FLAG);

RETURN;

END;



                                                        .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

43 41 21 20 65 6C 75 64 6F 6D 0A  002BE P.ABP:  .ASCII  <10>\module !AC\                       ;


                                                        .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                             01FC 00000                 .ENTRY  DBG$SRC_TYPE_PC_SOURCE, Save R2,R3,R4,R5,-  ; 3793
                                                                R6,R7,R8                                     ;
```

```
      58 00000000G   00  9E 00002        MOVAB   DBG$STA_SYMNAME, R8              ;
      57 00000000G   00  9E 00009        MOVAB   DBG$PC_TO_LINE_LOOKUP, R7        ;
      56 00000000G   00  9E 00010        MOVAB   LIB$SIGNAL, R6                   ;
      5E             30  C2 00017        SUBL2   #48, SP                          ;
      6D       0112  CF  DE 0001A        MOVAL   13$, (FP)                        ; 3833
      54         04  AC  D0 0001F        MOVL    LOW_PC, R4                       ; 3880
      52         08  AC  D0 00023        MOVL    HIGH_PC, R2                      ;
      52             54  D1 00027        CMPL    R4, R2                           ;
                     09  15 0002A        BLEQ    1$                               ;
           00028190  8F  DD 0002C        PUSHL   #164240                          ;
      66             01  FB 00032        CALLS   #1, LIB$SIGNAL                   ;
                     6E  D4 00035 1$:    CLRL    LOW_MODPTR                       ; 3885
                     5E  DD 00037        PUSHL   SP                               ; 3886
                 08  AE  9F 00039        PUSHAB  LLINE_END                        ;
                 10  AE  9F 0003C        PUSHAB  LLINE_START                      ;
                 18  AE  9F 0003F        PUSHAB  LSTMT_NUM                        ;
                 20  AE  9F 00042        PUSHAB  LLINE_NUM                        ;
                     54  DD 00045        PUSHL   R4                               ;
      67             06  FB 00047        CALLS   #6, DBG$PC_TO_LINE_LOOKUP        ;
      55             50  D0 0004A        MOVL    R0, STATUS                       ;
      11             55  E8 0004D        BLBS    STATUS, 2$                       ; 3893
      62         0C  AC  E9 00050        BLBC    MODPRTFLG, 5$                    ; 3896
                     54  DD 00054        PUSHL   R4                               ; 3898
                     01  DD 00056        PUSHL   #1                               ;
           00028CD0  8F  DD 00058        PUSHL   #167120                          ;
      66             03  FB 0005E        CALLS   #3, LIB$SIGNAL                   ;
                 14  AE  9F 00061 2$:    PUSHAB  LMODNAMEPTR                      ; 3907
      53         04  AE  D0 00064        MOVL    LOW_MODPTR, R3                   ;
                     53  DD 00068        PUSHL   R3                               ;
      68             02  FB 0006A        CALLS   #2, DBG$STA_SYMNAME              ;
      52             54  D1 0006D        CMPL    R4, R2                           ; 3914
                     07  12 00070        BNEQ    3$                               ;
   24 AE         0C  AE  7D 00072        MOVQ    LSTMT_NUM, HSTMT_NUM             ; 3918
                     56  11 00077        BRB     6$                               ; 3914
                 18  AE  D4 00079 3$:    CLRL    HIGH_MODPTR                      ; 3922
                 18  AE  9F 0007C        PUSHAB  HIGH_MODPTR                      ; 3923
                 20  AE  9F 0007F        PUSHAB  HLINE_END                        ;
                 28  AE  9F 00082        PUSHAB  HLINE_START                      ;
                 30  AE  9F 00085        PUSHAB  HSTMT_NUM                        ;
                 38  AE  9F 00088        PUSHAB  HLINE_NUM                        ;
                     52  DD 0008B        PUSHL   R2                               ;
      67             06  FB 0008D        CALLS   #6, DBG$PC_TO_LINE_LOOKUP        ;
      55             50  D0 00090        MOVL    R0, STATUS                       ;
      11             55  E8 00093        BLBS    STATUS, 4$                       ; 3930
      3F         0C  AC  E9 00096        BLBC    MODPRTFLG, 7$                    ; 3933
                     52  DD 0009A        PUSHL   R2                               ; 3935
                     01  DD 0009C        PUSHL   #1                               ;
           00028CD0  8F  DD 0009E        PUSHL   #167120                          ;
      66             03  FB 000A4        CALLS   #3, LIB$SIGNAL                   ;
                 2C  AE  9F 000A7 4$:    PUSHAB  HMODNAMEPTR                      ; 3944
                 1C  AE  DD 000AA        PUSHL   HIGH_MODPTR                      ;
      68             02  FB 000AD        CALLS   #2, DBG$STA_SYMNAME              ;
   18 AE             53  D1 000B0        CMPL    R3, HIGH_MODPTR                  ; 3949
                     19  13 000B4        BEQL    6$                               ;
      75         0C  AC  E9 000B6 5$:    BLBC    MODPRTFLG, 12$                   ; 3952
                 2C  AE  DD 000BA        PUSHL   HMODNAMEPTR                      ; 3954
                     52  DD 000BD        PUSHL   R2                               ;
```

