```
DDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBB      UUU         UUU     GGGGGGGGGGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUU         UUU     GGGGGGGGGGGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB     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 EEEEEEEEEEE      BBBBBBBBBBBB     UUU         UUU GGG
DDD         DDD EEEEEEEEEEE      BBBBBBBBBBBB     UUU         UUU GGG
DDD         DDD EEEEEEEEEEE      BBBBBBBBBBBB     UUU         UUU GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG   GGGGGGGGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG   GGGGGGGGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG   GGGGGGGGG
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      UUUUUUUUUUUUUUUUU     GGGGGGGGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB      UUUUUUUUUUUUUUUUU     GGGGGGGGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB      UUUUUUUUUUUUUUUUU     GGGGGGGGG
```

```
DDDDDDDD    BBBBBBBB      GGGGGGGG    SSSSSSSS    000000    UU      UU  RRRRRRRR      CCCCCCCC  EEEEEEEEEE
DDDDDDDD    BBBBBBBB      GGGGGGGG    SSSSSSSS    000000    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      000000    UUUUUUUUUU  RR      RR    CCCCCCCC  EEEEEEEEEE          ....
DDDDDDDD    BBBBBBBB      GGGGGG    SSSSSSSS      000000    UUUUUUUUUU  RR      RR    CCCCCCCC  EEEEEEEEEE          ....

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
```

```
:     1          0001  0 MODULE DBGSOURCE (IDENT = 'V04-000') =
:     2          0002  0
:     3          0003  1 BEGIN
:     4          0004  1
:     5          0005  1 !**************************************************************************
:     6          0006  1 !*                                                                        *
:     7          0007  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
:     8          0008  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
:     9          0009  1 !*  ALL RIGHTS RESERVED.                                                   *
:    10          0010  1 !*                                                                        *
:    11          0011  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
:    12          0012  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
:    13          0013  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
:    14          0014  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
:    15          0015  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
:    16          0016  1 !*  TRANSFERRED.                                                           *
:    17          0017  1 !*                                                                        *
:    18          0018  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
:    19          0019  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
:    20          0020  1 !*  CORPORATION.                                                           *
:    21          0021  1 !*                                                                        *
:    22          0022  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
:    23          0023  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
:    24          0024  1 !*                                                                        *
:    25          0025  1 !*                                                                        *
:    26          0026  1 !**************************************************************************
:    27          0027  1 !
:    28          0028  1 ! WRITTEN BY
:    29          0029  1 !       Bert Beander    August, 1981
:    30          0030  1 !       Ping Sager
:    31          0031  1 !       Rich Title
:    32          0032  1 !
:    33          0033  1 ! MODULE FUNCTION
:    34          0034  1 !       This module contains all routines connected with DEBUG's source line
:    35          0035  1 !       display feature.  It thus implements the TYPE, EXAMINE/SOURCE, SET
:    36          0036  1 !       SOURCE, SHOW SOURCE, and CANCEL SOURCE commands, and it contains all
:    37          0037  1 !       routines which look up source lines and open and read source files.
:    38          0038  1 !
:    39          0039  1
:    40          0040  1 REQUIRE 'SRC$:DBGPROLOG.REQ';
:    41          0174  1
:    42          0175  1 FORWARD ROUTINE
:    43          0176  1     DBG$SRC_CANCEL_SOURCE: NOVALUE,     ! Handle the CANCEL SOURCE command
:    44          0177  1     DBG$SRC_CLOSE_FILE: NOVALUE,        ! Close a source file
:    45          0178  1     DBG$SRC_INIT: NOVALUE,              ! Initialize data structures at DEBUG
:    46          0179  1                                         !       initialization time
:    47          0180  1     DBG$SRC_LINE_TO_REC: NOVALUE,       ! Translate line number to record number
:    48          0181  1     DBG$SRC_LNUM_RANGE: NOVALUE,        ! Returns a module's line number range
:    49          0182  1     DBG$SRC_OPEN,                       ! Actually does the open and verifies
:    50          0183  1                                         !       the attributes of a source file
:    51          0184  1     DBG$SRC_OPEN_FILE: NOVALUE,         ! Open a source file after applying all
:    52          0185  1                                         !       directory search rules
:    53          0186  1     DBG$SRC_OUTPUT_LINE: NOVALUE,       ! Output a source line in proper format
:    54          0187  1     DBG$SRC_READ_FILE: NOVALUE,         ! Read a source line from a source file
:    55          0188  1     DBG$SRC_SEARCH,                     ! Actually does the search
:    56          0189  1     DBG$SRC_SEARCH_CMD: NOVALUE,        ! Handle the SEARCH command
:    57          0190  1     DBG$SRC_SET_MAX_FILES: NOVALUE,     ! Set maximum number of source files
```

```
;    58          0191  1                                    !       that can be open at one time
;    59          0192  1      DBG$SRC_SET_SOURCE: NOVALUE,  ! Handle the SET SOURCE command
;    60          0193  1      DBG$SRC_SHOW_SOURCE: NOVALUE, ! Handle the SHOW SOURCE command
;    61          0194  1      DBG$SRC_TYPE_LINE,            ! Retrieve and type out a source line
;    62          0195  1      DBG$SRC_TYPE_LNUM_SOURCE: NOVALUE, ! Type a set of source lines given a
;    63          0196  1                                    !       module and a line number range
;    64          0197  1      DBG$SRC_TYPE_PC_SOURCE: NOVALUE, ! Type a set of source lines given a
;    65          0198  1                                    !       Program Counter range
;    66          0199  1      OUTPUT_DIR_LIST : NOVALUE,    ! Outputs a source directory list at
;    67          0200  1                                    ! the terminal.
;    68          0201  1      TYPE_PC_SOURCE_HANDLER;       ! Handler to catch the message
```

```
;    70          0202  1 EXTERNAL ROUTINE
;    71          0203  1     DBG$FORMAT_FAO_OUT: NOVALUE,        ! Build an output buffer for signaling
;    72          0204  1     DBG$GET_MEMORY,                     ! Get a permanent memory block
;    73          0205  1     DBG$GET_TEMPMEM,                    ! Get a tempory memory block
;    74          0206  1     DBG$NEWLINE: NOVALUE,               ! Output the current print buffer and
;    75          0207  1                                         !     initialize a new print line
;    76          0208  1     DBG$PC_TO_LINE_LOOKUP,              ! Matches an absolute PC address to a
;    77          0209  1                                         !     line number
;    78          0210  1     DBG$PRINT: NOVALUE,                 ! Output text to the print buffer using
;    79          0211  1                                         !     FAO formatting
;    80          0212  1     DBG$REL_MEMORY: NOVALUE,            ! Release a permanent memory block
;    81          0213  1     DBG$SCR_SOURCE_BEGIN: NOVALUE,      ! Begin source output to screen display
;    82          0214  1     DBG$SCR_SOURCE_END: NOVALUE,        ! End source output to screen display
;    83          0215  1     DBG$SCR_SOURCE_LINE: NOVALUE,       ! Output a source line to a screen display
;    84          0216  1     DBG$STA_SYMNAME: NOVALUE,           ! Get module name for a given module RST
;    85          0217  1                                         !     pointer
;    86          0218  1     LBR$CLOSE,                          ! Closes an open library
;    87          0219  1     LBR$FIND,                           ! Looks up a key by RFA in preparation
;    88          0220  1                                         !     for reading the key's associated
;    89          0221  1                                         !     text
;    90          0222  1     LBR$GET_RECORD,                     ! Reads a text record associated with a
;    91          0223  1                                         !     module in the library
;    92          0224  1     LBR$LOOKUP_KEY,                     ! Looks up a module in a text library
;    93          0225  1     LBR$INI_CONTROL,                    ! Initializes a library index for use
;    94          0226  1                                         !     by all other routines
;    95          0227  1     LBR$OPEN,                           ! Opens an existing library
;    96          0228  1     LBR$SET_MODULE,                     ! Reads an module header in a text library
;    97          0229  1     STR$UPCASE,                         ! Uppercase Conversion
;    98          0230  1     SYS$FAOL;                           ! Formatted ASCII Output routine to
;    99          0231  1                                         !     do text formatting
;   100          0232  1
;   101          0233  1 EXTERNAL
;   102          0234  1     DBG$GL_CHARTBL: REF CHRTBL$TABLE,   ! Pointer to DBGPARSER's Character Table
;   103          0235  1                                         !     for the currently set language
;   104          0236  1     DBG$GB_LANGUAGE: BYTE,              ! The current language code
;   105          0237  1     DBG$GL_SCREEN_SOURCE;               ! Screen Display ID for source or zero
;   106          0238  1
;   107          0239  1 LITERAL
;   108          0240  1     MAX_MAX_FILES       = 20,           ! Maximum number of Source File Control
;   109          0241  1                                         !     Blocks we allow user to create
;   110          0242  1     NAMBUFSIZE          = 256,          ! Size in bytes of name buffers pointed
;   111          0243  1                                         !     to by the RMS NAM block
;   112          0244  1     SEARCH_IDENT_ALL    = 1,            ! SEARCH/IDENT/ALL is active
;   113          0245  1     SEARCH_IDENT_NEXT   = 2,            ! SEARCH/IDENT/NEXT is active
;   114          0246  1     SEARCH_STRING_ALL   = 3,            ! SEARCH/STRING/ALL is active
;   115          0247  1     SEARCH_STRING_NEXT  = 4,            ! SEARCH/STRING/NEXT is active
;   116          0248  1     SRCBUFSIZE          = 256,          ! Size in bytes of source record buffer
;   117          0249  1     RFATBL_SIZE         = 300,          ! Size in bytes of the RFA table
;   118          0250  1     RFATBL_ENTRIES      = 50,           ! Number of entries in the RFA table
;   119          0251  1     INIT_RFATBL_SPACING = 5;            ! Number of source records per RFA
;   120          0252  1                                         ! table entry.
;   121          0253  1
;   122          0254  1 GLOBAL
;   123          0255  1     DBG$SRC_LEFT_MARGIN : INITIAL(1),   ! Left margin of source output
;   124          0256  1     DBG$SRC_RIGHT_MARGIN : INITIAL(255),! Right margin of source output
;   125          0257  1     DBG$SRC_TERM_WIDTH: INITIAL(80),    ! Terminal width for source line display
;   126          0258  1     DBG$SRC_MAX_FILES: INITIAL(5),      ! Maximum number of source files to be
```

```
;  127     0259  1                                    !       kept open at any one time
;  128     0260  1      DBG$SRC_NEXT_LNUM: INITIAL(1),  ! Next desired line in the current
;  129     0261  1                                    !       module to be typed
;  130     0262  1      DBG$SRC_NEXT_MODRSTPTR: INITIAL(0),  ! Associated modul- RST pointer
;  131     0263  1      DBG$SRC_NEXT_STMT: INITIAL(0).  ! Next desired statement number in the
;  132     0264  1                                    !       current module to be typed
;  133     0265  1      DBG$SRC_SEARCH_STRING: VECTOR[SRCBUFSIZE,BYTE]
;  134     0266  1          INITIAL (BYTE (0, REP (SRCBUFSIZE - 1) OF (%C' ')));
;  135     0267  1                                    ! Contains a counted ASCII search string
;  136     0268  1                                    !       filled by SEARCH Command.
;  137     0269  1
;  138     0270  1
;  139     0271  1  OWN
;  140     0272  1      DBG$SRC_DIR_LIST: REF SDSL$HEADER  ! Pointer to linked list of Source Dir-
;  141     0273  1                          INITIAL(0),    !       ectory Search List Header blocks
;  142     0274  1      DBG$SRC_FORMFEED_FLAG: INITIAL(0),  ! Flag set to indicate not to skip
;  143     0275  1                                    !       form feed
;  144     0276  1      DBG$SRC_MODRSTPTR: REF RST$ENTRY,  ! Pointer to the Module RST Entry for
;  145     0277  1                                    !       the "current module", the module
;  146     0278  1                                    !       we are now reading source from
;  147     0279  1      DBG$SRC_SEARCH_FLAG,           ! Set to SEARCH_FLAG's value which
;  148     0280  1                                    !       indicate whether the SEARCH
;  149     0281  1                                    !       command is in action and the
;  150     0282  1                                    !       kind of search option is used
;  151     0283  1      DBG$SRC_SFCB_PTR: REF SFCB$BLOCK  ! Pointer to first Source File Control
;  152     0284  1                          INITIAL(0),    !       Block on doubly linked list of
;  153     0285  1                                    !       such blocks
;  154     0286  1      DBG$SRC_REC_BUF: VECTOR[SRCBUFSIZE,  ! Buffer area to hold the source file
;  155     0287  1                          BYTE],         !       record
;  156     0288  1      SOURCE_TO_SCREEN_FLAG;         ! Flag used to direct source output to
;  157     0289  1                                    !       a source screen display
```

```
;  159     0290  1 GLOBAL ROUTINE DBG$SRC_CANCEL_SOURCE(MODRSTPTR): NOVALUE =
;  160     0291  1 !
;  161     0292  1 ! FUNCTION
;  162     0293  1 !       This routine handles the semantic processing of the CANCEL SOURCE and
;  163     0294  1 !       CANCEL SOURCE/MODULE=xxx commands.  It removes the directory list for
;  164     0295  1 !       the specified module, or the default directory list if the /MODULE
;  165     0296  1 !       qualifier was not present, from the list of directory search lists
;  166     0297  1 !       pointed to by DBG$SRC_DIR_LIST.  It thus cancels the effect of the
;  167     0298  1 !       SET SOURCE or SET SOURCE/MODULE=xxx command which created that direct-
;  168     0299  1 !       ory search list.
;  169     0300  1 !
;  170     0301  1 ! INPUTS
;  171     0302  1 !       MODRSTPTR - A pointer to the Module RST Entry of the module specified
;  172     0303  1 !                   on the CANCEL SOURCE/MODULE command.  If the /MODULE quali-
;  173     0304  1 !                   fier was not present on the command, MODRSTPTR must be zero.
;  174     0305  1 !
;  175     0306  1 ! OUTPUTS
;  176     0307  1 !       NONE
;  177     0308  1 !
;  178     0309  1 !--
;  179     0310  1
;  180     0311  2     BEGIN
;  181     0312  2
;  182     0313  2     MAP
;  183     0314  2         MODRSTPTR: REF RST$ENTRY;       ! Pointer to Module RST Entry or zero
;  184     0315  2
;  185     0316  2     LOCAL
;  186     0317  2         HDRPTR : REF SDSL$HEADER,       ! Pointer to a Source Directory Search
;  187     0318  2                                         !       List Header Block.
;  188     0319  2                                         !       This pointer is used to walk
;  189     0320  2                                         !       through the linked list of
;  190     0321  2                                         !       header blocks.
;  191     0322  2         ENTRY_PTR : REF SDSL$ENTRY,     ! Pointer to a Source Directory Search
;  192     0323  2                                         !       List Entry.This pointer is
;  193     0324  2                                         !       used to walk through the linked
;  194     0325  2                                         !       list of entries, once the
;  195     0326  2                                         !       appropriate source directory
;  196     0327  2                                         !       search list is found.
;  197     0328  2         MODNAMEPTR,                     ! Points to a counted string with the
;  198     0329  2                                         !       module name corresponding to
;  199     0330  2                                         !       MODRSTPTR. (Needed in case an
;  200     0331  2                                         !       error message is signalled).
;  201     0332  2         PREV_HDRPTR : REF SDSL$HEADER,  ! This variable contains the previous
;  202     0333  2                                         !       Source Directory Search List Header
;  203     0334  2                                         !       as we walk through the linked
;  204     0335  2                                         !       list. We need to remember this
;  205     0336  2                                         !       so we can unlink the header
;  206     0337  2                                         !       being cancelled.
;  207     0338  2         TEMP_PTR;                       ! Pointer to a source directory list
;  208     0339  2                                         !       entry. Used during freeing up
;  209     0340  2                                         !       of space.
;  210     0341  2
;  211     0342  2     ! Attempt to find a directory list for the module specified by MODRSTPTR.
;  212     0343  2     !
;  213     0344  2     PREV_HDRPTR = DBG$SRC_DIR_LIST;
;  214     0345  2     HDRPTR = .DBG$SRC_DIR_LIST;
;  215     0346  2     WHILE .HDRPTR NEQ 0 DO
```

```
;  216          0347  3          BEGIN
;  217          0348  3          IF .HDRPTR [SDSL$L_MODPTR] EQL .MODRSTPTR
;  218          0349  3          THEN
;  219          0350  3              EXITLOOP;
;  220          0351  3          PREV_HDRPTR = .HDRPTR;
;  221          0352  3          HDRPTR = .HDRPTR[SDSL$L_FLINK];
;  222          0353  2          END;
;  223          0354  2
;  224          0355  2
;  225          0356  2      ! If no source directory search list header block was found for the
;  226          0357  2      ! specified module, signal an error to the user.
;  227          0358  2      !
;  228          0359  2      IF .HDRPTR EQL 0
;  229          0360  2      THEN
;  230          0361  3          BEGIN
;  231          0362  3          IF .MODRSTPTR NEQ 0
;  232          0363  3          THEN
;  233          0364  4              BEGIN
;  234          0365  4              DBG$STA_SYMNAME (.MODRSTPTR, MODNAMEPTR);
;  235          0366  4              SIGNAL (DBG$_NODIRLISM, 1, .MODNAMEPTR);
;  236          0367  4              END
;  237          0368  4
;  238          0369  3          ELSE
;  239          0370  3              SIGNAL (DBG$_NODIRLIST);
;  240          0371  3
;  241          0372  2          END;
;  242          0373  2
;  243          0374  2
;  244          0375  2      ! Unlink the Source Directory Search List Header Block for the
;  245          0376  2      ! MODRSTPTR module from the Source Directory Search List. Then
;  246          0377  2      ! release that block and all associated Source Directory Search
;  247          0378  2      ! List Entries to the memory pool
;  248          0379  2      !
;  249          0380  2      PREV_HDRPTR[SDSL$L_FLINK] = .HDRPTR[SDSL$L_FLINK];
;  250          0381  2      ENTRY_PTR = .HDRPTR[SDSL$L_LIST_PTR];
;  251          0382  2      WHILE .ENTRY_PTR NEQ 0 DO
;  252          0383  3          BEGIN
;  253          0384  3          TEMP_PTR = .ENTRY_PTR[SDSL$L_ENT_FLINK];
;  254          0385  3          DBG$REL_MEMORY (.ENTRY_PTR);
;  255          0386  3          ENTRY_PTR = .TEMP_PTR;
;  256          0387  2          END;
;  257          0388  2      DBG$REL_MEMORY (.HDRPTR);
;  258          0389  2
;  259          0390  2
;  260          0391  2      ! Reinitialize all Source File Control Blocks. This forces all open source
;  261          0392  2      ! files to be reinitialized when accessed so that the modified Source
;  262          0393  2      ! Directory Search List will be used to select and open those files.
;  263          0394  2      !
;  264          0395  2      DBG$SRC_INIT();
;  265          0396  2
;  266          0397  2      RETURN;
;  267          0398  2
;  268          0399  1      END;

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

```
                        .PSECT  DBG$DWN,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
```

```
;  270     0400  1 ROUTINE DBG$SRC_CLOSE_FILE(SFCBPTR): NOVALUE =
;  271     0401  1 !
;  272     0402  1 ! FUNCTION
;  273     0403  1 !       This routine closes an open source file.  It accepts a Source File Con-
;  274     0404  1 !       trol Block (SFCB) pointer as input, determines whether the correspond-
;  275     0405  1 !       ing file is an RMS file or a module in a source library, and closes the
;  276     0406  1 !       file.  It returns immediately if the SFCB is not in use.
;  277     0407  1 !
;  278     0408  1 ! INPUTS
;  279     0409  1 !       SFCBPTR - A pointer to the Source File Control Block for the source file
;  280     0410  1 !                 to be closed.
;  281     0411  1 !
;  282     0412  1 ! OUTPUTS
;  283     0413  1 !       NONE
;  284     0414  1 !
;  285     0415  1
;  286     0416  2     BEGIN
;  287     0417  2
;  288     0418  2     MAP
;  289     0419  2         SFCBPTR: REF SFCB$BLOCK;        ! Pointer to Source File Control Block
;  290     0420  2
;  291     0421  2     LOCAL
;  292     0422  2         STATUS;                         ! Close status
;  293     0423  2
;  294     0424  2
;  295     0425  2
;  296     0426  2     ! Close the file specified by the Source File Control Block pointer
;  297     0427  2     ! SFCBPTR.  Determine the kind of the SFCBPTR block and act accordingly.
;  298     0428  2     !
;  299     0429  2     CASE .SFCBPTR[SFCB$B_KIND] FROM SFCB$K_NOTUSED TO SFCB$K_FILE_UNAVAIL OF
;  300     0430  2         SET
;  301     0431  2
;  302     0432  2
;  303     0433  2         ! This Source File Control Block is not used or is occupied by an
;  304     0434  2         ! available file.  Nothing needs to be done.
;  305     0435  2         !
;  306     0436  2         [SFCB$K_NOTUSED, SFCB$K_FILE_UNAVAIL]:
;  307     0437  2             0;
;  308     0438  2
;  309     0439  2
;  310     0440  2         ! This Source File Control Block is for an open RMS source file.  Close
;  311     0441  2         ! RMS file.
;  312     0442  2         !
;  313     0443  2         [SFCB$K_RMSFILE]:
;  314     0444  3             BEGIN
;  315     0445  3             STATUS = $CLOSE(FAB = .SFCBPTR[SFCB$L_FABPTR]);
;  316     0446  3             IF NOT .STATUS THEN SIGNAL (.STATUS);
;  317     0447  2             END;
;  318     0448  2
;  319     0449  2
;  320     0450  2         ! This Source File Control Block is for a module in an open source
;  321     0451  2         ! library.  Close the source library.
;  322     0452  2         !
;  323     0453  2         [SFCB$K_LBRFILE]:
;  324     0454  3             BEGIN
;  325     0455  3             STATUS = LBR$CLOSE(SFCBPTR[SFCB$L_LBRINDEX]);
;  326     0456  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
```

```
;  343     0472  1 GLOBAL ROUTINE DBG$SRC_INIT: NOVALUE =
;  344     0473  1 !
;  345     0474  1 ! FUNCTION
;  346     0475  1 !       This routine initializes the Source File Control Blocks and associated
;  347     0476  1 !       FABs and RABs needed by the Source Line Display feature to open and read
;  348     0477  1 !       source files.  It also fixes the maximum number of source files DEBUG
;  349     0478  1 !       will keep open at any one time.  It is called in two situations.  First,
;  350     0479  1 !       it is called during DEBUG initialization to set up the initial chain of
;  351     0480  1 !       Source File Control Blocks.  And second, it is called during the proc-
;  352     0481  1 !       essing of the SET SOURCE and CANCEL SOURCE commands to reset the Source
;  353     0482  1 !       Line Display feature; those commands can change the source file look-up
;  354     0483  1 !       rules and thus affect whether the currently open files are the correct
;  355     0484  1 !       files to use for source line display.
;  356     0485  1 !
;  357     0486  1 !       The routine starts by looping through all existing Source File Control
;  358     0487  1 !       Blocks (SFCBs) and closing any associated open files.  It calls routine
;  359     0488  1 !       DBG$SRC_CLOSE_FILE for each file to be closed.  It then releases all
;  360     0489  1 !       SFCBs and associated FABs and RABs on the chain.  After that, it creates
;  361     0490  1 !       a new list of SFCBs, again with associated FABs and RABs, and marks each
;  362     0491  1 !       SFCB as being not used.  The number of SFCBs it creates, which is also
;  363     0492  1 !       the maximum number of open source files henceforth, is specified by the
;  364     0493  1 !       variable DBG$SRC_MAX_FILES.  This maximum number can thus be changed by
;  365     0494  1 !       setting DBG$SRC_MAX_FILES appropriately and then calling DBG$SRC_INIT.
;  366     0495  1 !
;  367     0496  1 ! INPUTS
;  368     0497  1 !       NONE
;  369     0498  1 !
;  370     0499  1 ! OUTPUTS
;  371     0500  1 !       NONE
;  372     0501  1 !
;  373     0502  1
;  374     0503  2     BEGIN
;  375     0504  2
;  376     0505  2     LOCAL
;  377     0506  2         SFCB_BLINK: REF SFCB$BLOCK,     ! Backward link to previous SFCB
;  378     0507  2         SFCB_FLINK: REF SFCB$BLOCK,     ! Forward link to next SFCB
;  379     0508  2         SFCBPTR: REF SFCB$BLOCK;        ! Pointer to the current Source File
;  380     0509  2                                         !       Control Block (SFCB)
;  381     0510  2
;  382     0511  2
;  383     0512  2
;  384     0513  2     ! Make sure the number of new Source File Control Blocks is reasonable.
;  385     0514  2     !
;  386     0515  3     IF (.DBG$SRC_MAX_FILES LSS 1) OR (.DBG$SRC_MAX_FILES GTR MAX_MAX_FILES)
;  387     0516  2     THEN
;  388     0517  2         $DBG_ERROR('DBGSOURCE\DBG$SRC_INIT');
;  389     0518  2
;  390     0519  2
;  391     0520  2     ! Go through all current Source File Control Blocks and for each one, close
;  392     0521  2     ! the corresponding file, release the FAB, RAB, and other RMS blocks back to
;  393     0522  2     ! the memory pool, and then release the Source File Control Block itself.
;  394     0523  2     !
;  395     0524  2     SFCBPTR = .DBG$SRC_SFCB_PTR;
;  396     0525  2     SFCB_FLINK = 0;
;  397     0526  2     WHILE .SFCB_FLINK NEQ .DBG$SRC_SFCB_PTR DO
;  398     0527  3         BEGIN
;  399     0528  3         DBG$SRC_CLOSE_FILE(.SFCBPTR);
```

```
;  400     0529  3          DBG$REL_MEMORY(.SFCBPTR[SFCB$L_FABPTR]);
;  401     0530  3          DBG$REL_MEMORY(.SFCBPTR[SFCB$L_RABPTR]);
;  402     0531  3          DBG$REL_MEMORY(.SFCBPTR[SFCB$L_NAMPTR]);
;  403     0532  3          DBG$REL_MEMORY(.SFCBPTR[SFCB$L_XABDATPTR]);
;  404     0533  3          DBG$REL_MEMORY(.SFCBPTR[SFCB$L_XABFHCPTR]);
;  405     0534  3          DBG$REL_MEMORY(.SFCBPTR[SFCB$L_NAMBUFFER]);
;  406     0535  3          DBG$REL_MEMORY(.SFCBPTR[SFCB$L_RFATBLPTR]);
;  407     0536  3          SFCB_FLINK = .SFCBPTR[SFCB$L_FLINK];
;  408     0537  3          DBG$REL_MEMORY(.SFCBPTR);
;  409     0538  3          SFCBPTR = .SFCB_FLINK;
;  410     0539  2          END;
;  411     0540  2
;  412     0541  2
;  413     0542  2      ! Now loop DBG$SRC_MAX_FILES times to create that number of brand new Source
;  414     0543  2      ! File Control Blocks and the associated RMS blocks.
;  415     0544  2      !
;  416     0545  2      DBG$SRC_SFCB_PTR = 0;
;  417     0546  2      INCR I FROM 1 TO .DBG$SRC_MAX_FILES DO
;  418     0547  3          BEGIN
;  419     0548  3
;  420     0549  3
;  421     0550  3          ! Allocate memory for the Source File Control Block and mark it as
;  422     0551  3          ! being Not Used.
;  423     0552  3          !
;  424     0553  3          SFCBPTR = DBG$GET_MEMORY(SFCB$K_SIZE);
;  425     0554  3          SFCBPTR[SFCB$B_KIND] = SFCB$K_NOTUSED;
;  426     0555  3
;  427     0556  3
;  428     0557  3          ! Now allocate memory blocks for all the RMS blocks we will need for
;  429     0558  3          ! this source file.
;  430     0559  3          !
;  431     0560  3          SFCBPTR[SFCB$L_FABPTR] = DBG$GET_MEMORY(FAB$C_BLN/4);
;  432     0561  3          SFCBPTR[SFCB$L_RABPTR] = DBG$GET_MEMORY(RAB$C_BLN/4);
;  433     0562  3          SFCBPTR[SFCB$L_NAMPTR] = DBG$GET_MEMORY(NAM$C_BLN/4);
;  434     0563  3          SFCBPTR[SFCB$L_XABDATPTR] = DBG$GET_MEMORY(XAB$C_DATLEN/4);
;  435     0564  3          SFCBPTR[SFCB$L_XABFHCPTR] = DBG$GET_MEMORY(XAB$C_FHCLEN/4);
;  436     0565  3          SFCBPTR[SFCB$L_NAMBUFFER] = DBG$GET_MEMORY(2*NAMBUFSIZE/4);
;  437     0566  3          SFCBPTR[SFCB$L_RFATBLPTR] = DBG$GET_MEMORY(RFATBL_SIZE/4);
;  438     0567  3          SFCBPTR[SFCB$W_RFASPACING] = INIT_RFATBL_SPACING;
;  439     0568  3          SFCBPTR[SFCB$W_RFACURLEN] = 0;
;  440     0569  3
;  441     0570  3
;  442     0571  3          ! If this is the first new block, make DBG$SRC_SFCB_PTR point to it and
;  443     0572  3          ! doubly link it to itself.
;  444     0573  3          !
;  445     0574  3          IF .DBG$SRC_SFCB_PTR EQL 0
;  446     0575  3          THEN
;  447     0576  4              BEGIN
;  448     0577  4              SFCBPTR[SFCB$L_FLINK] = .SFCBPTR;
;  449     0578  4              SFCBPTR[SFCB$L_BLINK] = .SFCBPTR;
;  450     0579  4              DBG$SRC_SFCB_PTR = .SFCBPTR;
;  451     0580  4              END
;  452     0581  4
;  453     0582  4
;  454     0583  4          ! Otherwise, link this block into the end of the already existing doubly
;  455     0584  4          ! linked list of Source File Control Blocks.
;  456     0585  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      ! The Source File Control Block initialization is all done--now return.
;  470          0599  2      !
;  471          0600  2      RETURN;
;  472          0601  2
;  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
```

```
;  476     0604  1 GLOBAL ROUTINE DBG$SRC_LINE_TO_REC(MODRSTPTR, LINENO, NAMBLKPTR, RECNO) : NOVALUE =
;  477     0605  1 !
;  478     0606  1 ! FUNCTION
;  479     0607  1 !       This routine accepts a Module RST pointer and a line number.
;  480     0608  1 !       It translates this information to a file specification and record
;  481     0609  1 !       number within the file. It does this by scanning the source
;  482     0610  1 !       correlation records.
;  483     0611  1 !
;  484     0612  1 !       This routine is used by the DBG$NEXECUTE_EDIT routine to implement
;  485     0613  1 !       our EDITH interface.
;  486     0614  1 !
;  487     0615  1 ! INPUTS
;  488     0616  1 !       MODRSTPTR       - Module RST pointer
;  489     0617  1 !       LINENO          - The desired line number
;  490     0618  1 !
;  491     0619  1 ! OUTPUTS
;  492     0620  1 !       NAMBLKPTR       - Pointer to the name block of the current source file
;  493     0621  1 !       RECNO           - This output parameter is filled in to
;  494     0622  1 !                             be the source record within the file.
;  495     0623  1 !
;  496     0624  2     BEGIN
;  497     0625  2     MAP
;  498     0626  2         NAMBLKPTR : REF VECTOR[1,LONG], ! Returned pointer to current SFCB
;  499     0627  2         MODRSTPTR: REF RST$ENTRY;       ! Pointer to Module RST Entry of module
;  500     0628  2                                         !       containing desired source lines
;  501     0629  2
;  502     0630  2     LOCAL
;  503     0631  2         BUFFER: VECTOR[140,BYTE],       ! Output buffer for signaling
;  504     0632  2         BUF_DESC: VECTOR[2,LONG],       ! Output buffer string descriptor
;  505     0633  2         DSTPTR: REF DST$SRC_COMMAND,    ! Pointer to the Source File Correlation
;  506     0634  2                                         !       Command
;  507     0635  2         DSTREC_PTR: REF DST$RECORD,     ! Pointer to the Source File Correlation
;  508     0636  2                                         !       DST record
;  509     0637  2         HIGH_LINE_NUM,                  ! Place holder for Ending Line Number
;  510     0638  2         LENGTH,                         ! Length of current Source File Correlation
;  511     0639  2                                         !       DST Record
;  512     0640  2         LINE_NUM,                       ! The current listing line number
;  513     0641  2         MODNAMEPTR,
;  514     0642  2         MODSRCTBL: REF VECTOR[,LONG],   ! Pointer to a list of Source File
;  515     0643  2                                         !       Correlation DST Record pointers
;  516     0644  2                                         !       for the current module
;  517     0645  2         SFIT_PTR: REF SFIT$ENTRY,       ! Pointer to the Source File ID Table
;  518     0646  2         SFCBPTR: REF SFCB$BLOCK,
;  519     0647  2         SRC_FILEID_TBL,                 ! Pointer to a linked list of File ID
;  520     0648  2                                         !       Table entries for current module
;  521     0649  2         SRC_DSTPTR,                     ! Pointer to Declare Source File Command
;  522     0650  2                                         !       in DST Record
;  523     0651  2         SRC_REC;                        ! The record number in the current source
;  524     0652  2                                         !       file of the current source line
;  525     0653  2
;  526     0654  2
;  527     0655  2     ! Initialization.  Assume that Line Number is set to the beginning of the
;  528     0656  2     ! listing, Source File Record Number is set to the beginning of the
;  529     0657  2     ! source file, no Source File ID is found, Statement Mode is not set,
;  530     0658  2     ! no line is yet typed out, and statement number is set to 0.
;  531     0659  2     !
;  532     0660  2     DBG$SRC_FORMFEED_FLAG = FALSE;
```

```
;  533     0661  2     LINE_NUM = 1;
;  534     0662  2     SRC_REC = 1;
;  535     0663  2     SFIT_PTR = 0;
;  536     0664  2     SRC_DSTPTR = 0;
;  537     0665  2     SRC_FILEID_TBL = 0;
;  538     0666  2
;  539     0667  2
;  540     0668  2     ! Make sure Module Source Table pointer in module's Module RST Entry
;  541     0669  2     ! points to a list of Source File DST Record Pointers.
;  542     0670  2     !
;  543     0671  2     DBG$STA_SYMNAME(.MODRSTPTR, MODNAMEPTR);
;  544     0672  2     IF .MODRSTPTR[RST$L_MODSRCTBL] EQL 0
;  545     0673  2     THEN
;  546     0674  2         SIGNAL(DBG$_SRCLINNOT,1,.MODNAMEPTR);
;  547     0675  2
;  548     0676  2
;  549     0677  2     ! Loop thru Module's Source Line Correlation DST Records.
;  550     0678  2     !
;  551     0679  2     MODSRCTBL = .MODRSTPTR[RST$L_MODSRCTBL];
;  552     0680  2     INCR COUNT FROM 1 TO .MODSRCTBL[0] DO
;  553     0681  3         BEGIN
;  554     0682  3         IF .LINE_NUM GTR .LINENO THEN EXITLOOP;
;  555     0683  3
;  556     0684  3
;  557     0685  3         ! Get Source Line Correlation DST Record pointer.
;  558     0686  3         !
;  559     0687  3         DSTREC_PTR = .MODSRCTBL[.COUNT];
;  560     0688  3         IF .DSTREC_PTR[DST$B_TYPE] NEQ DST$K_SOURCE
;  561     0689  3         THEN
;  562     0690  3             $DBG_ERROR('DBGSOURCE\DBG$SRC_LINE_TO_REC 20');
;  563     0691  3
;  564     0692  3
;  565     0693  3         ! Scans thru Source File DST Record to decode the Source File Correlation
;  566     0694  3         ! Commands.  Pick up the length of this DST record.  Set pointer to the
;  567     0695  3         ! first command entry.
;  568     0696  3         !
;  569     0697  3         LENGTH = .DSTREC_PTR[DST$B_LENGTH];
;  570     0698  3         DSTPTR = DSTREC_PTR[DST$A_SRC_FIRST_CMD];
;  571     0699  3         WHILE (.DSTPTR = .DSTREC_PTR + 1) NEQ .LENGTH DO
;  572     0700  4             BEGIN
;  573     0701  4             IF .LINE_NUM GTR .LINENO THEN EXITLOOP;
;  574     0702  4
;  575     0703  4
;  576     0704  4             ! Pick up the command code defined in Source File DST Record
;  577     0705  4             ! and use it as a CASE index so each significant command code
;  578     0706  4             ! can be interpreted individually.
;  579     0707  4             !
;  580     0708  4             CASE .DSTPTR[DST$B_SRC_COMMAND] FROM DST$K_SRC_MIN_CMD TO
;  581     0709  4                                                  DST$K_SRC_MAX_CMD OF
;  582     0710  4                 SET
;  583     0711  4
;  584     0712  4
;  585     0713  4                 ! Declare Source File Command.  Allocate some space to build
;  586     0714  4                 ! the File ID Table.
;  587     0715  4                 !
;  588     0716  4                 [DST$K_SRC_DECLFILE]:
;  589     0717  5                     BEGIN
```

```
;  590     0718  5              IF .SFIT_PTR NEQ 0 THEN SFIT_PTR[SFIT$L_CURRECNUM] = .SRC_REC;
;  591     0719  5              SRC_REC = 1;
;  592     0720  5              SFIT_PTR = DBG$GET_TEMPMEM(SFIT$K_SIZE);
;  593     0721  5              SFIT_PTR[SFIT$L_FLINK] = .SRC_FILEID_TBL;
;  594     0722  5              SFIT_PTR[SFIT$L_FILE_ID] = .DSTPTR[DST$W_SRC_DF_FILEID];
;  595     0723  5              SFIT_PTR[SFIT$L_DSTPTR] = .DSTPTR;
;  596     0724  5              SFIT_PTR[SFIT$L_CURRECNUM] = .SRC_REC;
;  597     0725  5              SRC_FILEID_TBL = .SFIT_PTR;
;  598     0726  5              DSTPTR = .DSTPTR + .DSTPTR[DST$B_SRC_DF_LENGTH] + 2;
;  599     0727  4              END;
;  600     0728  4
;  601     0729  4
;  602     0730  4          ! Set Source File Command.  This command sets the current
;  603     0731  4          ! source file to the file denoted by the File ID given in
;  604     0732  4          ! the command.  This File ID must be pre-defined by the
;  605     0733  4          ! Declare Source File Command.
;  606     0734  4          !
;  607     0735  4          [DST$K_SRC_SETFILE]:
;  608     0736  5              BEGIN
;  609     0737  5              IF .SFIT_PTR NEQ 0 THEN SFIT_PTR[SFIT$L_CURRECNUM] = .SRC_REC;
;  610     0738  5              SRC_DSTPTR = 0;
;  611     0739  5              SFIT_PTR = .SRC_FILEID_TBL;
;  612     0740  5              WHILE .SFIT_PTR NEQ 0 DO
;  613     0741  6                  BEGIN
;  614     0742  6                  IF .DSTPTR[DST$W_SRC_UNSWORD] EQL .SFIT_PTR[SFIT$L_FILE_ID]
;  615     0743  6                  THEN
;  616     0744  7                      BEGIN
;  617     0745  7                      SRC_DSTPTR = .SFIT_PTR[SFIT$L_DSTPTR];
;  618     0746  7                      SRC_REC = .SFIT_PTR[SFIT$L_CURRECNUM];
;  619     0747  7                      EXITLOOP;
;  620     0748  6                      END;
;  621     0749  6
;  622     0750  6                  SFIT_PTR = .SFIT_PTR[SFIT$L_FLINK];
;  623     0751  5                  END;
;  624     0752  5
;  625     0753  5
;  626     0754  5              ! If this File ID has not yet been declared, signal an
;  627     0755  5              ! invalid DST Record.
;  628     0756  5              !
;  629     0757  5              IF .SRC_DSTPTR EQL 0 THEN SIGNAL(DBG$_INVDSTREC);
;  630     0758  5              DSTPTR = .DSTPTR + 3;
;  631     0759  4              END;
;  632     0760  4
;  633     0761  4
;  634     0762  4          ! Set Source Record Number Long Command.  Set the current source
;  635     0763  4          ! file record number.  Advance DSTPTR.
;  636     0764  4          !
;  637     0765  4          [DST$K_SRC_SETREC_L]:
;  638     0766  5              BEGIN
;  639     0767  5              SRC_REC = .DSTPTR[DST$L_SRC_UNSLONG];
;  640     0768  5              DSTPTR = .DSTPTR + 5;
;  641     0769  4              END;
;  642     0770  4
;  643     0771  4
;  644     0772  4          ! Set Source Record Number Word Command. (More compact form).
;  645     0773  4          !
;  646     0774  4          [DST$K_SRC_SETREC_W]:
```

```
;  647     0775  5                BEGIN
;  648     0776  5                SRC_REC = .DSTPTR[DST$W_SRC_UNSWORD];
;  649     0777  5                DSTPTR = .DSTPTR + 3;
;  650     0778  4                END;
;  651     0779  4
;  652     0780  4
;  653     0781  4              ! Set Line Number Long Command.  Set the current listing line
;  654     0782  4              ! number.  Advance DSTPTR.
;  655     0783  4              !
;  656     0784  4              [DST$K_SRC_SETLNUM_L]:
;  657     0785  5                BEGIN
;  658     0786  5                LINE_NUM = .DSTPTR[DST$L_SRC_UNSLONG];
;  659     0787  5                DSTPTR = .DSTPTR + 5;
;  660     0788  4                END;
;  661     0789  4
;  662     0790  4
;  663     0791  4              ! Set Line Number Word Command. (More compact form).
;  664     0792  4              !
;  665     0793  4              [DST$K_SRC_SETLNUM_W]:
;  666     0794  5                BEGIN
;  667     0795  5                LINE_NUM = .DSTPTR[DST$W_SRC_UNSWORD];
;  668     0796  5                DSTPTR = .DSTPTR + 3;
;  669     0797  4                END;
;  670     0798  4
;  671     0799  4
;  672     0800  4              ! Increment Line Number Byte Command.  Increment the current
;  673     0801  4              ! listing line number by a byte value given in this command.
;  674     0802  4              ! Advance DSTPTR.
;  675     0803  4              !
;  676     0804  4              [DST$K_SRC_INCRLNUM_B]:
;  677     0805  5                BEGIN
;  678     0806  5                LINE_NUM = .LINE_NUM + .DSTPTR[DST$B_SRC_UNSBYTE];
;  679     0807  5                DSTPTR = .DSTPTR + 2;
;  680     0808  4                END;
;  681     0809  4
;  682     0810  4
;  683     0811  4              ! Count Form-Feeds as Source Records Command.  This command says
;  684     0812  4              ! that form-feeds in the source should be counted as separate
;  685     0813  4              ! source records; the absence of this command says that such
;  686     0814  4              ! records should be ignored as control information and should
;  687     0815  4              ! not be assigned line nubmers.
;  688     0816  4              !
;  689     0817  4              [DST$K_SRC_FORMFEED]:
;  690     0818  5                BEGIN
;  691     0819  5                DBG$SRC_FORMFEED_FLAG = TRUE;
;  692     0820  5                DSTPTR = .DSTPTR + 1;
;  693     0821  4                END;
;  694     0822  4
;  695     0823  4
;  696     0824  4              ! Define N Separate Lines Word Command, and Define N Separate
;  697     0825  4              ! Lines Byte Command (More Compact Form).  If a Source File is not
;  698     0826  4              ! set, signal and give up.  Defines the Source File and Source
;  699     0827  4              ! Record Numbers for a specified number of Listing Line Numbers.
;  700     0828  4              ! Return if LINENO = LINE_NUM
;  701     0829  4              !
;  702     0830  4              [DST$K_SRC_DEFLINES_W, DST$K_SRC_DEFLINES_B]:
;  703     0831  5                BEGIN
```