```
                       1C    AE  DD 000BF        PUSHL   LMODNAMEPTR                        ;
                             54  DD 000C2        PUSHL   R4                                 ;
                             04  DD 000C4        PUSHL   #4                                 ;
               00028CD8      8F  DD 000C6        PUSHL   #167128                            ;
              66             06  FB 000CC        CALLS   #6, LIB$SIGNAL                     ;
                       10    AE  D5 000CF 6$:    TSTL    LLINE_NUM                          ; 3965
                             16  12 000D2        BNEQ    8$                                 ;
                       28    AE  D5 000D4        TSTL    HLINE_NUM                          ;
                             11  12 000D7        BNEQ    8$                                 ;
              52       0C    AC  E9 000D9 7$:    BLBC    MODPRTFLG, 12$                     ; 3968
                             54  DD 000DD        PUSHL   R4                                 ; 3970
                             01  DD 000DF        PUSHL   #1                                 ;
               00028CD0      8F  DD 000E1        PUSHL   #167120                            ;
              66             03  FB 000E7        CALLS   #3, LIB$SIGNAL                     ;
                       24    A3  D5 000EA 8$:    TSTL    36(R3)                             ; 3983
                             05  12 000ED        BNEQ    9$                                 ;
              05       0C    AC  E8 000EF        BLBS    MODPRTFLG, 10$                     ;
                             04     000F3        RET                                        ; 3985
              1F       0C    AC  E9 000F4 9$:    BLBC    MODPRTFLG, 11$                     ; 3992
               00000000G     00  D5 000F8 10$:   TSTL    DBG$GL_SCREEN_SOURCE               ;
                             17  12 000FE        BNEQ    11$                                ;
                       14    AE  DD 00100        PUSHL   LMODNAMEPTR                        ; 3995
               00000000'     EF  9F 00103        PUSHAB  P.ABP                              ;
    00000000G 00             02  FB 00109        CALLS   #2, DBG$PRINT                      ;
    00000000G 00             00  FB 00110        CALLS   #0, DBG$NEWLINE                    ; 3996
                       10    AC  DD 00117 11$:   PUSHL   STEP_FLAG                          ; 4003
                             7E  D4 0011A        CLRL    -(SP)                              ; 4002
                       2C    AE  DD 0011C        PUSHL   HSTMT_NUM                          ; 4003
                       34    AE  DD 0011F        PUSHL   HLINE_NUM                          ;
                       1C    AE  DD 00122        PUSHL   LSTMT_NUM                          ; 4002
                       24    AE  DD 00125        PUSHL   LLINE_NUM                          ;
                             53  DD 00128        PUSHL   R3                                 ;
         FA67 CF             07  FB 0012A        CALLS   #7, DBG$SRC_TYPE_LNUM_SOURCE       ;
                             04     0012F 12$:   RET                                        ; 4007
                           0000     00130 13$:   .WORD   Save nothing                       ; 3833
                             7E  D4 00132        CLRL    -(SP)                              ;
                             5E  DD 00134        PUSHL   SP                                 ;
                 7E    04    AC  7D 00136        MOVQ    4(AP), -(SP)                       ;
         0000V CF            03  FB 0013A        CALLS   #3, TYPE_PC_SOURCE_HANDLER         ;
                             04     0013F        RET                                        ;