```
;  704          0832  5          IF .SRC_DSTPTR EQL 0 THEN SIGNAL(DBG$_INVDSTREC);
;  705          0833  5
;  706          0834  5
;  707          0835  5          ! Calculate the listing line number range.
;  708          0836  5          !
;  709          0837  5          IF .DSTPTR[DST$B_SRC_COMMAND] EQL DST$K_SRC_DEFLINES_W
;  710          0838  5          THEN
;  711          0839  5              HIGH_LINE_NUM = .LINE_NUM + .DSTPTR[DST$W_SRC_UNSWORD] - 1
;  712          0840  5          ELSE
;  713          0841  5              HIGH_LINE_NUM = .LINE_NUM + .DSTPTR[DST$B_SRC_UNSBYTE] - 1;
;  714          0842  5
;  715          0843  5
;  716          0844  5          ! Test to see if the listing line number exceeds the type
;  717          0845  5          ! line range.  If it is, increments the source record number
;  718          0846  5          ! listing line number and DST pointer, then returns.
;  719          0847  5          !
;  720          0848  5          IF (.LINE_NUM LEQ .LINENO) AND
;  721          0849  6             (.LINENO LEQ .HIGH_LINE_NUM)
;  722          0850  5          THEN
;  723          0851  6              BEGIN
;  724          0852  6              MAP
;  725          0853  6                  SRC_DSTPTR: REF DST$SRC_COMMAND;
;  726          0854  6              LOCAL
;  727          0855  6                  FORMFEED_COUNT,
;  728          0856  6                  LENGTH,
;  729          0857  6                  RECORD_NUMBER;
;  730          0858  6
;  731          0859  6
;  732          0860  6              ! We have a range covering LINENO.
;  733          0861  6              ! If the language does not count formfeeds as
;  734          0862  6              ! source, then the source correlation tables
;  735          0863  6              ! are counting formfeeds differently from
;  736          0864  6              ! the editor. We need to count the number
;  737          0865  6              ! of formfeeds so we can correct for this.
;  738          0866  6              !
;  739          0867  6              RECORD_NUMBER = .SRC_REC + (.LINENO - .LINE_NUM);
;  740          0868  6              FORMFEED_COUNT = 0;
;  741          0869  6              IF NOT .DBG$SRC_FORMFEED_FLAG
;  742          0870  6              THEN
;  743          0871  7                  BEGIN
;  744          0872  7                  LOCAL
;  745          0873  7                      INPTR: BLOCK[8,BYTE],
;  746          0874  7                      OUTPTR: BLOCK[8,BYTE],
;  747          0875  7                      RFA_TABLE: REF RFATBL$BLOCKVECTOR(RFATBL_ENTRIES),
;  748          0876  7                      SRCNAM: REF $NAM_DECL,
;  749          0877  7                      SRCRAB: REF $RAB_DECL,
;  750          0878  7                      STATUS,
;  751          0879  7                      TXTRFA: VECTOR[2];
;  752          0880  7
;  753          0881  7
;  754          0882  7                  ! Open the file.
;  755          0883  7                  !
;  756          0884  7                  DBG$SRC_OPEN_FILE(.SRC_DSTPTR,
;  757          0885  7                                    SFCBPTR,
;  758          0886  7                                    FALSE);  !<< TRUE ?
;  759          0887  7
;  760          0888  7
```

```
;  761     0889  7 ! Determine whether to read from an RMS file or a module in a source library.
;  762     0890  7 !
;  763     0891  7 SRCNAM = .SFCBPTR[SFCB$L_NAMPTR];
;  764     0892  7 CASE .SFCBPTR[SFCB$B_KIND] FROM SFCB$K_RMSFILE TO SFCB$K_LBRFILE OF
;  765     0893  7     SET
;  766     0894  7
;  767     0895  7     ! Read from an RMS source file.
;  768     0896  7     !
;  769     0897  7     [SFCB$K_RMSFILE]:
;  770     0898  7         BEGIN
;  771     0899  8
;  772     0900  8         ! Set up the RAB pointer.
;  773     0901  8         !
;  774     0902  8         SRCRAB = .SFCBPTR[SFCB$L_RABPTR];
;  775     0903  8         RFA_TABLE = .SFCBPTR[SFCB$L_RFATBLPTR];
;  776     0904  8         IF .SFCBPTR[SFCB$L_CURRECNUM] EQL 1
;  777     0905  8         THEN
;  778     0906  8             BEGIN
;  779     0907  8             STATUS = $FIND(RAB = .SRCRAB);
;  780     0908  9             IF NOT .STATUS THEN SIGNAL(.STATUS);
;  781     0909  9             RFA_TABLE[0,RFATBL$L_RFA0] =
;  782     0910  9                 .SRCRAB[RAB$L_RFA0];
;  783     0911  9             RFA_TABLE[0,RFATB[$W_RFA4] =
;  784     0912  9                 .SRCRAB[RAB$W_RFA4];
;  785     0913  9             END;
;  786     0914  9
;  787     0915  8
;  788     0916  8         ! Sequentially read from the first record to the desired record
;  789     0917  8         ! number.
;  790     0918  8         !
;  791     0919  8         SFCBPTR[SFCB$L_CURRECNUM] = 1;
;  792     0920  8         STATUS = $REWIND(RAB = .SRCRAB);
;  793     0921  8         IF NOT .STATUS THEN SIGNAL(.STATUS);
;  794     0922  8         SRCRAB[RAB$B_RAC] = RAB$C_SEQ;
;  795     0923  8         INCR I FROM .SFCBPTR[SFCB$L_CURRECNUM] TO .RECORD_NUMBER DO
;  796     0924  8             BEGIN
;  797     0925  8             WHILE TRUE DO
;  798     0926  9                 BEGIN
;  799     0927  9                 STATUS = $GET(RAB = .SRCRAB);
;  800     0928 10                 IF NOT .STATUS
;  801     0929 10                 THEN
;  802     0930 10                     SIGNAL(DBG$_UNAREASRC,2,.SRCNAM[NAM$B_RSL],.SRCNAM[NAM$L_RSA
;  803     0931 10
;  804     0932 10                 ! Count records with formfeed in
;  805     0933 10                 ! formfeed_count. Do not count them
;  806     0934 10                 ! toward RECORD_COUNT.
;  807     0935 10                 !
;  808     0936 10                 IF (.DBG$SRC_REC_BUF[0] EQL 12) AND
;  809     0937 10                    (.SRCRAB[RAB$W_RSZ] EQL 1)
;  810     0938 10                 THEN
;  811     0939 10                     FORMFEED_COUNT = .FORMFEED_COUNT + 1
;  812     0940 11                 ELSE
;  813     0941 10                     EXITLOOP;
;  814     0942 10                 END;
;  815     0943 10
;  816     0944 10
;  817     0945  9
```

```
;  818          0946  9                    ! Update the current record number field ir SFCB. Set up the buffer
;  819          0947  9                    ! pointer and buffer length.
;  820          0948  9                    !
;  821          0949  9                    SFCBPTR[SFCB$L_CURRECNUM] = .RECORD_NUMBER + 1;
;  822          0950  9                    END;
;  823          0951  9                END; ! RMS file case alternative
;  824          0952  8
;  825          0953  7            ! Read from a module in a Source Library File.
;  826          0954  7            !
;  827          0955  7            [SFCB$K_LBRFILE]:
;  828          0956  7                BEGIN
;  829          0957  7
;  830          0958  7                ! If the current record number is beyond the desired record number,
;  831          0959  8                ! reset the current record to 1 and set the record file address to the
;  832          0960  8                ! beginning of the module.
;  833          0961  8                !
;  834          0962  8                IF .SFCBPTR[SFCB$L_CURRECNUM] GTR .RECORD_NUMBER
;  835          0963  8                THEN
;  836          0964  8                    BEGIN
;  837          0965  8                    TXTRFA[0] = .SFCBPTR[SFCB$L_CUR_RFA0];
;  838          0966  8                    TXTRFA[1] = .SFCBPTR[SFCB$W_CUR_RFA4];
;  839          0967  8                    SFCBPTR[SFCB$L_CURRECNUM] = 1;
;  840          0968  9                    STATUS = LBR$FIND(SFCBPTR[SFCB$L_LBRINDEX],TXTRFA);
;  841          0969  9                    IF NOT .STATUS THEN SIGNAL(.STATUS);
;  842          0970  9                    END;
;  843          0971  9
;  844          0972  9                ! Sequentially read from the current record to the desired record
;  845          0973  9                ! number.
;  846          0974  8                !
;  847          0975  8                INPTR[DSC$W_LENGTH] = SRCBUFSIZE;
;  848          0976  8                INPTR[DSC$A_POINTER] = DBG$SRC_REC_BUF[0];
;  849          0977  8                OUTPTR[DSC$A_POINTER] = DBG$SRC_REC_BUF[0];
;  850          0978  8                INCR I FROM .SFCBPTR[SFCB$L_CURRECNUM] TO .RECORD_NUMBER DO
;  851          0979  8                    BEGIN
;  852          0980  8                    WHILE TRUE DO
;  853          0981  8                        BEGIN
;  854          0982  8                        STATUS = LBR$GET_RECORD(SFCBPTR[SFCB$L_LBRINDEX],INPTR,OUTPTR);
;  855          0983  8                        IF NOT .STATUS
;  856          0984  9                        THEN
;  857          0985  9                            SIGNAL(DBG$_UNAREASRC,2,.SRCNAM[NAM$B_RSL],.SRCNAM[NAM$L_RSA
;  858          0986 10
;  859          0987 10                        IF (.DBG$SRC_REC_BUF[0] EQL 12) AND
;  860          0988 10                            (.OUTPTR[DSC$W_LENGTH] EQL 1)
;  861          0989 10                        THEN
;  862          0990 10                            FORMFEED_COUNT = .FORMFEED_COUNT + 1
;  863          0991 10                        ELSE
;  864          0992 10                            EXITLOOP;
;  865          0993 11                        END;
;  866          0994 10                    END;
;  867          0995 10
;  868          0996 10                ! Update the current record number field in SFCB.  Set up buffer
;  869          0997 10                ! pointer and buffer length.
;  870          0998  9
;  871          0999  8
;  872          1000  8
;  873          1001  8
;  874          1002  8
```

```
;  875          1003  8                        !
;  876          1004  8                        SFCBPTR[SFCB$L_CURRECNUM] = .RECORD_NUMBER + 1;
;  877          1005  7                        END;
;  878          1006  7
;  879          1007  7                    [INRANGE, OUTRANGE]:
;  880          1008  7                        $DBG_ERROR('DBGSOURCE\DBG$SRC_READ_FILE');
;  881          1009  7
;  882          1010  7                    TES;
;  883          1011  7
;  884          1012  7                RECORD_NUMBER = .RECORD_NUMBER + .FORMFEED_COUNT;
;  885          1013  6                END;
;  886          1014  6
;  887          1015  6            ! Return the record number and the NAM block pointer
;  888          1016  6            !
;  889          1017  6            NAMBLKPTR[0] = .SFCBPTR[SFCB$L_NAMPTR];
;  890          1018  6            .RECNO = .RECORD_NUMBER;
;  891          1019  6            RETURN;
;  892          1020  6            END
;  893          1021  6
;  894          1022  5        ELSE
;  895          1023  6            BEGIN
;  896          1024  6            IF .DSTPTR[DST$B_SRC_COMMAND] EQL DST$K_SRC_DEFLINES_W
;  897          1025  6            THEN
;  898          1026  6                SRC_REC = .SRC_REC + .DSTPTR[DST$W_SRC_UNSWORD]
;  899          1027  6            ELSE
;  900          1028  6                SRC_REC = .SRC_REC + .DSTPTR[DST$B_SRC_UNSBYTE];
;  901          1029  5            END;
;  902          1030  5
;  903          1031  5        LINE_NUM = .HIGH_LINE_NUM + 1;
;  904          1032  5        IF .DSTPTR[DST$B_SRC_COMMAND] EQL DST$K_SRC_DEFLINES_W
;  905          1033  5        THEN
;  906          1034  5            DSTPTR = .DSTPTR + 3
;  907          1035  5            ELSE
;  908          1036  5                DSTPTR = .DSTPTR + 2;
;  909          1037  4                END;
;  910          1038  4
;  911          1039  4
;  912          1040  4            ! Any other command code constitutes an error in the DST.
;  913          1041  4            !
;  914          1042  4            [INRANGE, OUTRANGE]:
;  915          1043  4                SIGNAL(DBG$_INVDSTREC);
;  916          1044  4
;  917          1045  4            TES;
;  918          1046  3        END;                        ! End of WHILE Loop through commands.
;  919          1047  2    END;                            ! End of INCR Loop through DST records.
;  920          1048  2
;  921          1049  2
;  922          1050  2    ! If we fall through to here, we did not find the line.
;  923          1051  2    !
;  924          1052  2    BUF_DESC[0] = 139;
;  925          1053  2    BUF_DESC[1] = BUFFER[1];
;  926          1054  2    DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC 'line '), 0);
;  927          1055  2    DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC '!UL'), .LINENO);
;  928          1056  2    DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC ' does'), 0);
;  929          1057  2    DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC ' not exist in module !AC'), .MODNAMEPTR);
;  930          1058  2    BUFFER[0] = 139 - .BUF_DESC[0];
;  931          1059  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
                                              58              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$:     BRW     5$                                ;
                                        08    AC              57  D1  00049 3$:     CMPL    LINE_NUM, LINENO                  ;
                                                              03  15  0004D         BLEQ    4$                                ;
                                                      0321    31  0004F         BRW     5$                                ;
                                              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$:     SUBL3   DSTREC_PTR, DSTPTR, R0                  ; 0699
                                                50  D7 0008B         DECL    R0                                      ;
                            24  AE              50  D1 0008D         CMPL    R0, LENGTH                              ;
                                                B3  13 0C091         BEQL    2$                                      ;
                            08  AC              57  D1 00093         CMPL    LINE_NUM, LINENO                        ; 0701
                                                AD  14 00097         BGTR    2$                                      ;
                            0F                  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 (                      BNEQ     35$                                  ;
                                 55  DD                        PUSHL    SRCRAB                               ; 0909
            00000000G  00        01  FB (                      CALLS    #1, SYS$FIND                         ;
                       59        50  D'                        MOVL     R0, STATUS                           ;
                       09        59  E8                        BLBS     STATUS, 34$                          ; 0910
                                 59  DD                        PUSHL    STATUS                               ;
            00000000G  00        01  FB                        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$:   PUSHAB  OUTPTR                                    ;
                                44  AE  9F 002EF        PUSHAB  INPTR                                     ;
                                    58  DD 002F2        PUSHL   R8                                        ;
                00000G000G      00      03  FB 002F4    CALLS   #3, LBR$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
```

```
;  934          1061  1 GLOBAL ROUTINE DBG$SRC_LNUM_RANGE(MODRSTPTR, LOWLNUM, HIGHLNUM): NOVALUE =
;  935          1062  1 !
;  936          1063  1 ! FUNCTION
;  937          1064  1 !       This routine accepts a Module RST Entry pointer as input and returns
;  938          1065  1 !       the lowest and the highest line numbers within that module as output.
;  939          1066  1 !       It is called by the screen facility to determine the line number range
;  940          1067  1 !       associated with the source display for a given module.
;  941          1068  1 !
;  942          1069  1 !       To perform its function, this routine scans through all the Source Line
;  943          1070  1 !       Correlation DST Records for the specified module from beginning to end
;  944          1071  1 !       to determine the minimum and maximum declared line numbers.
;  945          1072  1 !
;  946          1073  1 ! INPUTS
;  947          1074  1 !       MODRSTPTR - A pointer to the Module RST Entry of the module whose mini-
;  948          1075  1 !                   mum and maximum line numbers are to be looked up.
;  949          1076  1 !
;  950          1077  1 !       LOWLNUM - The address of a longword to receive the lowest line number
;  951          1078  1 !                 in the module.
;  952          1079  1 !
;  953          1080  1 !       HIGHLNUM - The address of a longword to receive the highest line number
;  954          1081  1 !                  in the module.
;  955          1082  1 !
;  956          1083  1 ! OUTPUTS
;  957          1084  1 !       LOWLNUM - The lowest line number in the module is returned to the
;  958          1085  1 !                 LOWLNUM location.
;  959          1086  1 !
;  960          1087  1 !       HIGHLNUM - The highest line number in the module is returned to the
;  961          1088  1 !                  HIGHLNUM location.
;  962          1089  1 !
;  963          1090  1
;  964          1091  2     BEGIN
;  965          1092  2
;  966          1093  2     MAP
;  967          1094  2         MODRSTPTR: REF RST$ENTRY,          ! Pointer to Module RST Entry
;  968          1095  2         LOWLNUM: REF VECTOR[1,LONG],       ! Pointer to low line number location
;  969          1096  2         HIGHLNUM: REF VECTOR[1,LONG];      ! Pointer to high line number location
;  970          1097  2
;  971          1098  2     LOCAL
;  972          1099  2         DSTPTR: REF DST$SRC_COMMAND,       ! Pointer to the Source File Correlation
;  973          1100  2                                            !       Command
;  974          1101  2         DSTREC_PTR: REF DST$RECORD,        ! Pointer to the Source File Correlation
;  975          1102  2                                            !       DST record
;  976          1103  2         HIGHEST_LNUM,                      ! Highest line number in module so far
;  977          1104  2         LENGTH,                            ! The length of the current DST record
;  978          1105  2         LINE_NUM,                          ! The current listing line number
;  979          1106  2         LOWEST_LNUM,                       ! Lowest line number in module so far
;  980          1107  2         MODSRCTBL: REF VECTOR[,LONG];      ! Pointer to the list of Source File
;  981          1108  2                                            !       Correlation DST Record pointers
;  982          1109  2                                            !       for the current module
;  983          1110  2
;  984          1111  2
;  985          1112  2
;  986          1113  2     ! Make sure Module RST pointer is not null.
;  987          1114  2     !
;  988          1115  2     IF .MODRSTPTR EQL 0
;  989          1116  2     THEN
;  990          1117  2         $DBG_ERROR('DBGSOURCE\DBG$SRC_HIGHEST_LNUM 10');
```

```
;  991          1118  2
;  992          1119  2
;  993          1120  2      ! Make sure Module Source Table pointer in module's Module RST Entry
;  994          1121  2      ! points to a list of Source File DST Record Pointers.
;  995          1122  2      !
;  996          1123  2      LOWLNUM[0] = 0;
;  997          1124  2      HIGHLNUM[0] = 0;
;  998          1125  2      IF .MODRSTPTR[RST$L_MODSRCTBL] EQL 0 THEN RETURN;
;  999          1126  2
; 1000          1127  2
; 1001          1128  2      ! Initialize flags and counters.
; 1002          1129  2      !
; 1003          1130  2      LINE_NUM = 1;
; 1004          1131  2      LOWEST_LNUM = 1000000000;
; 1005          1132  2      HIGHEST_LNUM = 0;
; 1006          1133  2
; 1007          1134  2
; 1008          1135  2      ! Loop thru Module's Source Line Correlation DST Records.
; 1009          1136  2      !
; 1010          1137  2      MODSRCTBL = .MODRSTPTR[RST$L_MODSRCTBL];
; 1011          1138  2      INCR COUNT FROM 1 TO .MODSRCTBL[0] DO
; 1012          1139  3          BEGIN
; 1013          1140  3
; 1014          1141  3
; 1015          1142  3          ! Get Source Line Correlation DST Record pointer.
; 1016          1143  3          !
; 1017          1144  3          DSTREC_PTR = .MODSRCTBL[.COUNT];
; 1018          1145  3          IF .DSTREC_PTR[DST$B_TYPE] NEQ DST$K_SOURCE
; 1019          1146  3          THEN
; 1020          1147  3              $DBG_ERROR('DBGSOURCE\DBG$SRC_HIGHEST_LNUM 20');
; 1021          1148  3
; 1022          1149  3
; 1023          1150  3          ! Scan through the Source File DST Record to decode the Source File
; 1024          1151  3          ! Correlation Commands.  Pick up the length of this DST record and set
; 1025          1152  3          ! DSTPTR to point to the first command entry.
; 1026          1153  3          !
; 1027          1154  3          LENGTH = .DSTREC_PTR[DST$B_LENGTH];
; 1028          1155  3          DSTPTR = DSTREC_PTR[DST$A_SRC_FIRST_CMD];
; 1029          1156  3          WHILE (.DSTPTR = .DSTREC_PTR) LEQ .LENGTH DO
; 1030          1157  4              BEGIN
; 1031          1158  4
; 1032          1159  4
; 1033          1160  4              ! Pick up the command code defined in Source File DST Record
; 1034          1161  4              ! and use it as a CASE index so each significant command code
; 1035          1162  4              ! can be interpreted individually.
; 1036          1163  4              !
; 1037          1164  4              CASE .DSTPTR[DST$B_SRC_COMMAND] FROM DST$K_SRC_MIN_CMD TO
; 1038          1165  4                                                   DST$K_SRC_MAX_CMD OF
; 1039          1166  4                  SET
; 1040          1167  4
; 1041          1168  4
; 1042          1169  4                  ! Declare Source File Command.
; 1043          1170  4                  !
; 1044          1171  4                  [DST$K_SRC_DECLFILE]:
; 1045          1172  4                      DSTPTR = .DSTPTR + .DSTPTR[DST$B_SRC_DF_LENGTH] + 2;
; 1046          1173  4
; 1047          1174  4
```

```
;  1048          1175  4              ! Set Source File Command.
;  1049          1176  4              !
;  1050          1177  4              [DST$K_SRC_SETFILE]:
;  1051          1178  4                  DSTPTR = .DSTPTR + 3;
;  1052          1179  4
;  1053          1180  4
;  1054          1181  4              ! Set Source Record Number Long Command.
;  1055          1182  4              !
;  1056          1183  4              [DST$K_SRC_SETREC_L]:
;  1057          1184  4                  DSTPTR = .DSTPTR + 5;
;  1058          1185  4
;  1059          1186  4
;  1060          1187  4              ! Set Source Record Number Word Command.
;  1061          1188  4              !
;  1062          1189  4              [DST$K_SRC_SETREC_W]:
;  1063          1190  4                  DSTPTR = .DSTPTR + 3;
;  1064          1191  4
;  1065          1192  4
;  1066          1193  4              ! Set Line Number Long Command.  Set the current listing line
;  1067          1194  4              ! number.  Advance DSTPTR.
;  1068          1195  4              !
;  1069          1196  4              [DST$K_SRC_SETLNUM_L]:
;  1070          1197  5                  BEGIN
;  1071          1198  5                  LINE_NUM = .DSTPTR[DST$L_SRC_UNSLONG];
;  1072          1199  5                  DSTPTR = .DSTPTR + 5;
;  1073          1200  4                  END;
;  1074          1201  4
;  1075          1202  4
;  1076          1203  4              ! Set Line Number Word Command. (More compact form).
;  1077          1204  4              !
;  1078          1205  4              [DST$K_SRC_SETLNUM_W]:
;  1079          1206  5                  BEGIN
;  1080          1207  5                  LINE_NUM = .DSTPTR[DST$W_SRC_UNSWORD];
;  1081          1208  5                  DSTPTR = .DSTPTR + 3;
;  1082          1209  4                  END;
;  1083          1210  4
;  1084          1211  4
;  1085          1212  4              ! Increment Line Number Byte Command.  Increment the current
;  1086          1213  4              ! listing line number by a byte value given in this command.
;  1087          1214  4              ! Advance DSTPTR.
;  1088          1215  4              !
;  1089          1216  4              [DST$K_SRC_INCRLNUM_B]:
;  1090          1217  5                  BEGIN
;  1091          1218  5                  LINE_NUM = .LINE_NUM + .DSTPTR[DST$B_SRC_UNSBYTE];
;  1092          1219  5                  DSTPTR = .DSTPTR + 2;
;  1093          1220  4                  END;
;  1094          1221  4
;  1095          1222  4
;  1096          1223  4              ! Do not skip the form feed.
;  1097          1224  4              !
;  1098          1225  4              [DST$K_SRC_FORMFEED]:
;  1099          1226  4                  DSTPTR = .DSTPTR + 1;
;  1100          1227  4
;  1101          1228  4
;  1102          1229  4              ! Define N Separate Lines Word Command.
;  1103          1230  4              !
;  1104          1231  4              [DST$K_SRC_DEFLINES_W]:
```

```
;  1105          1232  5                  BEGIN
;  1106          1233  5                  LOWEST_LNUM = MIN(.LOWEST_LNUM, .LINE_NUM);
;  1107          1234  5                  LINE_NUM = .LINE_NUM + .DSTPTR[DST$W_SRC_UNSWORD];
;  1108          1235  5                  HIGHEST_LNUM = MAX(.HIGHEST_LNUM, .LINE_NUM - 1);
;  1109          1236  5                  DSTPTR = .DSTPTR + 3;
;  1110          1237  4                  END;
;  1111          1238  4
;  1112          1239  4
;  1113          1240  4                  ! Define N Separate Lines Byte Command.
;  1114          1241  4                  !
;  1115          1242  4                  [DST$K_SRC_DEFLINES_B]:
;  1116          1243  5                      BEGIN
;  1117          1244  5                      LOWEST_LNUM = MIN(.LOWEST_LNUM, .LINE_NUM);
;  1118          1245  5                      LINE_NUM = .LINE_NUM + .DSTPTR[DST$B_SRC_UNSBYTE];
;  1119          1246  5                      HIGHEST_LNUM = MAX(.HIGHEST_LNUM, .LINE_NUM - 1);
;  1120          1247  5                      DSTPTR = .DSTPTR + 2;
;  1121          1248  4                      END;
;  1122          1249  4
;  1123          1250  4
;  1124          1251  4                  ! Any other command code constitutes an error in the DST.
;  1125          1252  4                  !
;  1126          1253  4                  [INRANGE, OUTRANGE]:
;  1127          1254  5                      BEGIN
;  1128          1255  5                      SIGNAL(DBG$_INVDSTREC);
;  1129          1256  4                      END;
;  1130          1257  4
;  1131          1258  4                  TES;
;  1132          1259  4
;  1133          1260  3              END;                        ! End of WHILE Loop through commands.
;  1134          1261  3
;  1135          1262  2          END;                            ! End of INCR Loop through DST records.
;  1136          1263  2
;  1137          1264  2
;  1138          1265  2      ! Return the lowest and highest line numbers that we found.  Then return
;  1139          1266  2      ! to the caller.
;  1140          1267  2      !
;  1141          1268  2      IF .LOWEST_LNUM GTR .HIGHEST_LNUM THEN LOWEST_LNUM = 0;
;  1142          1269  2      LOWLNUM[0] = .LOWEST_LNUM;
;  1143          1270  2      HIGHLNUM[0] = .HIGHEST_LNUM;
;  1144          1271  2      RETURN;
;  1145          1272  2
;  1146          1273  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$SRC_LNUM_RANGE, Save R2,R3,R4,R5,R6,R7,-;  1061
                                                            R8,R9,R10
                            5A  00000000G   00  9E 00002          MOVAB   LIB$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
```

```
;1148        1274  1 ROUTINE DBG$SRC_OPEN(SFCB_KIND,SRCDSTPTR,PROPER_FLAG) =
;1149        1275  1 !
;1150        1276  1 ! FUNCTION
;1151        1277  1 !       This routine is called by DBG$SRC_OPEN_FILE.  It actually does the
;1152        1278  1 !       open for RMS or LBR file.
;1153        1279  1 !
;1154        1280  1 !       The opened file is then checked to see if it is the right file.  This
;1155        1281  1 !       consists of checking the Creation Date and Time and several other file
;1156        1282  1 !       attributes.  If this search fails to find the desired file, the proper
;1157        1283  1 !       flag is set to indicate the status.  If file could not be open at all,
;1158        1284  1 !       then the returned status from RMS routines is returned.
;1159        1285  1 !
;1160        1286  1 !       If the desired file is a module within a source library, then checking
;1161        1287  1 !       is modified to check the library in the given directory which has a
;1162        1288  1 !       module of the desired name which in turn has the proper insertion date.
;1163        1289  1 !       If this checking fails to find the desired module in the library,
;1164        1290  1 !       the proper flag is set to indicate the proper module is not found.
;1165        1291  1 !       If no library at all could be opened, then status from LBR is returned.
;1166        1292  1 !
;1167        1293  1 ! INPUTS
;1168        1294  1 !       SFCB_KIND - Indictates whether the file to be opened is an RMS file
;1169        1295  1 !                   (SFCB_KIND = 2) or an LBR file (SFCB_KIND = 3).
;1170        1296  1 !
;1171        1297  1 !       SRCDSTPTR - Pointer to the Declare Source File DST command which declares
;1172        1298  1 !                   the file to be opened.
;1173        1299  1 !
;1174        1300  1 ! OUTPUTS
;1175        1301  1 !       PROPER_FLAG - 0: Proper file is found,
;1176        1302  1 !                     non-0: file or module in the library is not proper file.
;1177        1303  1 !
;1178        1304  1 !                     The "proper" is defined as the first version with the
;1179        1305  1 !                     same creation date and time, the same file size, and
;1180        1306  1 !                     the same record format and file organization as specified
;1181        1307  1 !                     in the Source File Correlation DST Record.  The "proper"
;1182        1308  1 !                     file is found means such file is found during the search
;1183        1309  1 !                     directory dir1, ..., dirN process.  If the "proper"
;1184        1310  1 !                     version of the file is not found in any of dir1, ..., dirN,
;1185        1311  1 !                     the latest version of the file in the first directory on
;1186        1312  1 !                     the list to have a version of it is used.  A warning is
;1187        1313  1 !                     issued.  If no file can be found at all, a warning
;1188        1314  1 !                     message to this effect is issued.
;1189        1315  1 !
;1190        1316  1 !       STATUS - returned value from RMS or LBR, return value for this routine.
;1191        1317  1 !
;1192        1318  1
;1193        1319  2     BEGIN
;1194        1320  2
;1195        1321  2     MAP
;1196        1322  2         PROPER_FLAG: REF VECTOR[1],      ! Flag set to indicate the status of
;1197        1323  2                                          !       the opened file
;1198        1324  2         SRCDSTPTR: REF DST$SRC_COMMAND;  ! Pointer to the Declare Source File
;1199        1325  2                                          !       DST command
;1200        1326  2
;1201        1327  2
;1202        1328  2     LOCAL
;1203        1329  2         BUFDESC: BLOCK[8,BYTE],          ! String descriptor, it contains length
;1204        1330  2                                          !       and address of the buffer
```

```
;  1205          1331  2          BUFLEN,                          ! Contains the length of the returned
;  1206          1332  2                                           !      module header
;  1207          1333  2          DSTCDT: REF VECTOR[2],           ! Pointer to the creation date and time
;  1208          1334  2                                           !      in the DST Declare Source File
;  1209          1335  2                                           !      command
;  1210          1336  2          LIBMOD_PTR: REF DST$SRC_CMDTRLR, ! Points to the Source Library Module
;  1211          1337  2                                           !      name in the Source File
;  1212          1338  2                                           !      Correlation DST Record
;  1213          1339  2          MHDPTR: REF BLOCK[,BYTE],        ! Points to Module Header
;  1214          1340  2          MODPTR: BLOCK[8,BYTE],           ! Length and address of module name
;  1215          1341  2          SRCXABDAT: REF $XABDAT_DECL,     ! XABDAT for RMS file
;  1216          1342  2          SRCXABFHC: REF $XABFHC_DECL,     ! XABFHC for RMS file
;  1217          1343  2          STATUS,                          ! RMS status or LBR status
;  1218          1344  2          TXTRFA: VECTOR[2],               ! The RFA of the module text in library
;  1219          1345  2          XABCDT: REF VECTOR[2],           ! Pointer to the creation date and time
;  1220          1346  2                                           !      in the RMS XAB block
;  1221          1347  2          XABRDT: REF VECTOR[2];           ! Pointer to the revision date and time
;  1222          1348  2                                           !      in the RMS XAB block
;  1223          1349  2
;  1224          1350  2
;  1225          1351  2
;  1226          1352  2      ! Determine whether we have an RMS or LBR file.
;  1227          1353  2      !
;  1228          1354  2      PROPER_FLAG[0] = 0;
;  1229          1355  2      CASE .SFCB_KIND FROM SFCB$K_RMSFILE TO SFCB$K_LBRFILE OF
;  1230          1356  2          SET
;  1231          1357  2
;  1232          1358  2
;  1233          1359  2          ! Open an RMS source file and check its attributes to see if this is
;  1234          1360  2          ! the exact file the DST Declare Source File Command specifies.
;  1235          1361  2          !
;  1236          1362  2          [SFCB$K_RMSFILE]:
;  1237          1363  3              BEGIN
;  1238          1364  3
;  1239          1365  3
;  1240          1366  3              ! Open the file and connect the RAB to the FAB.
;  1241          1367  3              !
;  1242          1368  3              STATUS = $OPEN(FAB = .DBG$SRC_SFCB_PTR[SFCB$L_FABPTR]);
;  1243          1369  3              IF NOT .STATUS THEN RETURN .STATUS;
;  1244          1370  3              STATUS = $CONNECT(RAB = .DBG$SRC_SFCB_PTR[SFCB$L_RABPTR]);
;  1245          1371  3              IF NOT .STATUS THEN RETURN .STATUS;
;  1246          1372  3
;  1247          1373  3
;  1248          1374  3              ! Check that the file has the proper creation date and time.  (We
;  1249          1375  3              ! also accept a matching revision date and time for compatibility
;  1250          1376  3              ! with the usage prior to VMS V4.0.)  If not, set PROPER_FLAG to 1.
;  1251          1377  3              !
;  1252          1378  3              SRCXABDAT = .DBG$SRC_SFCB_PTR[SFCB$L_XABDATPTR];
;  1253          1379  3              SRCXABFHC = .DBG$SRC_SFCB_PTR[SFCB$L_XABFHCPTR];
;  1254          1380  3              XABCDT = SRCXABDAT[XAB$Q_CDT];
;  1255          1381  3              XABRDT = SRCXABDAT[XAB$Q_RDT];
;  1256          1382  3              DSTCDT = SRCDSTPTR[DST$Q_SRC_DF_RMS_CDT];
;  1257          1383  3              IF (.XABCDT[0] NEQ .DSTCDT[0] OR .XABCDT[1] NEQ .DSTCDT[1]) AND
;  1258          1384  4                 (.XABRDT[0] NEQ .DSTCDT[0] OR .XABRDT[1] NEQ .DSTCDT[1])
;  1259          1385  3              THEN
;  1260          1386  3                  PROPER_FLAG[0] = 1;
;  1261          1387  3
```

```
;  1262          1388  3
;  1263          1389  3              ! Also check that the file's byte size and record format and file
;  1264          1390  3              ! organization is as specified in the DST.  If not, set PROPER_FLAG
;  1265          1391  3              ! to 2, 3, or 4.
;  1266          1392  3              !
;  1267          1393  3              IF .SRCXABFHC[XAB$L_EBK] NEQ .SRCDSTPTR[DST$L_SRC_DF_RMS_EBK]
;  1268          1394  3              THEN
;  1269          1395  3                  PROPER_FLAG[0] = 2;
;  1270          1396  3
;  1271          1397  3              IF .SRCXABFHC[XAB$W_FFB] NEQ .SRCDSTPTR[DST$W_SRC_DF_RMS_FFB]
;  1272          1398  3              THEN
;  1273          1399  3                  PROPER_FLAG[0] = 3;
;  1274          1400  3
;  1275          1401  3              IF .SRCXABFHC[XAB$B_RFO] NEQ .SRCDSTPTR[DST$B_SRC_DF_RMS_RFO]
;  1276          1402  3              THEN
;  1277          1403  3                  PROPER_FLAG[0] = 4;
;  1278          1404  3
;  1279          1405  2              END;
;  1280          1406  2
;  1281          1407  2
;  1282          1408  2          ! Open an Source Library File.  Then look up and open the desired
   1283          1409  2          ! module within that source library.  Also compare the module's
;  1284          1410  2          ! inseration date and time with that in the DST Declare Source
;  1285          1411  2          ! File command.
;  1286          1412  2          !
;  1287          1413  2          [SFCB$K_LBRFILE]:
;  1288          1414  3              BEGIN
;  1289          1415  3
;  1290          1416  3
;  1291          1417  3              ! Open the desired source library.
;  1292          1418  3              !
;  1293          1419  3              STATUS = LBR$INI_CONTROL(DBG$SRC_SFCB_PTR[SFCB$L_LBRINDEX],
;  1294          1420  3                  UPLIT(LBR$C_READ), UPLIT(LBR$C_TYP_TXT), .DBG$SRC_SFCB_PTR[SFCB$L_NAMPTR]);
;  1295          1421  3              IF NOT .STATUS THEN RETURN .STATUS;
;  1296          1422  3              STATUS = LBR$OPEN(DBG$SRC_SFCB_PTR[SFCB$L_LBRINDEX]);
;  1297          1423  3              IF NOT .STATUS THEN RETURN .STATUS;
;  1298          1424  3
;  1299          1425  3
;  1300          1426  3              ! Look up the desired module and open the module header.
;  1301          1427  3              !
;  1302          1428  3              LIBMOD_PTR
;  1303          1429  3                  = SRCDSTPTR[DST$A_SRC_DF_FILENAME] + .SRCDSTPTR[DST$B_SRC_DF_FILENAME];
;  1304          1430  3              MODPTR[DSC$W_LENGTH] = .LIBMOD_PTR[DST$B_SRC_DF_LIBMODNAME];
;  1305          1431  3              MODPTR[DSC$A_POINTER] = LIBMOD_PTR[DST$A_SRC_DF_LIBMODNAME];
;  1306          1432  3              STATUS = LBR$LOOKUP_KEY(DBG$SRC_SFCB_PTR[SFCB$L_LBRINDEX],MODPTR,TXTRFA);
;  1307          1433  3              IF NOT .STATUS THEN RETURN .STATUS;
;  1308          1434  3
;  1309          1435  3
;  1310          1436  3              ! Get module's header information.
;  1311          1437  3              !
;  1312          1438  3              BUFDESC[DSC$W_LENGTH] = SRCBUFSIZE;
;  1313          1439  3              BUFDESC[DSC$A_POINTER] = DBG$SRC_REC_BUF[0];
;  1314          1440  3              STATUS = LBR$SET_MODULE(DBG$SRC_SFCB_PTR[SFCB$L_LBRINDEX],TXTRFA,
;  1315          1441  3                  BUFDESC,BUFLEN);
;  1316          1442  3              IF NOT .STATUS THEN RETURN .STATUS;
;  1317          1443  3
;  1318          1444  3
```

```
;  1319          1445  3            ! Save the module's start Record File Address (RFA) in the Source
;  1320          1446  3            ! File Control Block.
;  1321          1447  3            !
;  1322          1448  3            DBG$SRC_SFCB_PTR[SFCB$L_CUR_RFA0] = .TXTRFA[0];
;  1323          1449  3            DBG$SRC_SFCB_PTR[SFCB$W_CUR_RFA4] = .TXTRFA[1];
;  1324          1450  3
;  1325          1451  3
;  1326          1452  3            ! Check the module insertion date and time.  If it is not the same
;  1327          1453  3            ! as that in the DST Declare Source File Command, set PROPER_FLAG
;  1328          1454  3            ! to 1.
;  1329          1455  3            !
;  1330          1456  3            MHDPTR = DBG$SRC_REC_BUF[0];
;  1331          1457  3            XABCDT = MHDPTR[MHD$L_DATIM];
;  1332          1458  3            DSTCDT = SRCDSTPTR[DST$Q_SRC_DF_RMS_CDT];
;  1333          1459  3            IF .XABCDT[0] NEQ .DSTCDT[0] OR .XABCDT[1] NEQ .DSTCDT[1]
;  1334          1460  3            THEN
;  1335          1461  3                PROPER_FLAG[0] = 1;
;  1336          1462  3
;  1337          1463  2            END;
;  1338          1464  2
;  1339          1465  2
;  1340          1466  2        ! Any other kind of Source File Control Block is an internal error.
;  1341          1467  2        !
;  1342          1468  2        [INRANGE, OUTRANGE]:
;  1343          1469  2            $DBG_ERROR('DBGSOURCE\DBG$SRC_OPEN');
;  1344          1470  2
;  1345          1471  2        TES;
;  1346          1472  2
;  1347          1473  2
;  1348          1474  2    ! The source file or module was successfully opened although the file
;  1349          1475  2    ! attributes (date and time, etc.) may not have matched those in the
;  1350          1476  2    ! DST.  Return the last status we got from RMS or LBR.
;  1351          1477  2    !
;  1352          1478  2    RETURN .STATUS;
;  1353          1479  2
;  1354          1480  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
                           78        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$SRC_SFCB_PTR, R2                          ; 1378
                           51   24   A2  D0 00057          MOVL    36(R2), SRCXABDAT
                           55   28   A2  D0 0005B          MOVL    40(R2), SRCXABFHC                             ; 1379
                           53   14   A1  9E 0005F          MOVAB   20(R1), XABCDT                                ; 1380
                           51        0C  C0 00063          ADDL2   #12, XABRDT                                   ; 1381
                           54   08   AC  D0 00066          MOVL    SRCDSTPTR, R4                                 ; 1382
                           52   05   A4  9F 0006A          MOVAB   5(R4), DSTCDT
                           62        63  D1 0006E          CMPL    (XABCDT), (DSTCDT)                            ; 1383
                                     07  12 00071          BNEQ    3$
                     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
```

```
;  1357     1482  1 ROUTINE DBG$SRC_OPEN_FILE(FILDSTPTR, SFCBPTR, STEP_FLAG): NOVALUE =
;  1358     1483  1 !
;  1359     1484  1 !  FUNCTION
;  1360     1485  1 !       This routine opens a specified source file (unless the file is already
;  1361     1486  1 !       opened) and sets up the corresponding Source File Control Block so that
;  1362     1487  1 !       the file can be read with subsequent calls on DBG$SRC_READ_FILE.  In the
;  1363     1488  1 !       process, it applies all source directory search rules to make sure it
;  1364     1489  1 !       opens the file in the directory desired by the user.
;  1365     1490  1 !
;  1366     1491  1 !       The routine starts by searching the list of Source File Control Blocks
;  1367     1492  1 !       for the input DST pointer.  If such a control block is found, the file
;  1368     1493  1 !       is already open and the Source File Control Block (SFCB) pointer can be
;  1369     1494  1 !       returned immediately.  The SFCB is moved to the Most Recently Used posi-
;  1370     1495  1 !       tion in the doubly linked SFCB list before DBG$SRC_OPEN_FILE returns.
;  1371     1496  1 !
;  1372     1497  1 !       Otherwise, an SFCB must be allocated for this file.  This is done by
;  1373     1498  1 !       finding the Least Recently Used SFCB and closing the file using that
;  1374     1499  1 !       control block (if any).  The SFCB is then moved to the Most Recently
;  1375     1500  1 !       Used position.  Next, the Source Directory Search List for the current
;  1376     1501  1 !       module (which may be the default search list) is found in the linked
;  1377     1502  1 !       list pointed to by DBG$SRC_DIR_LIST.
;  1378     1503  1 !
;  1379     1504  1 !       At this point, the routine is ready to search all directories on the
;  1380     1505  1 !       list to locate the desired source file.  For each directory, it does
;  1381     1506  1 !       a $PARSE and loops over a $SEARCH system service.  Each file found by
;  1382     1507  1 !       $SEARCH is checked to see if it is the right file.  This consists of
;  1383     1508  1 !       checking the Creation Date and Time and several other file attributes.
;  1384     1509  1 !       If this search fails to find the desired file, the first file returned
;  1385     1510  1 !       by $SEARCH for the first directory on the list is used even though its
;  1386     1511  1 !       date and time or other attributes may be wrong.  An informational
;  1387     1512  1 !       message is signalled in this case.  If no file at all could be found,
;  1388     1513  1 !       an error message to that effect is signalled.
;  1389     1514  1 !
;  1390     1515  1 !       If the desired file is a module within a source library, the search is
;  1391     1516  1 !       modified to find the first library in the directory list which has a
;  1392     1517  1 !       module of the desired name which in turn has the proper insertion date.
;  1393     1518  1 !
;  1394     1519  1 !       If no Source Directory Search List was specified for this module (either
;  1395     1520  1 !       directly or by default), the file name as it appears in the Declare
;  1396     1521  1 !       Source File DST command is taken to be the file we want to open.
;  1397     1522  1 !
;  1398     1523  1 !       Once the desired file has been identified, that file is opened via the
;  1399     1524  1 !       RMS $OPEN and $CONNECT services.  Or, if the source file is a module in
;  1400     1525  1 !       a source library, the library and its module are opened via the appro-
;  1401     1526  1 !       priate LIBRARIAN (LBR$xxx) calls.  DBG$SRC_OPEN_FILE then returns.
;  1402     1527  1 !
;  1403     1528  1 !       The routine actually does the open and verify the attributes is in
;  1404     1529  1 !       "DBG$SRC_OPEN".
;  1405     1530  1 !
;  1406     1531  1 !  INPUTS
;  1407     1532  1 !       FILDSTPTR - A pointer to the Declare Source File command in the Source
;  1408     1533  1 !                   File Correlation DST Record which declares the source file to
;  1409     1534  1 !                   be opened.  This command includes the desired file name and
;  1410     1535  1 !                   other file attributes.
;  1411     1536  1 !
;  1412     1537  1 !       SFCBPTR - A longword location to receive a pointer to the Source File
;  1413     1538  1 !                 Control Block set up for the desired file.
```