; Routine Size:  320 bytes,    Routine Base:  DBG$CODE + 1973
```

```
1 ROUTINE OUTPUT_DIR_LIST (DIRPTR) : NOVALUE =
1 !
1 ! FUNCTION
1 !       This routine outputs the Source Directory Search List
1 !       for one module (or the default list) on the user's terminal.
1 !       It prints each directory name on a separate line, indented
1 !       eight spaces. This routine is called during the SHOW SOURCE
1 !       command.
1 !
1 ! INPUTS
1 !       DIRPTR  -       The first of a linked list of Source Directory
1 !                       Search List Entries.
1 !
1 ! OUTPUTS
1 !       NONE
1
1
2     BEGIN
2     MAP
2       DIRPTR : REF SDSL$ENTRY;          ! A pointer to a Source Directory
2                                         !       Search List Entry. This is
2                                         !       used to walk through the
2                                         !       linked list.
2
2
2
2     ! Loop through the list, outputting the directory name in each Source
2     ! Directory Search List Entry.
2     !
2     WHILE .DIRPTR NEQ 0 DO
3       BEGIN
3       DBG$PRINT (UPLIT BYTE(
3           %ASCIC '        !AC'), DIRPTR[SDSL$B_ENT_DIRLEN]);
3       DBG$NEWLINE();
3       DIRPTR = .DIRPTR[SDSL$L_ENT_FLINK];
2       END;
2
2     RETURN;
2
1     END;


                                                              .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

        43  41  21  20  20  20  20  20  20  20  20  0B  002C9 P.ABQ:  .ASCII  <11>\        !AC\                 ;


                                                              .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                         0004 00000 OUTPUT_DIR_LIST:
                                                              .WORD   Save R2                              ; 4008
                      52        04   AC  D0 00002 1$:         MOVL    DIRPTR, R2                           ; 4037
                                     1D  13 00006             BEQL    2$                                   ;
                                04   A2  9F 00008             PUSHAB  4(R2)                                ; 4040
                         00000000'   EF  9F 0000B             PUSHAB  P.ABQ                                ; 4039
```

```
                00000000G  00       02  FB 00011          CALLS   #2, DBG$PRINT                          ; 4040
                00000000G  00       00  FB 00018          CALLS   #0, DBG$NEWLINE                        ; 4041
                       04  AC       62  D0 0001F          MOVL    (R2), DIRPTR                           ; 4042
                                    DD  11 00023          BRB     1$                                     ; 4037
                                        04 00025 2$:      RET                                            ; 4047