```
;1414    1539  1 !
;1415    1540  1 !       STEP_FLAG - Flag set to true to indicate the call is from STEP command.
;1416    1541  1 !
;1417    1542  1 ! OUTPUTS
;1418    1543  1 !       SFCBPTR - A Pointer to the Source File Control Block set up for the
;1419    1544  1 !                 opened source file is returned to SFCBPTR.  This pointer
;1420    1545  1 !                 can then be used when later calling DBG$SRC_READ_FILE.
;1421    1546  1 !
;1422    1547  1
;1423    1548  2     BEGIN
;1424    1549  2
;1425    1550  2     MAP
;1426    1551  2         FILDSTPTR:                          ! Pointer to the Declare Source File DST
;1427    1552  2                 REF DST$SRC_COMMAND,        !      command which declares the file
;1428    1553  2                                             !      to be opened
;1429    1554  2         SFCBPTR: REF VECTOR[1];             ! Pointer to longword to receive the
;1430    1555  2                                             !      Source File Control Block pointer
;1431    1556  2
;1432    1557  2     LOCAL
;1433    1558  2         BAD_FILE,                           ! Flag to indicate there is a bad
;1434    1559  2                                             !      file stuck in the SFCB
;1435    1560  2         BLINK: REF SFCB$BLOCK,              ! Backward link
;1436    1561  2         DEFAULT: VECTOR[SRCBUFSIZE,BYTE],   !      Contains Default File Name
;1437    1562  2                                             !           length and filename for RMS
;1438    1563  2         FIRST_FILE: VECTOR[SRCBUFSIZE,BYTE],!      Contains first file returned FROM
;1439    1564  2                                             !           $SEARCH the directory list
;1440    1565  2         FIRST_TIME,                         ! Flag set to indicate first time $SEARCH
;1441    1566  2         FLINK: REF SFCB$BLOCK,              ! Forward link
;1442    1567  2         LIBMOD_PTR: REF DST$SRC_CMDTRLR,    ! Trailer field in the Declare
;1443    1568  2                                             !      Source File Command
;1444    1569  2         LRUPTR: REF SFCB$BLOCK,             ! Pointer to Least Recently Used Source
;1445    1570  2                                             !      File Control Block
;1446    1571  2         PROPER_FLAG,                        ! Flag set to indicate proper file/module
;1447    1572  2                                             !      is used
;1448    1573  2         SDSL_PTR: REF SDSL$HEADER,          ! Pointer to Source Directory Search
;1449    1574  2                                             !      List Header Block
;1450    1575  2         SDSL_DIRPTR: REF SDSL$ENTRY,        ! Pointer to Source Directory Search
;1451    1576  2                                             !      List Entry
;1452    1577  2         SFCB_KIND,                          ! Indicate SFCB used for RMS file or
;1453    1578  2                                             !      Source Library file
;1454    1579  2         SRCFAB: REF $FAB_DECL,              ! FAB for RMS file
;1455    1580  2         SRCNAM: REF $NAM_DECL,              ! NAM for RMS file
;1456    1581  2         STATUS;                             ! RMS routine status
;1457    1582  2
;1458    1583  2
;1459    1584  2
;1460    1585  2     ! In here there is a case that the file may not be available.
;1461    1586  2     ! If this file is not available, then signal the way
;1462    1587  2     ! out now, when the handler catches this signal, unwinds, so the normal
;1463    1588  2     ! operation will continue on -- for example, if stepping by inst., we do
;1464    1589  2     ! not want to see the message coming out for each step when the file
;1465    1590  2     ! is not available for the source.
;1466    1591  2     !
;1467    1592  2     IF .STEP_FLAG
;1468    1593  2     THEN
;1469    1594  3         BEGIN
;1470    1595  3         IF (.DBG$SRC_SFCB_PTR[SFCB$B_KIND] EQL SFCB$K_FILE_UNAVAIL) AND
```

```
;1471    1596  4            (.DBG$SRC_SFCB_PTR[SFCB$L_DSTPTR] EQL .FILDSTPTR)
;1472    1597  3        THEN
;1473    1598  3            SIGNAL(DBG$_FILEUNAL);
;1474    1599  3
;1475    1600  2        END;
;1476    1601  2
;1477    1602  2
;1478    1603  2        ! Check to see if the given file is the Most Recently Used file.  If it
;1479    1604  2        ! is, simply return.  There is no need to reorder the SFCB list.
;1480    1605  2        !
;1481    1606  3        IF (.DBG$SRC_SFCB_PTR[SFCB$B_KIND] EQL SFCB$K_RMSFILE OR
;1482    1607  2            .DBG$SRC_SFCB_PTR[SFCB$B_KIND] EQL SFCB$K_LBRFILE) AND
;1483    1608  3            (.DBG$SRC_SFCB_PTR[SFCB$L_DSTPTR] EQL .FILDSTPTR)
;1484    1609  2        THEN
;1485    1610  3            BEGIN
;1486    1611  3            SFCBPTR[0] = .DBG$SRC_SFCB_PTR;
;1487    1612  3            RETURN;
;1488    1613  2            END;
;1489    1614  2
;1490    1615  2
;1491    1616  2        ! Start at the location pointed to by DBG$SRC_SFCB_PTR and circle thru the
;1492    1617  2        ! doubly linked list SFCB to see if the desired file is already on the
;1493    1618  2        ! list.  If it is already on the list, move it's Source File Control Block
;1494    1619  2        ! to the Most Recently Used position (which is pointed to by DBG$SRC_SFCB_PTR)
;1495    1620  2        ! and then return.
;1496    1621  2        !
;1497    1622  2        BAD_FILE = FALSE;
;1498    1623  2        LRUPTR = .DBG$SRC_SFCB_PTR[SFCB$L_BLINK];
;1499    1624  2        WHILE .LRUPTR NEQ .DBG$SRC_SFCB_PTR DO
;1500    1625  3            BEGIN
;1501    1626  3            IF (.LRUPTR[SFCB$B_KIND] NEQ SFCB$K_NOTUSED) AND
;1502    1627  4                (.LRUPTR[SFCB$L_DSTPTR] EQL .FILDSTPTR)
;1503    1628  3            THEN
;1504    1629  4                BEGIN
;1505    1630  4
;1506    1631  4
;1507    1632  4                ! The file already has a Source File Control Block.  Unlink it
;1508    1633  4                ! from the SFCB List.
;1509    1634  4                !
;1510    1635  4                BLINK = .LRUPTR[SFCB$L_BLINK];
;1511    1636  4                FLINK = .LRUPTR[SFCB$L_FLINK];
;1512    1637  4                BLINK[SFCB$L_FLINK] = .FLINK;
;1513    1638  4                FLINK[SFCB$L_BLINK] = .BLINK;
;1514    1639  4
;1515    1640  4
;1516    1641  4                ! Insert the SFCB at the Most Recently Used position.  Then return
;1517    1642  4                ! its address to the caller.
;1518    1643  4                !
;1519    1644  4                BLINK = .DBG$SRC_SFCB_PTR[SFCB$L_BLINK];
;1520    1645  4                LRUPTR[SFCB$L_FLINK] = .DBG$SRC_SFCB_PTR;
;1521    1646  4                LRUPTR[SFCB$L_BLINK] = .BLINK;
;1522    1647  4                BLINK[SFCB$L_FLINK] = .LRUPTR;
;1523    1648  4                DBG$SRC_SFCB_PTR[SFCB$L_BLINK] = .LRUPTR;
;1524    1649  4                DBG$SRC_SFCB_PTR = .LRUPTR;
;1525    1650  4
;1526    1651  4
;1527    1652  4                ! Check to see if this has been marked as unavailable file.
```

```
;  1528          1653  4              !
;  1529          1654  4              IF .DBG$SRC_SFCB_PTR[SFCB$B_KIND] EQL SFCB$K_FILE_UNAVAIL
;  1530          1655  4              THEN
;  1531          1656  5                  BEGIN
;  1532          1657  5                  IF .STEP_FLAG
;  1533          1658  5                  THEN
;  1534          1659  5                      SIGNAL(DBG$_FILEUNAL)
;  1535          1660  5
;  1536          1661  5                  ELSE
;  1537          1662  6                      BEGIN
;  1538          1663  6                      BAD_FILE = TRUE;
;  1539          1664  6                      EXITLOOP;
;  1540          1665  5                      END;
;  1541          1666  5
;  1542          1667  5                  END
;  1543          1668  5
;  1544          1669  4              ELSE
;  1545          1670  5                  BEGIN
;  1546          1671  5                  SFCBPTR[0] = .DBG$SRC_SFCB_PTR;
;  1547          1672  5                  RETURN;
;  1548          1673  4                  END;
;  1549          1674  4
;  1550          1675  3              END;
;  1551          1676  3
;  1552          1677  3          LRUPTR = .LRUPTR[SFCB$L_BLINK];
;  1553          1678  2          END;
;  1554          1679  2
;  1555          1680  2
;  1556          1681  2      ! We have circled the whole doubly linked list (the given file is not
;  1557          1682  2      ! already on the list).  Advance the pointer to take the Least Recently
;  1558          1683  2      ! Used SFCB and close the file using that control block (if any).
;  1559          1684  2      !
;  1560          1685  2      ! If we locate a bad file on the list, we want to force it to execute
;  1561          1686  2      ! the code to get the bad status out to the user.  So this is the case
;  1562          1687  2      ! we are by passing the IF.
;  1563          1688  2      !
;  1564          1689  2      IF NOT .BAD_FILE
;  1565          1690  2      THEN
;  1566          1691  3          BEGIN
;  1567          1692  3          DBG$SRC_SFCB_PTR = .DBG$SRC_SFCB_PTR[SFCB$L_BLINK];
;  1568          1693  3          IF .DBG$SRC_SFCB_PTR[SFCB$B_KIND] NEQ SFCB$K_NOTUSED
;  1569          1694  3          THEN
;  1570          1695  3              DBG$SRC_CLOSE_FILE(.DBG$SRC_SFCB_PTR);
;  1571          1696  2          END;
;  1572          1697  2
;  1573          1698  2
;  1574          1699  2      ! Determine whether the desired file is an RMS or LBR file.
;  1575          1700  2      !
;  1576          1701  2      LIBMOD_PTR =
;  1577          1702  2          FI[DSTPTR[DST$A_SRC_DF_FILENAME] + .FILDSTPTR[DST$B_SRC_DF_FILENAME];
;  1578          1703  2      IF .LIBMOD_PTR[DST$B_SRC_DF_LIBMODNAME] EQL 0
;  1579          1704  2      THEN
;  1580          1705  2          SFCB_KIND = SFCB$K_RMSFILE
;  1581          1706  2      ELSE
;  1582          1707  2          SFCB_KIND = SFCB$K_LBRFILE;
;  1583          1708  2
;  1584          1709  2
```

```
;  1585          1710  2      ! Initialize RMS source file control block.
;  1586          1711  2      !
;  1587        P 1712  2      $FAB_INIT(FAB = .DBG$SRC_SFCB_PTR[SFCB$L_FABPTR],
;  1588        P 1713  2                FAC = <GET>,
;  1589        P 1714  2                FOP = <NAM>,
;  1590        P 1715  2                NAM = .DBG$SRC_SFCB_PTR[SFCB$L_NAMPTR],
;  1591          1716  2                XAB = .DBG$SRC_SFCB_PTR[SFCB$L_XABDATPTR]);
;  1592        P 1717  2      $RAB_INIT(RAB = .DBG$SRC_SFCB_PTR[SFCB$L_RABPTR],
;  1593        P 1718  2                FAB = .DBG$SRC_SFCB_PTR[SFCB$L_FABPTR],
;  1594        P 1719  2                UBF = DBG$SRC_REC_BUF[0],
;  1595          1720  2                USZ = MIN(SRCBUFSIZE,255));
;  1596        P 1721  2      $NAM_INIT(NAM = .DBG$SRC_SFCB_PTR[SFCB$L_NAMPTR],
;  1597        P 1722  2                ESA = .DBG$SRC_SFCB_PTR[SFCB$L_NAMBUFFER],
;  1598        P 1723  2                ESS = MIN(NAMBUFSIZE,255),
;  1599        P 1724  2                RSA = .DBG$SRC_SFCB_PTR[SFCB$L_NAMBUFFER] + NAMBUFSIZE,
;  1600          1725  2                RSS = MIN(NAMBUFSIZE,255));
;  1601        P 1726  2      $XABDAT_INIT(XAB = .DBG$SRC_SFCB_PTR[SFCB$L_XABDATPTR],
;  1602          1727  2                   NXT = .DBG$SRC_SFCB_PTR[SFCB$L_XABFHCPTR]);
;  1603          1728  2      $XABFHC_INIT(XAB = .DBG$SRC_SFCB_PTR[SFCB$L_XABFHCPTR]);
;  1604          1729  2      SRCFAB = .DBG$SRC_SFCB_PTR[SFCB$L_FABPTR];
;  1605          1730  2      SRCNAM = .DBG$SRC_SFCB_PTR[SFCB$L_NAMPTR];
;  1606          1731  2      PROPER_FLAG = 0;
;  1607          1732  2
;  1608          1733  2
;  1609          1734  2      ! Set up default file name to be the same as that in the Source File
;  1610          1735  2      ! Correlation DST Record except that we change the version number to '*'.
;  1611          1736  2      !
;  1612          1737  2      DEFAULT[0] = .FILDSTPTR[DST$B_SRC_DF_FILENAME];
;  1613          1738  2      CH$MOVE(.DEFAULT[0],FILDSTPTR[DST$A_SRC_DF_FILENAME],DEFAULT[1]);
;  1614          1739  2      DECR I FROM .DEFAULT[0] TO 1 DO
;  1615          1740  3          BEGIN
;  1616          1741  3          IF .DEFAULT[.I] EQL ';'
;  1617          1742  3          THEN
;  1618          1743  4              BEGIN
;  1619          1744  4              DEFAULT[.I + 1] = '*';
;  1620          1745  4              DEFAULT[0] = .I + 1;
;  1621          1746  4              EXITLOOP;
;  1622          1747  3              END;
;  1623          1748  2          END;
;  1624          1749  2
;  1625          1750  2      SRCFAB[FAB$L_DNA] = DEFAULT[1];
;  1626          1751  2      SRCFAB[FAB$B_DNS] = .DEFAULT[0];
;  1627          1752  2
;  1628          1753  2
;  1629          1754  2      ! Search the Source Directory Search List so we can open the given file in
;  1630          1755  2      ! the proper directory.  Search all the Header Blocks for the Module RST
;  1631          1756  2      ! Entry pointer for the current module.  If the desired module is not found,
;  1632          1757  2      ! the Header Block with the zero Module RST Entry pointer is used.
;  1633          1758  2      !
;  1634          1759  2      SDSL_DIRPTR = 0;
;  1635          1760  2      SDSL_PTR = .DBG$SRC_DIR_LIST;
;  1636          1761  2      WHILE .SDSL_PTR NEQ 0 DO
;  1637          1762  3          BEGIN
;  1638          1763  3          IF .SDSL_PTR[SDSL$L_MODPTR] EQL .DBG$SRC_MODRSTPTR
;  1639          1764  3          THEN
;  1640          1765  4              BEGIN
;  1641          1766  4              SDSL_DIRPTR = .SDSL_PTR[SDSL$L_LIST_PTR];
```

```
;  1642     1767  4              IF .SDSL_DIRPTR EQL 0
;  1643     1768  4              THEN
;  1644     1769  4                  $DBG_ERROR('DBGSOURCE\DBG$SRC_OPEN_FILE 10');
;  1645     1770  4
;  1646     1771  4              EXITLOOP;
;  1647     1772  3              END;
;  1648     1773  3
;  1649     1774  3          IF .SDSL_PTR[SDSL$L_MODPTR] EQL 0
;  1650     1775  3          THEN
;  1651     1776  4              BEGIN
;  1652     1777  4              SDSL_DIRPTR = .SDSL_PTR[SDSL$L_LIST_PTR];
;  1653     1778  4              IF .SDSL_DIRPTR EQL 0
;  1654     1779  4              THEN
;  1655     1780  4                  $DBG_ERROR('DBGSOURCE\DBG$SRC_OPEN_FILE 20');
;  1656     1781  4
;  1657     1782  3              END;
;  1658     1783  3
;  1659     1784  3          SDSL_PTR = .SDSL_PTR[SDSL$L_FLINK];
;  1660     1785  2          END;
;  1661     1786  2
;  1662     1787  2      ! If no Source Directory Search List entry is present, the file name from
;  1663     1788  2      ! the Declare Source File DST command is used as is.  We thus open the
;  1664     1789  2      ! file and if this succeeds, return to the caller.
;  1665     1790  2      !
;  1666     1791  2      IF .SDSL_DIRPTR EQL 0
;  1667     1792  2      THEN
;  1668     1793  2          BEGIN
;  1669     1794  3
;  1670     1795  3
;  1671     1796  3          ! Set up the exact DST file name in the FAB.
;  1672     1797  3          !
;  1673     1798  3          SRCFAB[FAB$L_DNA] = 0;
;  1674     1799  3          SRCFAB[FAB$B_DNS] = 0;
;  1675     1800  3          SRCFAB[FAB$L_FNA] = FILDSTPTR[DST$A_SRC_DF_FILENAME];
;  1676     1801  3          SRCFAB[FAB$B_FNS] = .FILDSTPTR[DST$B_SRC_DF_FILENAME];
;  1677     1802  3          !
;  1678     1803  3          ! Do the $PARSE to initiate the $SEARCH loop.
;  1679     1804  3          !
;  1680     1805  3          STATUS = $PARSE(FAB = .SRCFAB);
;  1681     1806  3          IF NOT .STATUS
;  1682     1807  3          THEN
;  1683     1808  3              BEGIN
;  1684     1809  4              IF .STEP_FLAG
;  1685     1810  4              THEN
;  1686     1811  5                  BEGIN
;  1687     1812  5                  DBG$SRC_SFCB_PTR[SFCB$B_KIND] = SFCB$K_FILE_UNAVAIL;
;  1688     1813  5                  DBG$SRC_SFCB_PTR[SFCB$L_DSTPTR] = .FILDSTPTR;
;  1689     1814  5                  DBG$SRC_SFCB_PTR[SFCB$L_CURRECNUM] = 1;
;  1690     1815  5                  DBG$SRC_SFCB_PTR[SFCB$W_RFASPACING] = INIT_RFATBL_SPACING;
;  1691     1816  5                  DBG$SRC_SFCB_PTR[SFCB$W_RFACURLEN] = 0;
;  1692     1817  5                  END;
;  1693     1818  4
;  1694     1819  4              SIGNAL(DBG$_UNAOPNSRC,2,.SRCNAM[NAM$B_ESL],.SRCNAM[NAM$L_ESA],.STATUS);
;  1695     1820  4
;  1696     1821  4              END;
;  1697     1822  4
;  1698     1823  3          END;
```

```
;  1699          1824  3
;  1700          1825  3
;  1701          1826  3          ! Do $SEARCH to initate the NAME block.
;  1702          1827  3          !
;  1703          1828  3          STATUS = $SEARCH(FAB = .SRCFAB);
;  1704          1829  3          IF NOT .STATUS
;  1705          1830  3          THEN
;  1706          1831  4              BEGIN
;  1707          1832  4              IF .STEP_FLAG
;  1708          1833  4              THEN
;  1709          1834  5                  BEGIN
;  1710          1835  5                  DBG$SRC_SFCB_PTR[SFCB$B_KIND] = SFCB$K_FILE_UNAVAIL;
;  1711          1836  5                  DBG$SRC_SFCB_PTR[SFCB$L_DSTPTR] = .FILDSTPTR;
;  1712          1837  5                  DBG$SRC_SFCB_PTR[SFCB$L_CURRECNUM] = 1;
;  1713          1838  5                  DBG$SRC_SFCB_PTR[SFCB$W_RFASPACING] = INIT_RFATBL_SPACING;
;  1714          1839  5                  DBG$SRC_SFCB_PTR[SFCB$W_RFACURLEN] = 0;
;  1715          1840  4                  END;
;  1716          1841  4
;  1717          1842  4              SIGNAL(DBG$_UNAOPNSRC,2,.SRCNAM[NAM$B_ESL],.SRCNAM[NAM$L_ESA],.STATUS);
;  1718          1843  4
;  1719          1844  3              END;
;  1720          1845  3
;  1721          1846  3
;  1722          1847  3          ! Open the file and check the status.
;  1723          1848  3          !
;  1724          1849  3          STATUS = DBG$SRC_OPEN(.SFCB_KIND,.FILDSTPTR,PROPER_FLAG);
;  1725          1850  3          IF NOT .STATUS
;  1726          1851  3          THEN
;  1727          1852  3              SIGNAL(DBG$_UNAOPNSRC,2,.SRCNAM[NAM$B_ESL],.SRCNAM[NAM$L_ESA],.STATUS);
;  1728          1853  3
;  1729          1854  3
;  1730          1855  3              ! Check to see if we got the exact version of the desired file.  If not,
;  1731          1856  3              ! issue the warning message.
;  1732          1857  3              !
;  1733          1858  4              IF (.PROPER_FLAG NEQ 0)
;  1734          1859  3              THEN
;  1735          1860  3                  SIGNAL(DBG$_NOTORIGSRC,2,.SRCNAM[NAM$B_RSL],.SRCNAM[NAM$L_RSA]);
;  1736          1861  3
;  1737          1862  3
;  1738          1863  3              ! The open succeeded.  Fill in the source file control block and return
;  1739          1864  3              ! to the caller.
;  1740          1865  3              !
;  1741          1866  3              DBG$SRC_SFCB_PTR[SFCB$B_KIND] = .SFCB_KIND;
;  1742          1867  3              DBG$SRC_SFCB_PTR[SFCB$L_DSTPTR] = .FILDSTPTR;
;  1743          1868  3              DBG$SRC_SFCB_PTR[SFCB$L_CURRECNUM] = 1;
;  1744          1869  3              DBG$SRC_SFCB_PTR[SFCB$W_RFASPACING] = INIT_RFATBL_SPACING;
;  1745          1870  3              DBG$SRC_SFCB_PTR[SFCB$W_RFACURLEN] = 0;
;  1746          1871  3              SFCBPTR[0] = .DBG$SRC_SFCB_PTR;
;  1747          1872  3              RETURN;
;  1748          1873  2              END;
;  1749          1874  2
;  1750          1875  2
;  1751          1876  2          ! The found Header Block pointer then points to a list of directories.
;  1752          1877  2          ! Search this list to find the desired file in one of the directories on
;  1753          1878  2          ! that list.
;  1754          1879  2          !
;  1755          1880  2          FIRST_TIME = TRUE;
```

```
; 1756        1881  2     WHILE .SDSL_DIRPTR NEQ 0 DO
; 1757        1882  3         BEGIN
; 1758        1883  3
; 1759        1884  3
; 1760        1885  3         ! Make the current directory name the primary file name in the FAB.
; 1761        1886  3         ! Then do the $PARSE to initiate the $SEARCH loop.
; 1762        1887  3         !
; 1763        1888  3         $RCFAB[FAB$L_FNA] = SDSL_DIRPTR[SDSL$A_ENT_DIRNAME];
; 1764        1889  3         $RCFAB[FAB$B_FNS] = .SDSL_DIRPTR[SDSL$B_ENT_DIRLEN];
; 1765        1890  3         STATUS = $PARSE(FAB = .$RCFAB);
; 1766        1891  4         IF NOT ((.STATUS EQL RMS$_DNF) OR (.STATUS))
; 1767        1892  3         THEN
; 1768        1893  3             SIGNAL(DBG$_UNAOPNSRC,2,.SRCNAM[NAM$B_ESL],.SRCNAM[NAM$L_ESA],.STATUS);
; 1769        1894  3
; 1770        1895  3         IF .STATUS
; 1771        1896  3         THEN
; 1772        1897  4             BEGIN
; 1773        1898  4
; 1774        1899  4
; 1775        1900  4             ! Do wild-card processing by going through a $SEARCH loop.  For
; 1776        1901  4             ! each file found by $SEARCH, we see if it is the file we want.
; 1777        1902  4             !
; 1778        1903  4             WHILE TRUE DO
; 1779        1904  5                 BEGIN
; 1780        1905  5
; 1781        1906  5
; 1782        1907  5                 ! Get the next file from $SEARCH.
; 1783        1908  5                 !
; 1784        1909  5                 STATUS = $SEARCH(FAB = .$RCFAB);
; 1785        1910  6                 IF (.STATUS EQL RMS$_FNF) OR (.STATUS EQL RMS$_NMF)
; 1786        1911  5                 THEN
; 1787        1912  5                     EXITLOOP;
; 1788        1913  5                 IF NOT .STATUS
; 1789        1914  5                 THEN
; 1790        1915  5                     SIGNAL(DBG$_UNAOPNSRC,2,.SRCNAM[NAM$B_ESL],.SRCNAM[NAM$L_ESA],
; 1791        1916  5                             .STATUS);
; 1792        1917  5
; 1793        1918  5
; 1794        1919  5                 ! Try to open that file and see if it is the exact file we
; 1795        1920  5                 ! want.  If we find the exact file we want, exit the search
; 1796        1921  5                 ! loop.
; 1797        1922  5                 !
; 1798        1923  5                 STATUS = DBG$SRC_OPEN(.$FCB_KIND,.FILDSTPTR,PROPER_FLAG);
; 1799        1924  5                 IF NOT .STATUS
; 1800        1925  5                 THEN
; 1801        1926  5                     SIGNAL(DBG$_UNAOPNSRC,2,.SRCNAM[NAM$B_ESL],.SRCNAM[NAM$L_ESA],
; 1802        1927  5                             .STATUS);
; 1803        1928  5
; 1804        1929  5                 IF .PROPER_FLAG EQL 0 THEN EXITLOOP;
; 1805        1930  5
; 1806        1931  5
; 1807        1932  5                 ! The first time we find any file with the right name, we remember
; 1808        1933  5                 ! that file's name even if it file characteristics are wrong.
; 1809        1934  5                 !
; 1810        1935  5                 IF .FIRST_TIME
; 1811        1936  5                 THEN
; 1812        1937  6                     BEGIN
```

```
;  1813          1938  6                 FIRST_FILE[0] = .SRCFAB[FAB$B_FNS];
;  1814          1939  6                 CH$MOVE(.FIRST_FILE[0],.SRCFAB[FAB$L_FNA],FIRST_FILE[1]);
;  1815          1940  6                 FIRST_TIME = FALSE;
;  1816          1941  5                 END;
;  1817          1942  5
;  1818          1943  5
;  1819          1944  5                 ! It is not the right file - close it and loop for another file.
;  1820          1945  5                 !
;  1821          1946  5                 DBG$SRC_SFCB_PTR[SFCB$B_KIND] = .SFCB_KIND;
;  1822          1947  5                 DBG$SRC_CLOSE_FILE(.DBG$SRC_SFCB_PTR);
;  1823          1948  5
;  1824          1949  4                 END;              ! End of $SEARCH WHILE loop.
;  1825          1950  4
;  1826          1951  3             END;                  ! End of if on good $PARSE status.
;  1827          1952  3
;  1828          1953  3
;  1829          1954  3             ! If we have not yet found the desired file, get the next Source
;  1830          1955  3             ! Directory Search List Entry and loop to try it.
;  1831          1956  3             !
;  1832          1957  3             IF .STATUS AND (.PROPER_FLAG EQL 0) THEN EXITLOOP;
;  1833          1958  3             SDSL_DIRPTR = .SDSL_DIRPTR[SDSL$L_ENT_FLINK];
;  1834          1959  2             END;
;  1835          1960  2
;  1836          1961  2
;  1837          1962  2         ! If no proper file is found, the first file for the first directory
;  1838          1963  2         ! on the list is used, so try to open it again.
;  1839          1964  2         !
;  1840          1965  3         IF (.PROPER_FLAG NEQ 0) AND (NOT .FIRST_TIME)
;  1841          1966  2         THEN
;  1842          1967  3             BEGIN
;  1843          1968  3             SRCFAB[FAB$L_FNA] = FIRST_FILE[1];
;  1844          1969  3             SRCFAB[FAB$B_FNS] = .FIRST_FILE[0];
;  1845          1970  3             STATUS = $PARSE(FAB = .SRCFAB);
;  1846          1971  3             IF NOT .STATUS THEN SIGNAL(.STATUS);
;  1847          1972  3             STATUS = $SEARCH(FAB = .SRCFAB);
;  1848          1973  3             IF NOT .STATUS THEN SIGNAL(.STATUS);
;  1849          1974  3             STATUS = DBG$SRC_OPEN(.SFCB_KIND,.FILDSTPTR,PROPER_FLAG);
;  1850          1975  3             IF NOT .STATUS THEN SIGNAL(.STATUS);
;  1851          1976  2             END;
;  1852          1977  2
;  1853          1978  2
;  1854          1979  2         ! If the status returned from RMS is not 'no more files' or good status,
;  1855          1980  2         ! signal the bad status.
;  1856          1981  2         !
;  1857          1982  2         IF NOT ((.STATUS EQL RMS$_NMF) OR (.STATUS))
;  1858          1983  2         THEN
;  1859          1984  3             BEGIN
;  1860          1985  3             IF .STEP_FLAG
;  1861          1986  3             THEN
;  1862          1987  4                 BEGIN
;  1863          1988  4                 DBG$SRC_SFCB_PTR[SFCB$B_KIND] = SFCB$K_FILE_UNAVAIL;
;  1864          1989  4                 DBG$SRC_SFCB_PTR[SFCB$L_DSTPTR] = .FILDSTPTR;
;  1865          1990  4                 DBG$SRC_SFCB_PTR[SFCB$L_CURRECNUM] = 1;
;  1866          1991  4                 END;
;  1867          1992  3
;  1868          1993  3             SIGNAL(DBG$_UNAOPNSRC,2,.SRCNAM[NAM$B_ESL],.SRCNAM[NAM$L_ESA],.STATUS);
;  1869          1994  2             END;
```

```
;  1870          1995  2
;  1871          1996  2
;  1872          1997  2      ! Issue the warning message, if the exact file is not found.
;  1873          1998  2      !
;  1874          1999  3      IF (.PROPER_FLAG NEQ 0)
;  1875          2000  2      THEN
;  1876          2001  2          SIGNAL(DBG$_NOTORIGSRC,2,.SRCNAM[NAM$B_RSL],.SRCNAM[NAM$L_RSA]);
;  1877          2002  2
;  1878          2003  2
;  1879          2004  2      ! We have now successfully opened a source file in the search directory.
;  1880          2005  2      ! Fill in the Source File Control Block and return.
;  1881          2006  2      !
;  1882          2007  2      DBG$SRC_SFCB_PTR[SFCB$B_KIND] = .SFCB_KIND;
;  1883          2008  2      DBG$SRC_SFCB_PTR[SFCB$L_DSTPTR] = .FI[DSTPTR;
;  1884          2009  2      DBG$SRC_SFCB_PTR[SFCB$L_CURRECNUM] = 1;
;  1885          2010  2      SFCBPTR[0] = .DBG$SRC_SFCB_PTR;
;  1886          2011  2      RETURN;
;  1887          2012  2
;  1888          2013  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
000000000G      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$SRC_SFCB_PTR, R0                    ; 1692
00000000'       04        EF   A0   D0 000C1        MOVL    4(R0), DBG$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), 1?(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$:     CLRL    48(SRCFAB)                              ; 1799
                        2C    A9  15  AB 9E 0023D      MOVAB   21(R11), 44(SRCFAB)                     ; 1801
                        34    A9  04  AE 9B 00242      MOVZBW  4(SP), 52(SRCFAB)                       ; 1802
                                 59  DD 00247          PUSHL   SRCFAB                                  ; 1807
                  00000000G   00    FB 00249          CALLS   #1, SYS$PARSE                           ;
                              57    D0 00250          MOVL    R0, STATUS                              ;
                              32    E8 00253          BLBS    STATUS, 21$                             ; 1808
                              17    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    FB 00281          CALLS   #5, LIB$SIGNAL                          ;
                                 59  DD 00288 21$:     PUSHL   SRCFAB                                  ; 1828
                  00000000G   00    FB 0028A          CALLS   #1, SYS$SEARCH                          ;
                              57    D0 00291          MOVL    R0, STATUS                              ;
                              32    E8 00294          BLBS    STATUS, 23$                             ; 1829
                              17    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    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$SRC_OPEN                        ;
                              57    D0 002D6          MOVL    R0, STATUS                              ;
                              18    E8 002D9          BLBS    STATUS, 24$                             ; 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    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
```

```
;1891        2015  1 ROUTINE DBG$SRC_OUTPUT_LINE(LINE_NUM, STMT_NUM, BUFPTR, BUFLEN): NOVALUE =
;1892        2016  1 !
;1893        2017  1 ! FUNCTION
;1894        2018  1 !       This routine formats and outputs one source line.  The actual text of
;1895        2019  1 !       the source line is passed in as a parameter.  The line's line number and
;1896        2020  1 !       statement number are also passed in.  This routine then formats the line
;1897        2021  1 !       number and statement number in front of the line's text and outputs the
;1898        2022  1 !       line.  The statement number is omitted if it is zero (as it will be for
;1899        2023  1 !       most languages).  The output for lines 12, 123, 1234, 123456, 1234567,
;1900        2024  1 !       12345.6, and 123.4 would look as follows:
;1901        2025  1 !
;1902        2026  1 !                               12: TEXT OF LINE 12
;1903        2027  1 !                              123: TEXT OF LINE 123
;1904        2028  1 !                             1234: TEXT OF LINE 1234
;1905        2029  1 !                           123456: TEXT OF LINE 123456
;1906        2030  1 !                          1234567: TEXT OF LINE 1234567
;1907        2031  1 !                          12345.6: TEXT OF LINE 12345, STATEMENT 6
;1908        2032  1 !                           123.4: TEXT OF LINE 123, STATEMENT 4
;1909        2033  1 !
;1910        2034  1 !       Note that all lines line up unless the line number and statement number
;1911        2035  1 !       require more that six characters.
;1912        2036  1 !
;1913        2037  1 !       This routine also folds long source lines when they would make the out-
;1914        2038  1 !       put line exceed terminal width in length.  The following format is used:
;1915        2039  1 !
;1916        2040  1 !                             123: TEXT OF LINE 123, FIRST PART
;1917        2041  1 !                               -: TEXT OF LINE 123, SECOND PART
;1918        2042  1 !                               -: TEXT OF LINE 123, LAST PART
;1919        2043  1 !
;1920        2044  1 !       The string "-: " thus serves as a continuation marker.  Note that this
;1921        2045  1 !       routine must expand tabs to spaces in the source text.
;1922        2046  1 !
;1923        2047  1 !       User can use the SET MARGIN Command to set the display window size of
;1924        2048  1 !       the source text.  For example, if user gives SET MARGIN 10:40, user
;1925        2049  1 !       will see the source text from the 10th character to the 40th character
;1926        2050  1 !       displayed on the terminal.  However, the window size is adjused by the
;1927        2051  1 !       the length of the source record (after the expension of TABs), and the
;1928        2052  1 !       terminal width set by the SET TERM/WIDTH = n Command.
;1929        2053  1 !
;1930        2054  1 !       The final output lines are printed using DBG$PRINT in the normal way.
;1931        2055  1 !
;1932        2056  1 ! INPUTS
;1933        2057  1 !       LINE_NUM - The line number of the source line to be printed.
;1934        2058  1 !
;1935        2059  1 !       STMT_NUM - The statement number of the source line to be printed.
;1936        2060  1 !                  If there is no statement number, STMT_NUM must be zero.
;1937        2061  1 !
;1938        2062  1 !       BUFPTR   - A pointer to a buffer which contains the ASCII text of the
;1939        2063  1 !                  source line to be printed.
;1940        2064  1 !
;1941        2065  1 !       BUFLEN   - The length in characters of the source line pointed to by
;1942        2066  1 !                  BUFPTR.
;1943        2067  1 !
;1944        2068  1 ! OUTPUTS
;1945        2069  1 !       NONE
;1946        2070  1 !
;1947        2071  1
```

```
;  1948          2072  2     BEGIN
;  1949          2073  2
;  1950          2074  2     MAP
;  1951          2075  2         BUFPTR: REF VECTOR[,BYTE];      ! Pointer to the ASCII image to print
;  1952          2076  2
;  1953          2077  2     BIND
;  1954          2078  2         SPACES = UPLIT BYTE (%ASCIC '        '): VECTOR[,BYTE];
;  1955          2079  2                                          A vector of 8 spaces
;  1956          2080  2
;  1957          2081  2     LOCAL
;  1958          2082  2         FST_LINE_FLAG,                 ! Flag set to indicate this output line
;  1959          2083  2                                        !     has a line number
;  1960          2084  2         BGNPTR: REF VECTOR[,BYTE],     ! Pointer to the begin search tab position
;  1961          2085  2                                        !     in source text buffer
;  1962          2086  2         FORMFEED_PTR: REF VECTOR[,BYTE],! Pointer to the formfeed character in
;  1963          2087  2                                        !     output line
;  1964          2088  2         GAP,                           ! Distance between the begin pointer and
;  1965          2089  2                                        !     the tab position
;  1966          2090  2         LENGTH,                        ! Search tab range length
;  1967          2091  2         LEFT_MARGIN,                   ! Source Text left column
;  1968          2092  2         RIGHT_MARGIN,                  ! Source Text right column
;  1969          2093  2         TABCNT,                        ! Number of tab characters
;  1970          2094  2         TABLEN,                        ! Spaces needed for the TAB character
;  1971          2095  2         TAB_POSITION,                  ! Point to the tab character
;  1972          2096  2         TMPBUF: VECTOR[SRCBUFSIZE,BYTE],! Expanded source text buffer
;  1973          2097  2         TMPBUFPTR,                     ! Pointer to the Expanded source text buffer
;  1974          2098  2         TERM_WIDTH,                    ! Terminal size - 8
;  1975          2099  2         WINDOW_WIDTH;                  ! The size of the source text to be displayed
;  1976          2100  2
;  1977          2101  2
;  1978          2102  2
;  1979          2103  2     ! Replace form feed character by space character.
;  1980          2104  2     !
;  1981          2105  2     BGNPTR = .BUFPTR;
;  1982          2106  2     LENGTH = .BUFLEN;
;  1983          2107  2     WHILE TRUE DO
;  1984          2108  3         BEGIN
;  1985          2109  3         FORMFEED_PTR = CH$FIND_CH(.LENGTH, .BGNPTR, 12);
;  1986          2110  3         IF .FORMFEED_PTR EQL 0 THEN EXITLOOP;
;  1987          2111  3         FORMFEED_PTR[0] = %C' ';
;  1988          2112  3         LENGTH = .LENGTH - (.FORMFEED_PTR - .BGNPTR + 1);
;  1989          2113  3         BGNPTR = .FORMFEED_PTR + 1;
;  1990          2114  2         END;
;  1991          2115  2
;  1992          2116  2
;  1993          2117  2     ! Expand the TAB character to spaces in the buffer which contains the
;  1994          2118  2     ! ASCII text of the source line.  Finally adjust the length of the original
;  1995          2119  2     ! buffer length.
;  1996          2120  2     !
;  1997          2121  2     TMPBUFPTR = TMPBUF;
;  1998          2122  2     BGNPTR = .BUFPTR;
;  1999          2123  2     LENGTH = .BUFLEN;
;  2000          2124  2     TABCNT = 0;
;  2001          2125  2     WHILE TRUE DO
;  2002          2126  3         BEGIN
;  2003          2127  3
;  2004          2128  3
```

```
;  2005          2129  3          ! Locate the 'TAB' character in the buffer.  Pointed to by BGNPTR as
;  2006          2130  3          ! the start search position, LENGTH is the given search range.
;  2007          2131  3          !
;  2008          2132  3          TAB_POSITION = CH$FIND_CH(.LENGTH,.BGNPTR,%C'    ');
;  2009          2133  3
;  2010          2134  3
;  2011          2135  3          ! There is no TAB character can be found, move the source text from
;  2012          2136  3          ! BGNPTR position to TMPBUFPTR positior.  Then exit the serach TAB loop.
;  2013          2137  3          !
;  2014          2138  3          IF .TAB_POSITION EQL 0
;  2015          2139  3          THEN
;  2016          2140  4              BEGIN
;  2017          2141  4              CH$MOVE(.LENGTH, .BGNPTR, .TMPBUFPTR);
;  2018          2142  4              EXITLOOP;
;  2019          2143  4              END;
;  2020          2144  3
;  2021          2145  3
;  2022          2146  3          ! Count the number of the TAB characters in the source text.
;  2023          2147  3          !
;  2024          2148  3          TABCNT = .TABCNT + 1;
;  2025          2149  3
;  2026          2150  3
;  2027          2151  3          ! Move the source text before the TAB from BGNPTR postion to TMPBUFPTR
;  2028          2152  3          ! position.
;  2029          2153  3          !
;  2030          2154  3          GAP = .TAB_POSITION - .BGNPTR;
;  2031          2155  3          IF .GAP NEQ 0 THEN CH$MOVE(.GAP, .BGNPTR, .TMPBUFPTR);
;  2032          2156  3          TMPBUFPTR = .TMPBUFPTR + .GAP;
;  2033          2157  3
;  2034          2158  3
;  2035          2159  3          ! Calculate the number of spaces, this TAB character shcuLd expand to.
;  2036          2160  3          ! Then move that number of spaces to the TMPBUFPTR position.
;  2037          2161  3          !
;  2038          2162  3          TABLEN = .SPACES[0] - ((.TMPBUFPTR - TMPBUF) MOD .SPACES[0]);
;  2039          2163  3          CH$MOVE(.TABLEN , SPACES[1], .TMPBUFPTR);
;  2040          2164  3          TMPBUFPTR = .TMPBUFPTR + .TABLEN;
;  2041          2165  3
;  2042          2166  3
;  2043          2167  3          ! Update the source text record length by the number of spaces added.
;  2044          2168  3          ! Adjust the search TAB range.  Skip the TAB charcter in the original
;  2045          2169  3          ! source text buffer.
;  2046          2170  3          !
;  2047          2171  3          BUFLEN = .BUFLEN + .TABLEN;
;  2048          2172  3          LENGTH = .LENGTH - .GAP - 1;
;  2049          2173  3          IF .LENGTH EQL 0 THEN EXITLOOP;
;  2050          2174  3          BGNPTR = .TAB_POSITION + 1;
;  2051          2175  2          END;
;  2052          2176  2
;  2053          2177  2
;  2054          2178  2      ! Expansion is all done.  Adjust the source record length by taking off the
;  2055          2179  2      ! number of TABs encounted.
;  2056          2180  2      !
;  2057          2181  2      BUFLEN = .BUFLEN  - .TABCNT;
;  2058          2182  2
;  2059          2183  2
;  2060          2184  2      ! Adjust the window margin by the length of the source text.
;  2061          2185  2      !
```