; Routine Size:  38 bytes,    Routine Base:  DBG$CODE + 1AB3


; 3934          4048  1
```

```
; 3936          4049  1 ROUTINE TYPE_PC_SOURCE_HANDLER(SIGARG, MECHARG, ENBLARG) =
; 3937          4050  1 !
; 3938          4051  1 ! FUNCTION
; 3939          4052  1 !       This routine is the handler for DBG$SRC_TYPE_PC_HANDLER routine.
; 3940          4053  1 !       It catches the DBG$_FILEUNAL signal and unwinds, it resignals
; 3941          4054  1 !       everything else.
; 3942          4055  1 !
; 3943          4056  1 ! INPUTS
; 3944          4057  1 !       SIGARG  - The signal argument vector.
; 3945          4058  1 !
; 3946          4059  1 !       MECHARG - The mechanism argument vector.
; 3947          4060  1 !
; 3948          4061  1 !       ENBLARG - The enable argument vector. (not used).
; 3949          4062  1 !
; 3950          4063  1
; 3951          4064  2     BEGIN
; 3952          4065  2
; 3953          4066  2     MAP
; 3954          4067  2         SIGARG: REF VECTOR[,LONG];        ! Pointer to the signal argument vector
; 3955          4068  2
; 3956          4069  2     IF .SIGARG[1] EQL DBG$_FILEUNAL
; 3957          4070  2     THEN
; 3958          4071  3         BEGIN
; 3959          4072  3         SETUNWIND();
; 3960          4073  3         RETURN 0;
; 3961          4074  2         END;
; 3962          4075  2
; 3963          4076  2     RETURN SS$_RESIGNAL;
; 3964          4077  2
; 3965          4078  1     END;



                                          0000 00000 TYPE_PC_SOURCE_HANDLER:
                                                                  .WORD   Save nothing                       ; 4049
                        50         04    AC  D0 00002             MOVL    SIGARG, R0                         ; 4069
             00028D10   8F         04    A0  D1 00006             CMPL    4(R0), #167184                     ;
                                         0B  12 0000E             BNEQ    1$                                 ;
                                         7E  7C 00010             CLRQ    -(SP)                              ; 4072
             00000000G  00               02  FB 00012             CALLS   #2, SYS$UNWIND                     ;
                                         06  11 00019             BRB     2$                                 ; 4073
                        50       0918    8F  3C 0001B 1$:         MOVZWL  #2328, R0                          ; 4076
                                             04 00020             RET                                        ;
                                         50  D4 00021 2$:         CLRL    R0                                 ; 4078
                                             04 00023             RET                                        ;

; Routine Size:  36 bytes,    Routine Base:  DBG$CODE + 1AD9


; 3966          4079  1
; 3967          4080  0 END ELUDOM
```

```
                                                                 .EXTRN  LIB$SIGNAL, SYS$UNWIND

;                                 PSECT SUMMARY
;
;       Name                    Bytes                        Attributes
;
;   DBG$GLOBAL                       284  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;   DBG$OWN                          280  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;   DBG$CODE                        6909  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(0)
;   DBG$PLIT                         725  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(0)


;                            Library Statistics
;
;                                               -------- Symbols --------        Pages       Processing
;       File                                    Total    Loaded   Percent        Mapped      Time
;
;   _$255$DUA28:[SYSLIB]LIB.L32;1               18619      122         0          1000        00:01.9
;   _$255$DUA28:[DEBUG.OBJ]STRUCDEF.L32;1          32        0         0             7        00:00.2
;   _$255$DUA28:[DEBUG.OBJ]DBGLIB.L32;1          1545      125         8            97        00:01.9
;   _$255$DUA28:[DEBUG.OBJ]DSTRECRDS.L32;1
;                                                 418      135        32            31        00:00.3
;   _$255$DUA28:[DEBUG.OBJ]DBGMSG.L32;1           386       19         4            22        00:00.3



;                               COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:DBGSOURCE/OBJ=OBJ$:DBGSOURCE MSRC$:DBGSOURCE/UPDATE=(ENH$:DBGSOURCE)

; Size:            6909 code + 1289 data bytes
; Run Time:           02:11.1
; Elapsed Time:       02:24.2
; Lines/CPU Min:       1867
; Lexemes/CPU-Min: 16219
; Memory Used:  477 pages
; Compilation Complete
```

DBGSSV
LIS

DBGSTEPGO
LIS

DBGSTART
LIS

DBGSOURCE
LIS