```
; 2062          2186  2      LEFT_MARGIN = MAX(1, .DBG$SRC_LEFT_MARGIN);
; 2063          2187  2      RIGHT_MARGIN = MIN(.BUFLEN,.DBG$SRC_RIGHT_MARGIN);
; 2064          2188  2
; 2065          2189  2
; 2066          2190  2      ! Check to see if the left margin is exceed the actual source text length.
; 2067          2191  2      !
; 2068          2192  2      IF .LEFT_MARGIN GTR .RIGHT_MARGIN
; 2069          2193  2      THEN
; 2070          2194  2          WINDOW_WIDTH = 0
; 2071          2195  2      ELSE
; 2072          2196  2          WINDOW_WIDTH = .RIGHT_MARGIN - .LEFT_MARGIN + 1;
; 2073          2197  2
; 2074          2198  2      TERM_WIDTH = .DBG$SRC_TERM_WIDTH - 8;
; 2075          2199  2
; 2076          2200  2
; 2077          2201  2      ! Formats the line number and statement number, and outputs the line.
; 2078          2202  2      ! Also folds long source lines when the output line exceed terminal width
; 2079          2203  2      ! in length.
; 2080          2204  2      !
; 2081          2205  2      FST_LINE_FLAG = TRUE;
; 2082          2206  2      TMPBUFPTR = TMPBUF + .LEFT_MARGIN - 1;
; 2083          2207  2      WHILE TRUE DO
; 2084          2208  3          BEGIN
; 2085          2209  3
; 2086          2210  3
; 2087          2211  3          ! Format the Line's line number for the first line to be outputed.  For
; 2088          2212  3          ! the line exceed terminal width in length, the second line starts with
; 2089          2213  3          ! '         -:'.
; 2090          2214  3          !
; 2091          2215  3          IF .FST_LINE_FLAG
; 2092          2216  3          THEN
; 2093          2217  4              BEGIN
; 2094          2218  4              IF .STMT_NUM EQL 0
; 2095          2219  4              THEN
; 2096          2220  5                  BEGIN
; 2097          2221  5                  SELECTONE .LINE_NUM OF
; 2098          2222  5                      SET
; 2099          2223  5                      [1 TO 9]:
; 2100          2224  5                          DBG$PRINT(UPLIT BYTE(%ASCIC '     '), 0);
; 2101          2225  5                      [10 TO 99]:
; 2102          2226  5                          DBG$PRINT(UPLIT BYTE(%ASCIC '    '), 0);
; 2103          2227  5                      [100 TO 999]:
; 2104          2228  5                          DBG$PRINT(UPLIT BYTE(%ASCIC '   '), 0);
; 2105          2229  5                      [1000 TO 9999]:
; 2106          2230  5                          DBG$PRINT(UPLIT BYTE(%ASCIC '  '), 0);
; 2107          2231  5                      [10000 TO 99999]:
; 2108          2232  5                          DBG$PRINT(UPLIT BYTE(%ASCIC ' '), 0);
; 2109          2233  5
; 2110          2234  5                      TES;
; 2111          2235  5
; 2112          2236  4                  END;
; 2113          2237  4
; 2114          2238  4              DBG$PRINT(UPLIT BYTE(%ASCIC '!UL: '), .LINE_NUM);
; 2115          2239  4              FST_LINE_FLAG = FALSE;
; 2116          2240  4              END
; 2117          2241  4
; 2118          2242  3          ELSE
```

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

                                                00FFC 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                                      ;
                                58              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$:      INCL    TABCNT                                  ; 2148
                      58        58              57  C3 00054          SUBL3   BGNPTR, TAB_POSITION, GAP               ; 2154
                                                  04  13 00058          BEQL    7$                                      ; 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$SRC_TERM_WIDTH, TERM_WIDTH      ; 2198
                                55              01  D0 000E1          MOVL    #1, FST_LINE_FLAG                       ; 2205
                                5A        03 AE41   9E 000E4          MOVAB   TMPBUF-T[LEFT_MARGIN], TMPBUFPTR        ; 2206
                                03              55  E8 000E9 13$:      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
```

```
;  2140     2263  1 ROUTINE DBG$SRC_READ_FILE(SFCBPTR, REC_NUM, BUFPTR, BUFLEN): NOVALUE =
;  2141     2264  1 !
;  2142     2265  1 !  FUNCTION
;  2143     2266  1 !        This routine reads a specified source record from a specified source
;  2144     2267  1 !        file and returns the address and length of the read record.  This is
;  2145     2268  1 !        thus the routine which actually reads source lines from source files.
;  2146     2269  1 !
;  2147     2270  1 !        This routine assumes that the source file has already been opened by
;  2148     2271  1 !        DBG$SRC_OPEN_FILE.  It reads the source file via RMS or the LIBRARIAN
;  2149     2272  1 !        until the desired record number is reached.  The image corresponding
;  2150     2273  1 !        to that record number is then returned to the caller.
;  2151     2274  1 !
;  2152     2275  1 !        This routine also builds the Record File Address (RFA) Table for the
;  2153     2276  1 !        current source file as it reads.  This table can then be used on later
;  2154     2277  1 !        calls to position the file pointer directly at the desired line or on
;  2155     2278  1 !        a line shortly before the desired line.  The RFA table is obviously
;  2156     2279  1 !        used whenever available to take us closer to the desired line.  That
;  2157     2280  1 !        table is also expanded whenever we have to read past the last RFA in
;  2158     2281  1 !        the table.  In other words, no source line is read just to build up the
;  2159     2282  1 !        RFA Table, but whenever a source line must be scanned anyway to find a
;  2160     2283  1 !        desired source record, its RFA is recovered for use in the RFA Table.
;  2161     2284  1 !
;  2162     2285  1 !  INPUTS
;  2163     2286  1 !        SFCBPTR - A pointer to the Source File Control Block for the file to be
;  2164     2287  1 !                  read.  It is assumed that this control block has already been
;  2165     2288  1 !                  set up by routine DBG$SRC_OPEN_FILE.
;  2166     2289  1 !
;  2167     2290  1 !        REC_NUM - The Record Number within the file of the source record to be
;  2168     2291  1 !                  read in.
;  2169     2292  1 !
;  2170     2293  1 !        BUFPTR  - A longword location to receive a pointer to the input buffer.
;  2171     2294  1 !
;  2172     2295  1 !        BUFLEN  - A longword location to receive the input line's length.
;  2173     2296  1 !
;  2174     2297  1 !  OUTPUTS
;  2175     2298  1 !        BUFPTR  - A pointer to the buffer which contains the just read input
;  2176     2299  1 !                  line is returned to BUFPTR.
;  2177     2300  1 !
;  2178     2301  1 !        BUFLEN  - The length in characters of the just read input line is
;  2179     2302  1 !                  returned to BUFLEN.
;  2180     2303  1 !
;  2181     2304  1
;  2182     2305  2     BEGIN
;  2183     2306  2
;  2184     2307  2     MAP
;  2185     2308  2         SFCBPTR: REF SFCB$BLOCK,        ! Pointer to Source File Control Block
;  2186     2309  2         BUFPTR: REF VECTOR[1],          ! Pointer to a longword to receive the
;  2187     2310  2                                         !       input line's buffer pointer
;  2188     2311  2         BUFLEN: REF VECTOR[1];          ! Pointer to a longword to receive the
;  2189     2312  2                                         !       input line's length
;  2190     2313  2
;  2191     2314  2     LOCAL
;  2192     2315  2         CLOSEST_TABLE_ENTRY,
;  2193     2316  2         INPTR: BLOCK[8,BYTE],           ! Length and string address for the user
;  2194     2317  2                                         !       -supplied buffer
;  2195     2318  2         OUTPTR: BLOCK[8,BYTE],          ! Pointer to a string descriptor for the
;  2196     2319  2                                         !       actual record returned from LBR
```

```
;  2197          2320  2          SRCNAM: REF $NAM_DECL,                  ! NAME Block for RMS file
;  2198          2321  2          SRCRAB: REF $RAB_DECL,                  ! RAB for RMS file
;  2199          2322  2          STARTING_REC,
;  2200          2323  2          STATUS,                                 ! RAB and FIND status
;  2201          2324  2          TXTRFA: VECTOR[2],                      ! Pointer to RFA
;  2202          2325  2          RFA_TABLE : REF RFATBL$BLOCKVECTOR(RFATBL_ENTRIES),
;  2203          2326  2                                                  ! The RFA table for the file
;  2204          2327  2          RFATBL_SPACING;                         ! Number of source records per RFA
;  2205          2328  2                                                  ! table entry.
;  2206          2329  2
;  2207          2330  2
;  2208          2331  2      ! Determine whether to read from an RMS file or a module in a source library.
;  2209          2332  2      !
;  2210          2333  2      SRCNAM = .SFCBPTR[SFCB$L_NAMPTR];
;  2211          2334  2      CASE .SFCBPTR[SFCB$B_KIND] FROM SFCB$K_RMSFILE TO SFCB$K_LBRFILE OF
;  2212          2335  2          SET
;  2213          2336  2
;  2214          2337  2
;  2215          2338  2          ! Read from an RMS source file.
;  2216          2339  2          !
;  2217          2340  2          [SFCB$K_RMSFILE]:
;  2218          2341  3              BEGIN
;  2219          2342  3
;  2220          2343  3              ! Set up the RAB pointer and RFA table.
;  2221          2344  3              !
;  2222          2345  3              SRCRAB = .SFCBPTR[SFCB$L_RABPTR];
;  2223          2346  3              RFA_TABLE = .SFCBPTR[SFCB$L_RFATBLPTR];
;  2224          2347  3              RFATBL_SPACING = .SFCBPTR[SFCB$W_RFASPACING];
;  2225          2348  3
;  2226          2349  3
;  2227          2350  3              ! Find the line in the RFA table which represents the
;  2228          2351  3              ! record which is closest to the one we are looking for.
;  2229          2352  3              !
;  2230          2353  3              CLOSEST_TABLE_ENTRY = MIN( .SFCBPTR[SFCB$W_RFACURLEN],
;  2231          2354  3                                         (.REC_NUM-1)/.RFATBL_SPACING);
;  2232          2355  3              STARTING_REC = .CLOSEST_TABLE_ENTRY*.RFATBL_SPACING;
;  2233          2356  3
;  2234          2357  3
;  2235          2358  3              ! If the Source File is newly opened, get the first records's file
;  2236          2359  3              ! address field and store it in RFA table.
;  2237          2360  3              !
;  2238          2361  3              IF .SFCBPTR[SFCB$L_CURRECNUM] EQL 1
;  2239          2362  3              THEN
;  2240          2363  4                  BEGIN
;  2241          2364  4                  STATUS = $FIND(RAB = .SRCRAB);
;  2242          2365  4                  IF NOT .STATUS THEN SIGNAL(.STATUS);
;  2243          2366  4                  RFA_TABLE[0,RFATBL$L_RFA0] = .SRCRAB[RAB$L_RFA0];
;  2244          2367  4                  RFA_TABLE[0,RFATBL$W_RFA4] = .SRCRAB[RAB$W_RFA4];
;  2245          2368  4                  CLOSEST_TABLE_ENTRY = 0;
;  2246          2369  3                  END;
;  2247          2370  3
;  2248          2371  3
;  2249          2372  3              ! Take the RFA table entry that is closest to the desired record,
;  2250          2373  3              ! and position the RAB so that we are pointing to the record
;  2251          2374  3              ! corresponding to this RFA.
;  2252          2375  3              ! (If we are already positioned at the desired record we
;  2253          2376  3              ! skip this)
```

```
;  2254          2377  3          !
;  2255          2378  3          IF .REC_NUM NEQ .SFCBPTR[SFCB$L_CURRECNUM]
;  2256          2379  3          THEN
;  2257          2380  4              BEGIN
;  2258          2381  4              SRCRAB[RAB$L_RFA0] = .RFA_TABLE[.CLOSEST_TABLE_ENTRY,
;  2259          2382  4                      RFATBL$L_RFA0];
;  2260          2383  4              SRCRAB[RAB$W_RFA4] = .RFA_TABLE[.CLOSEST_TABLE_ENTRY,
;  2261          2384  4                      RFATBL$W_RFA4];
;  2262          2385  5              SFCBPTR[SFCB$L_CURRECNUM] = 1 + (.CLOSEST_TABLE_ENTRY *
;  2263          2386  4                      .RFATBL_SPACING);
;  2264          2387  4              SRCRAB[RAB$B_RAC] = RAB$C_RFA;
;  2265          2388  4              STATUS = $FIND(RAB = .SRCRAB);
;  2266          2389  4              IF NOT .STATUS THEN SIGNAL(.STATUS);
;  2267          2390  3              END;
;  2268          2391  3
;  2269          2392  3
;  2270          2393  3          ! Sequentially read from the current record to the desired record
;  2271          2394  3          ! number.  Skip over records containing only <Form Feed>;
;  2272          2395  3          ! do not count them toward the record count.
;  2273          2396  3          !
;  2274          2397  3          SRCRAB[RAB$B_RAC] = RAB$C_SEQ;
;  2275          2398  3          INCR I FROM .SFCBPTR[SFCB$L_CURRECNUM]-1 TO .REC_NUM-1 DO
;  2276          2399  4              BEGIN
;  2277          2400  4              WHILE TRUE DO
;  2278          2401  5                  BEGIN
;  2279          2402  5                  STATUS = $GET(RAB = .SRCRAB);
;  2280          2403  5                  IF NOT .STATUS
;  2281          2404  5                  THEN
;  2282          2405  5                      SIGNAL(DBG$_UNAREASRC,2,.SRCNAM[NAM$B_RSL],.SRCNAM[NAM$L_RSA],.STATUS);
;  2283          2406  5
;  2284          2407  5
;  2285          2408  5                  ! If this flag is set, that means we do not want to skip
;  2286          2409  5                  ! the formfeed record.
;  2287          2410  5                  !
;  2288          2411  5                  IF .DBG$SRC_FORMFEED_FLAG THEN EXITLOOP;
;  2289          2412  6                  IF NOT ((.DBG$SRC_REC_BUF[0] EQL 12) AND
;  2290          2413  6                          (.SRCRAB[RAB$W_RSZ] EQL 1))
;  2291          2414  5                  THEN
;  2292          2415  5                      EXITLOOP;
;  2293          2416  5
;  2294          2417  4                  END;
;  2295          2418  4
;  2296          2419  4
;  2297          2420  4          ! Check for whether we need to update the RFA table.
;  2298          2421  4          ! We do this every "n" records, where "n" is RFATBL_SPACING.
;  2299          2422  4          ! As a minor optimization, we do not update the table
;  2300          2423  4          ! for the very first record we read - this table entry
;  2301          2424  4          ! is already filled in.
;  2302          2425  4          !
;  2303          2426  4          IF ((.I MOD .RFATBL_SPACING) EQL 0) AND
;  2304          2427  5             (.I NEQ .STARTING_REC)
;  2305          2428  4          THEN
;  2306          2429  5              BEGIN ! update table
;  2307          2430  5
;  2308          2431  5              ! First check whether the table is full.
;  2309          2432  5              !
;  2310          2433  5              IF (.I / .RFATBL_SPACING) GEQ RFATBL_ENTRIES
```

```
;  2311          2434  5                THEN
;  2312          2435  6                    BEGIN
;  2313          2436  6
;  2314          2437  6                    ! Compact the table.
;  2315          2438  6                    !
;  2316          2439  6                    RFATBL_SPACING = 2 * .RFATBL_SPACING;
;  2317          2440  6                    SFCBPTR[SFCB$W_RFASPACING] = .RFATBL_SPACING;
;  2318          2441  6                    INCR J FROM 0 TO RFATBL_ENTRIES-2 BY 2 DO
;  2319          2442  7                        BEGIN
;  2320          2443  7                        RFA_TABLE[.J/2,RFATBL$L_RFA0] =
;  2321          2444  7                            .RFA_TABLE[.J,RFATBL$L_RFA0];
;  2322          2445  7                        RFA_TABLE[.J/2,RFATBL$W_RFA4] =
;  2323          2446  7                            .RFA_TABLE[.J,RFATBL$W_RFA4];
;  2324          2447  6                        END;
;  2325          2448  5                    END;
;  2326          2449  5
;  2327          2450  5                ! Update the table length and fill in the new
;  2328          2451  5                ! table entry.
;  2329          2452  5                !
;  2330          2453  5                SFCBPTR[SFCB$W_RFACURLEN] = .I/.RFATBL_SPACING;
;  2331          2454  5                IF .I MOD .RFATBL_SPACING EQL 0
;  2332          2455  5                THEN
;  2333          2456  6                    BEGIN
;  2334          2457  6                    RFA_TABLE[.I/.RFATBL_SPACING,RFATBL$L_RFA0] =
;  2335          2458  6                        .SRCRAB[RAB$L_RFA0];
;  2336          2459  6                    RFA_TABLE[.I/.RFATBL_SPACING,RFATBL$W_RFA4] =
;  2337          2460  6                        .SRCRAB[RAB$W_RFA4];
;  2338          2461  5                    END;
;  2339          2462  5
;  2340          2463  4                END; ! update the table
;  2341          2464  4
;  2342          2465  3            END; ! incr loop
;  2343          2466  3
;  2344          2467  3            ! Update the current record number field in SFCB. Set up the buffer
;  2345          2468  3            ! pointer and buffer length.
;  2346          2469  3            !
;  2347          2470  3            SFCBPTR[SFCB$L_CURRECNUM] = .REC_NUM + 1;
;  2348          2471  3            BUFPTR[0] = .SRCRAB[RAB$L_RBF];
;  2349          2472  3            BUFLEN[0] = .SRCRAB[RAB$W_RSZ];
;  2350          2473  3
;  2351          2474  2            END; ! RMS file case alternative
;  2352          2475  2
;  2353          2476  2
;  2354          2477  2        ! Read from a module in a Source Library File.
;  2355          2478  2        !
;  2356          2479  2        [SFCB$K_LBRFILE]:
;  2357          2480  3            BEGIN
;  2358          2481  3
;  2359          2482  3
;  2360          2483  3            ! If the current record number is beyond the desired record number,
;  2361          2484  3            ! reset the current record to 1 and set the record file address to the
;  2362          2485  3            ! beginning of the module.
;  2363          2486  3            !
;  2364          2487  3            IF .SFCBPTR[SFCB$L_CURRECNUM] GTR .REC_NUM
;  2365          2488  3            THEN
;  2366          2489  4                BEGIN
;  2367          2490  4                TXTRFA[0] = .SFCBPTR[SFCB$L_CUR_RFA0];
```

```
;  2368          2491  4              TXTRFA[1] = .SFCBPTR[SFCB$W_CUR_RFA4];
;  2369          2492  4              SFCBPTR[SFCB$L_CURRECNUM] = 1;
;  2370          2493  4              STATUS = LBR$FIND(SFCBPTR[SFCB$L_LBRINDEX],TXTRFA);
;  2371          2494  4              IF NOT .STATUS THEN SIGNAL(.STATUS);
;  2372          2495  3              END;
;  2373          2496  3
;  2374          2497  3
;  2375          2498  3          ! Sequentially read from the current record to the desired record
;  2376          2499  3          ! number.
;  2377          2500  3          !
;  2378          2501  3          INPTR[DSC$W_LENGTH] = SRCBUFSIZE;
;  2379          2502  3          INPTR[DSC$A_POINTER] = DBG$SRC_REC_BUF[0];
;  2380          2503  3          OUTPTR[DSC$A_POINTER] = DBG$SRC_REC_BUF[0];
;  2381          2504  3          INCR I FROM .SFCBPTR[SFCB$L_CURRECNUM] TO .REC_NUM DO
;  2382          2505  4              BEGIN
;  2383          2506  4              WHILE TRUE DO
;  2384          2507  5                  BEGIN
;  2385          2508  5                  STATUS = LBR$GET_RECORD(SFCBPTR[SFCB$L_LBRINDEX],INPTR,OUTPTR);
;  2386          2509  5                  IF NOT .STATUS
;  2387          2510  5                  THEN
;  2388          2511  5                      SIGNAL(DBG$_UNAREASRC,2,.SRCNAM[NAM$B_RSL],.SRCNAM[NAM$L_RSA],.STATUS);
;  2389          2512  5
;  2390          2513  5                  IF .DBG$SRC_FORMFEED_FLAG THEN EXITLOOP;
;  2391          2514  6                  IF NOT ((.DBG$SRC_REC_BUF[0] EQL 12) AND
;  2392          2515  6                          (.OUTPTR[DSC$W_LENGTH] EQL 1))
;  2393          2516  5                  THEN
;  2394          2517  5                      EXITLOOP;
;  2395          2518  5
;  2396          2519  4                  END;
;  2397          2520  4
;  2398          2521  3              END;
;  2399          2522  3
;  2400          2523  3
;  2401          2524  3              ! Update the current record number field in SFCB.  Set up buffer
;  2402          2525  3              ! pointer and buffer length.
;  2403          2526  3              !
;  2404          2527  3              SFCBPTR[SFCB$L_CURRECNUM] = .REC_NUM + 1;
;  2405          2528  3              BUFPTR[0] = .OUTPTR[DSC$A_POINTER];
;  2406          2529  3              BUFLEN[0] = .OUTPTR[DSC$W_LENGTH];
;  2407          2530  3
;  2408          2531  2              END;
;  2409          2532  2
;  2410          2533  2          [INRANGE, OUTRANGE]:
;  2411          2534  2              $DBG_ERROR('DBGSOURCE\DBG$SRC_READ_FILE');
;  2412          2535  2
;  2413          2536  2          TES;
;  2414          2537  2
;  2415          2538  2      RETURN;
;  2416          2539  2
;  2417          2540  1      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 00  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                            ; 2506
  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
```

```
;  2419          2541  1 ROUTINE DBG$SRC_SEARCH(BUFPTR, BUFLEN): =
;  2420          2542  1 !
;  2421          2543  1 ! FUNCTION
;  2422          2544  1 !       This routine actually performs the search.  This routine locates the
;  2423          2545  1 !       desired search string in the input buffer.  Lowercase letters are
;  2424          2546  1 !       translated to uppercase letter before the search is performed.
;  2425          2547  1 !       If there is a match, this routine returns TRUE, otherwise FALSE is
;  2426          2548  1 !       returned.  There are two ways of searching for the desired string.
;  2427          2549  1 !       One is to search the string that has the match, ie. SEARCH for "ABC",
;  2428          2550  1 !       then "ABCXXX", or "XXXABC", or "XXABCXX" would be found.  The other
;  2429          2551  1 !       is to search the string that has the exact match, i.e. the characters
;  2430          2552  1 !       to the left and the right of the search string "ABC" can not be a valid
;  2431          2553  1 !       character in identifiers in the currently set language.
;  2432          2554  1 !
;  2433          2555  1 ! INPUT
;  2434          2556  1 !       BUFPTR  - A pointer to a buffer which contains the ASCII text of the
;  2435          2557  1 !                 source line to be printed.
;  2436          2558  1 !
;  2437          2559  1 !       BUFLEN  - The length in characters of the source line pointed to by
;  2438          2560  1 !                 BUFPTR.
;  2439          2561  1 !
;  2440          2562  1 ! OUTPUT
;  2441          2563  1 !       This routine returns TRUE of the search string was found in the current
;  2442          2564  1 !                 line and FALSE if it was not.
;  2443          2565  1 !
;  2444          2566  2     BEGIN
;  2445          2567  2
;  2446          2568  2     MAP
;  2447          2569  2         BUFPTR: REF VECTOR[,BYTE];      ! Pointer to the ASCII image to print
;  2448          2570  2
;  2449          2571  2     LOCAL
;  2450          2572  2         CHAR,                           ! Serve as an index character index
;  2451          2573  2                                         !      into the Translate Table
;  2452          2574  2         DES_STR: BLOCK[8,BYTE],         ! Address of destination string
;  2453          2575  2                                         !      descriptor
;  2454          2576  2         SRC_STR: BLOCK[8,BYTE],         ! Address of source string descriptor
;  2455          2577  2         STATUS,                         ! Return status from System routines
;  2456          2578  2         TMPBUF: VECTOR[SRCBUFSIZE,BYTE],! Tempory buffer which holds all the
;  2457          2579  2                                         !      letters in uppercase
;  2458          2580  2         TMPLEN,                         ! Remaining size of tempory buffer
;  2459          2581  2         TMPPTR;                         ! Pointer to the tempory buffer
;  2460          2582  2
;  2461          2583  2
;  2462          2584  2
;  2463          2585  2     ! Translate all lowercase letters to uppercase via STR$UPCASE.  Before
;  2464          2586  2     ! the call, build string descriptors for STR$UPCASE as passing parameters.
;  2465          2587  2     ! Array descriptor, character-coded text.
;  2466          2588  2     !
;  2467          2589  2     DES_STR[DSC$B_CLASS] = DSC$K_CLASS_S;
;  2468          2590  2     DES_STR[DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  2469          2591  2     DES_STR[DSC$W_LENGTH] = .BUFLEN;
;  2470          2592  2     DES_STR[DSC$A_POINTER] = TMPBUF;
;  2471          2593  2     SRC_STR[DSC$B_CLASS] = DSC$K_CLASS_S;
;  2472          2594  2     SRC_STR[DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  2473          2595  2     SRC_STR[DSC$W_LENGTH] = .BUFLEN;
;  2474          2596  2     SRC_STR[DSC$A_POINTER] = .BUFPTR;
;  2475          2597  2     STATUS = STR$UPCASE(DES_STR,SRC_STR);
```

```
; 2476          2598  2
; 2477          2599  2
; 2478          2600  2      ! Search for the desired string in the tempory buffer.  If the desired
; 2479          2601  2      ! string is not found, return FALSE.  If the desired string is found
; 2480          2602  2      ! and search qualifier is SEARCH, return TRUE.  If the desired
; 2481          2603  2      ! string is found and search qualifier is IDENTIFIER, then the left
; 2482          2604  2      ! and right characters of the desired string are checked.  If the
; 2483          2605  2      ! desired string is identifier, return TRUE.  Otherwise search goes
; 2484          2606  2      ! on till reaching to end of the buffer.
; 2485          2607  2      !
; 2486          2608  2      TMPLEN = .BUFLEN;
; 2487          2609  2      TMPPTR = TMPBUF;
; 2488          2610  2      STATUS = TRUE;
; 2489          2611  2      WHILE TRUE DO
; 2490          2612  3          BEGIN
; 2491          2613  3          TMPPTR = CH$FIND_SUB(.TMPLEN, .TMPPTR, .DBG$SRC_SEARCH_STRING[0],
; 2492          2614  3                   DBG$SRC_SEARCH_STRING[1]);
; 2493          2615  3
; 2494          2616  3
; 2495          2617  3          ! Check to see if the matched string is found.  If such string is not
; 2496          2618  3          ! found, return the status.
; 2497          2619  3          !
; 2498          2620  3          IF .TMPPTR EQL 0
; 2499          2621  3          THEN
; 2500          2622  4              BEGIN
; 2501          2623  4              STATUS = FALSE;
; 2502          2624  4              EXITLOOP;
; 2503          2625  3              END;
; 2504          2626  3
; 2505          2627  3
; 2506          2628  3          ! Check to see if the exact match option /IDENT is used.  If it is, goes on
; 2507          2629  3          ! with more checking.  Otherwise, we are all done.
; 2508          2630  3          !
; 2509          2631  3          IF (.DBG$SRC_SEARCH_FLAG EQL SEARCH_STRING_ALL) OR
; 2510          2632  4             (.DBG$SRC_SEARCH_FLAG EQL SEARCH_STRING_NEXT)
; 2511          2633  3          THEN
; 2512          2634  4              BEGIN
; 2513          2635  4              STATUS = TRUE;
; 2514          2636  4              EXITLOOP;
; 2515          2637  3              END;
; 2516          2638  3
; 2517          2639  3
; 2518          2640  3          ! Check to see if the characters on each side of the pattern string are
; 2519          2641  3          ! valid in identifiers for the current language.  This check is done by
; 2520          2642  3          ! looking up each character's attributes in the DBGPARSER Character
; 2521          2643  3          ! Table for the currently SET language.  First we check the character
; 2522          2644  3          ! to the left of the found string, and then the character on the right.
; 2523          2645  3          ! STATUS is set to FALSE if the string appears to be part of a larger
; 2524          2646  3          ! identifier.
; 2525          2647  3          !
; 2526          2648  3          IF .TMPPTR NEQ TMPBUF
; 2527          2649  3          THEN
; 2528          2650  4              BEGIN
; 2529          2651  4              CHAR = (.TMPPTR - 1) - TMPBUF;
; 2530          2652  4              STATUS = TRUE;
; 2531          2653  4              IF .DBG$GL_CHARTBL[.TMPBUF[.CHAR], CHRTBL$V_IDENT_START]  OR
; 2532          2654  4                 .DBG$GL_CHARTBL[.TMPBUF[.CHAR], CHRTBL$V_IDENT_MIDDLE] OR
```

```
;  2533          2655  4                  .DBG$GL_CHARTBL[.TMPBUF[.CHAR], CHRTBL$V_IDENT_END]
;  2534          2656  4              THEN
;  2535          2657  4                  STATUS = FALSE;
;  2536          2658  4
;  2537          2659  3              END;
;  2538          2660  3
;  2539          2661  3
;  2540          2662  3              ! Check the right hand side character.  First make sure there is a
;  2541          2663  3              ! right hand side character.  Again, we set STATUS to FALSE if the
;  2542          2664  3              ! string appears to be part of a larger identifier.
;  2543          2665  3              !
;  2544          2666  3              TMPPTR = .TMPPTR + .DBG$SRC_SEARCH_STRING[0];
;  2545          2667  3              IF .TMPPTR GEQ (TMPBUF + .BUFLEN) THEN EXITLOOP;
;  2546          2668  3              IF (.TMPPTR LEQ (TMPBUF + .BUFLEN)) AND .STATUS
;  2547          2669  3              THEN
;  2548          2670  4                  BEGIN
;  2549          2671  4                  CHAR = .TMPPTR - TMPBUF;
;  2550          2672  4                  STATUS = TRUE;
;  2551          2673  4                  IF .DBG$GL_CHARTBL[.TMPBUF[.CHAR], CHRTBL$V_IDENT_START]  OR
;  2552          2674  4                     .DBG$GL_CHARTBL[.TMPBUF[.CHAR], CHRTBL$V_IDENT_MIDDLE] OR
;  2553          2675  4                     .DBG$GL_CHARTBL[.TMPBUF[.CHAR], CHRTBL$V_IDENT_END]
;  2554          2676  4                  THEN
;  2555          2677  4                      STATUS = FALSE;
;  2556          2678  4
;  2557          2679  3                  END;
;  2558          2680  3
;  2559          2681  3              ! We have found the desired match, return good status to the caller.
;  2560          2682  3              !
;  2561          2683  3              IF .STATUS THEN EXITLOOP;
;  2562          2684  3              TMPLEN = TMPBUF + .BUFLEN - .TMPPTR;
;  2563          2685  3              IF .TMPLEN LEQ 0 THEN EXITLOOP;
;  2564          2686  3
;  2565          2687  3              END;                ! End of WHILE loop.
;  2566          2688  2
;  2567          2689  2          RETURN .STATUS;
;  2568          2690  2
;  2569          2691  1          END;


                                                OFFC 00000 DBG$SRC_SEARCH:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 2541
                    5B 00000000G   00   9E 00002          MOVAB   DBG$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 DO 000F7 12$:  MOVL    STATUS, R0                              ; 2690
                              04 000FA          RET                                             ; 2692

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

```
; 2572          2693  1 GLOBAL ROUTINE DBG$SRC_SEARCH_CMD(MODRSTPTR, LOW_LNUM, LOW_STMT,
; 2573          2694  1                                   HIGH_LNUM, HIGH_STMT,
; 2574          2695  1                                   SEARCH_BY_STRING_FLAG, SEARCH_NEXT_FLAG): NOVALUE =
; 2575          2696  1 !
; 2576          2697  1 ! FUNCTION
; 2577          2698  1 !       This routine accepts a Module RST Entry pointer, a line number range,
; 2578          2699  1 !       and flags set to indicate the options are used in SEARCH command.
; 2579          2700  1 !       The options for the SEARCH command are SEARCH by IDENTIFIER or SEARCH
; 2580          2701  1 !       by STRING; find NEXT matched string or ALL matched string in the range.
; 2581          2702  1 !       This routine then calls DBG$SRC_TYPE_LNUM_SOURCE to type out source
; 2582          2703  1 !       lines in that range from that module which contains matched search
; 2583          2704  1 !       string.
; 2584          2705  1 !
; 2585          2706  1 ! INPUTS
; 2586          2707  1 !       MODRSTPTR - A pointer to the Module RST Entry of the module in which
; 2587          2708  1 !                   the source lines are to be looked up.
; 2588          2709  1 !
; 2589          2710  1 !       LOW_LNUM - The line number of the first source line to be searched.
; 2590          2711  1 !                  This is assumed to be a line number in the MODRSTPTR module.
; 2591          2712  1 !
; 2592          2713  1 !       LOW_STMT - The statement number, if any, associated with the first
; 2593          2714  1 !                  source line to be searched.
; 2594          2715  1 !
; 2595          2716  1 !       HIGH_LNUM - The line number of the last source line to be searched.
; 2596          2717  1 !                   This too is assumed to be a line in the MODRSTPTR module.
; 2597          2718  1 !
; 2598          2719  1 !       HIGH_STMT - The statement number, if any, associated with the last
; 2599          2720  1 !                   source line to be searched.
; 2600          2721  1 !
; 2601          2722  1 !       SEARCH_BY_STRING_FLAG - Flag set to TRUE if /STRING is used, this
; 2602          2723  1 !                               means search for any string that matches the
; 2603          2724  1 !                               given search string.  Otherwise flag set to
; 2604          2725  1 !                               FALSE if /IDENTIFIER is used, this means
; 2605          2726  1 !                               search for the exact match for the given string.
; 2606          2727  1 !
; 2607          2728  1 !       SEARCH_NEXT_FLAG - Flag set to TRUE if /NEXT is used, this means
; 2608          2729  1 !                          search for the first match string in the given range.
; 2609          2730  1 !                          Otherwise flag set to FALSE if /ALL is used, this
; 2610          2731  1 !                          means search for all the match stings in the range.
; 2611          2732  1 !                                                                              ;
; 2612          2733  1 ! IMPLICIT INPUT
; 2613          2734  1 !       DBG$SRC_SEARCH_STRING - Buffer contains the desired search string.
; 2614          2735  1 !
; 2615          2736  1 ! OUTPUTS
; 2616          2737  1 !       NONE
; 2617          2738  1 !
; 2618          2739  1
; 2619          2740  2     BEGIN
; 2620          2741  2
; 2621          2742  2     LOCAL
; 2622          2743  2         SEARCH_FLAG;                    ! Flag set to non-zero value to indicate
; 2623          2744  2                                         !       the kind of SEARCH option is
; 2624          2745  2                                         !       active
; 2625          2746  2
; 2626          2747  2
; 2627          2748  2     ! If no search string string is given, signal the warning message.
; 2628          2749  2     !
```

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


                                        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
```

```
;  2689          2808  1 GLOBAL ROUTINE DBG$SRC_SET_SOURCE(MODRSTPTR, DIRLIST): NOVALUE =
;  2690          2809  1 !
;  2691          2810  1 ! FUNCTION
;  2692          2811  1 !       This routine handles the semantic processing of the SET SOURCE and SET
;  2693          2812  1 !       SOURCE/MODULE=xxx commands.  It accepts a Module RST Entry pointer if
;  2694          2813  1 !       /MODULE was specified on the command and it accepts a list of directory
;  2695          2814  1 !       names, each given by a Source Directory Search List Entry.  It then
;  2696          2815  1 !       builds a Source Directory Search List Header Block for the specified
;  2697          2816  1 !       module and enters this block on the linked list pointed to by variable
;  2698          2817  1 !       DBG$SRC_DIR_LIST.  It also throws away any directory search list super-
;  2699          2818  1 !       ceded by the new list.
;  2700          2819  1 !
;  2701          2820  1 !       This routine verfies each directory name on the list, using the $PARSE
;  2702          2821  1 !       system service. If any are invalid, an error is signalled.
;  2703          2822  1 !
;  2704          2823  1 !       The source directory search lists are eventually used by the routine
;  2705          2824  1 !       DBG$SRC_OPEN_FILE.
;  2706          2825  1 !
;  2707          2826  1 ! INPUTS
;  2708          2827  1 !       MODRSTPTR - A pointer to the Module RST Entry of the module specified
;  2709          2828  1 !                   on the SET SOURCE/MODULE command.  If the /MODULE qualifier
;  2710          2829  1 !                   was not present on the command, MODRSTPTR must be zero.
;  2711          2830  1 !
;  2712          2831  1 !       DIRLIST - A pointer to a source directory search list.  More precisely,
;  2713          2832  1 !                 DIRLIST points to the first of a list of singly linked Source
;  2714          2833  1 !                 Directory Search List (SDSL) Entries.  Each SDSL Entry con-
;  2715          2834  1 !                 tains a directory name (or even a full file name) in counted
;  2716          2835  1 !                 ASCII.  DIRLIST may not be zero--an empty directory list on
;  2717          2836  1 !                 the SET SOURCE command is a syntax error.
;  2718          2837  1 !
;  2719          2838  1 ! OUTPUTS
;  2720          2839  1 !       NONE
;  2721          2840  1 !
;  2722          2841  1
;  2723          2842  2     BEGIN
;  2724          2843  2
;  2725          2844  2     MAP
;  2726          2845  2         MODRSTPTR: REF RST$ENTRY,       ! Pointer to Module RST Entry or zero
;  2727          2846  2         DIRLIST: REF SDSL$ENTRY;        ! Pointer to directory list
;  2728          2847  2
;  2729          2848  2     LOCAL
;  2730          2849  2         BLANK_COUNT,                    ! Holds a count of the number of
;  2731          2850  2                                         !     leading blanks in the directory
;  2732          2851  2                                         !     name as the user entered it.
;  2733          2852  2         CHAR,                           ! Used to hold a single character
;  2734          2853  2                                         !     during blank-stripping.
;  2735          2854  2         CHAR_PTR,                       ! Pointer that is used to step through
;  2736          2855  2                                         !     a directory name
;  2737          2856  2                                         !     character by character, during
;  2738          2857  2                                         !     blank-stripping.
;  2739          2858  2         COUNT,                          ! Holds a count of the number of
;  2740          2859  2                                         !     non-blank characters in a
;  2741          2860  2                                         !     directory name that is being
;  2742          2861  2                                         !     verified.
;  2743          2862  2         DIRPTR: REF SDSL$HEADER,        ! Pointer that is used to walk through
;  2744          2863  2                                         !     the linked list of headers.
;  2745          2864  2         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;

```

```
;  2803          2922  3          ! Initialize blank_count and count
;  2804          2923  3          !
;  2805          2924  3          BLANK_COUNT = 0;
;  2806          2925  3          COUNT = 0;
;  2807          2926  3
;  2808          2927  3
;  2809          2928  3          ! Loop through the leading blanks to strip them off.
;  2810          2929  3          !
;  2811          2930  3          WHILE .BLANK_COUNT LSS .ENTRYPTR[SDSL$B_ENT_DIRLEN] DO
;  2812          2931  4              BEGIN
;  2813          2932  4              CHAR = CH$RCHAR_A (CHAR_PTR);
;  2814          2933  4              IF .CHAR NEQ DBG$K_BLANK THEN EXITLOOP;
;  2815          2934  4              BLANK_COUNT = .BLANK_COUNT + 1;
;  2816          2935  4              END;
;  2817          2936  3
;  2818          2937  3
;  2819          2938  3          ! If the name is all blanks, signal an error. This is caused by
;  2820          2939  3          ! a pair of commas with only blanks between them.
;  2821          2940  3          !
;  2822          2941  3          IF .BLANK_COUNT EQL .ENTRYPTR[SDSL$B_ENT_DIRLEN]
;  2823          2942  3          THEN
;  2824          2943  3              SIGNAL(DBG$_INVDIRNAM, 1, UPLIT BYTE (%ASCIC ' '));
;  2825          2944  3
;  2826          2945  3
;  2827          2946  3          ! Now loop through the non-blank characters. Move each character
;  2828          2947  3          ! to the STRING_BUF_PTR buffer temporarily and translate to
;  2829          2948  3          ! upper case.
;  2830          2949  3          !
;  2831          2950  3          WHILE .COUNT + .BLANK_COUNT LSS .ENTRYPTR[SDSL$B_ENT_DIRLEN] DO
;  2832          2951  4              BEGIN
;  2833          2952  4              COUNT = .COUNT + 1;
;  2834          2953  4              IF .CHAR LEQ 'z' AND .CHAR GEQ 'a'
;  2835          2954  4              THEN
;  2836          2955  4                  CHAR = .CHAR - ('a' - 'A');
;  2837          2956  4
;  2838          2957  4              CH$WCHAR_A (.CHAR, STRING_BUF_PTR);
;  2839          2958  4              CHAR = CH$RCHAR_A (CHAR_PTR);
;  2840          2959  4              END;
;  2841          2960  3
;  2842          2961  3
;  2843          2962  3          ! Copy the blank_stripped and translated directory name back
;  2844          2963  3          ! into the original entry.
;  2845          2964  3          !
;  2846          2965  3          CH$MOVE (.COUNT, STRING_BUF, ENTRYPTR[SDSL$A_ENT_DIRNAME]);
;  2847          2966  3          ENTRYPTR[SDSL$B_ENT_DIRLEN] = .COUNT;
;  2848          2967  3
;  2849          2968  3
;  2850          2969  3          ! Now check the directory name for syntactic validity. This is done
;  2851          2970  3          ! by $PARSE so that RMS does the checking.
;  2852          2971  3          !
;  2853          2972  3          TSTFAB[FAB$B_DNS] = .COUNT;
;  2854          2973  3          STATUS = $PARSE (FAB = TSTFAB);
;  2855          2974  3          IF NOT .STATUS
;  2856          2975  3          THEN
;  2857          2976  3              SIGNAL(DBG$_INVDIRNAM, 1, ENTRYPTR[SDSL$B_ENT_DIRLEN], .STATUS);
;  2858          2977  3
;  2859          2978  3          ENTRYPTR = .ENTRYPTR[SDSL$L_ENT_FLINK];
```

```
;  2860          2979  2          END;
;  2861          2980  2
;  2862          2981  2
;  2863          2982  2      ! If we get this far we have a valid directory list.  Attempt to find a
;  2864          2983  2      ! directory list for the same module in the current Source Directory Search
;  2865          2984  2      ! List.  If we find one, we remove it from the list.
;  2866          2985  2      !
;  2867          2986  2      DIRPTR = .DBG$SRC_DIR_LIST;
;  2868          2987  2      WHILE .DIRPTR NEQ 0 DO
;  2869          2988  3          BEGIN
;  2870          2989  3          IF .DIRPTR[SDSL$L_MODPTR] EQL .MODRSTPTR
;  2871          2990  3          THEN
;  2872          2991  4              BEGIN
;  2873          2992  4              DBG$SRC_CANCEL_SOURCE (.DIRPTR[SDSL$L_MODPTR]);
;  2874          2993  4              EXITLOOP;
;  2875          2994  3              END;
;  2876          2995  3
;  2877          2996  3          DIRPTR = .DIRPTR[SDSL$L_FLINK];
;  2878          2997  2          END;
;  2879          2998  2
;  2880          2999  2
;  2881          3000  2      ! Allocate and build a new Source Directory List Header Block.
;  2882          3001  2      ! Insert it at the end of the Source Directory Search List.
;  2883          3002  2      !
;  2884          3003  2      NEW_DIRPTR = DBG$GET_MEMORY (SDSL$K_HDR_SIZE);
;  2885          3004  2      NEW_DIRPTR[SDSL$L_LIST_PTR] = .DIRLIST;
;  2886          3005  2      NEW_DIRPTR[SDSL$L_MODPTR] = .MODRSTPTR;
;  2887          3006  2
;  2888          3007  2
;  2889          3008  2      ! Walk to the end of the Source Directory Search List.
;  2890          3009  2      !
;  2891          3010  2      DIRPTR = DBG$SRC_DIR_LIST;
;  2892          3011  2      WHILE .DIRPTR[SDSL$L_FLINK] NEQ 0 DO
;  2893          3012  2          DIRPTR = .DIRPTR[SDSL$L_FLINK];
;  2894          3013  2
;  2895          3014  2      DIRPTR[SDSL$L_FLINK] = .NEW_DIRPTR;
;  2896          3015  2      RETURN;
;  2897          3016  2
;  2898          3017  1      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  FB  00096              CALLS   #3, LIB$SIGNAL
                      50                     56  C1  0009D 5$:          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  D1  000AB              CMPL    CHAR, #122                               ; 2953
                                             0C  14  000B2              BGTR    6$
                           00000061          8F  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
                5          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
```

```
;  2900          3018  1 GLOBAL ROUTINE DBG$SRC_SHOW_SOURCE: NOVALUE =
;  2901          3019  1 !
;  2902          3020  1 ! FUNCTION
;  2903          3021  1 !       This routine handles the semantics of the SHOW SOURCE command.  It thus
;  2904          3022  1 !       formats and prints the contents of the directory search lists pointed to
;  2905          3023  1 !       by DBG$SRC_DIR_LIST.  If one or more SET SOURCE/MODULE commands are in
;  2906          3024  1 !       effect, the output has this general format:
;  2907          3025  1 !
;  2908          3026  1 !                       source directory search list for MODNAME1:
;  2909          3027  1 !                               [FIRST.DIR]
;  2910          3028  1 !                               LOGNAME$
;  2911          3029  1 !                       source directory search list for MODNAME2:
;  2912          3030  1 !                               [DIR1]
;  2913          3031  1 !                               [DIR2.SUBDIR]
;  2914          3032  1 !                               DB1:[DIR3.DIR31.DIR32]
;  2915          3033  1 !                       source directory search list for all other modules:
;  2916          3034  1 !                               [.SUBDIR]
;  2917          3035  1 !
;  2918          3036  1 !       If a SET SOURCE command is in effect but no SET SOURCE/MODULE command is
;  2919          3037  1 !       in effect, this format is used:
;  2920          3038  1 !
;  2921          3039  1 !                       source directory search list for all modules:
;  2922          3040  1 !                               [DIRNAME1]
;  2923          3041  1 !                               [DIRNAME2]
;  2924          3042  1 !
;  2925          3043  1 !       And if no SET SOURCE command is in effect at all, this is output:
;  2926          3044  1 !
;  2927          3045  1 !                       no source directory search list in effect
;  2928          3046  1 !
;  2929          3047  1 !       All directory names are output exactly as the user originally entered
;  2930          3048  1 !       them.  There is no attempt to translate logical names or determine
;  2931          3049  1 !       defaults until the directory name is actually used to open a file.
;  2932          3050  1 !       Hence no such translations are done for the SHOW SOURCE display.
;  2933          3051  1 !
;  2934          3052  1 ! INPUTS
;  2935          3053  1 !       NONE
;  2936          3054  1 !
;  2937          3055  1 ! OUTPUTS
;  2938          3056  1 !       NONE
;  2939          3057  1 !
;  2940          3058  1
;  2941          3059  2     BEGIN
;  2942          3060  2
;  2943          3061  2     LOCAL
;  2944          3062  2         DEF_DIRPTR : REF SDSL$HEADER,    ! Points to the default
;  2945          3063  2                                          !       Source Directory Search List,
;  2946          3064  2                                          !       if one exists.
;  2947          3065  2         DIRPTR : REF SDSL$HEADER,        ! A pointer used to walk through the
;  2948          3066  2                                          !       linked list of Source Directory
;  2949          3067  2                                          !       Search List header blocks.
;  2950          3068  2         MODNAMEPTR;                      ! Points to a module name.
;  2951          3069  2
;  2952          3070  2     ! If there is no Source Directory Search List in effect, print a message
;  2953          3071  2     ! to that effect and return.
;  2954          3072  2     !
;  2955          3073  2     IF .DBG$SRC_DIR_LIST EQL 0
;  2956          3074  2     THEN
```

```
;  2957     3075  3          BEGIN
;  2958     3076  3          DBG$PRINT(UPLIT BYTE(%ASCIC 'no directory search list in effect'));
;  2959     3077  3          DBG$NEWLINE();
;  2960     3078  3          RETURN;
;  2961     3079  2          END;
;  2962     3080  2
;  2963     3081  2
;  2964     3082  2      ! If SET SOURCE is in effect but no SET SOURCE/MODULE= is in effect,
;  2965     3083  2      ! print the "for all modules" header message and the corresponding
;  2966     3084  2      ! list of directory names. Then return.
;  2967     3085  2      !
;  2968     3086  2      DIRPTR = .DBG$SRC_DIR_LIST;
;  2969     3087  2      IF   .DIRPTR[SDSL$L_MODPTR] EQL 0
;  2970     3088  2           AND
;  2971     3089  2           .DIRPTR[SDSL$L_FLINK] EQL 0
;  2972     3090  2      THEN
;  2973     3091  3          BEGIN
;  2974     3092  3          DBG$PRINT (UPLIT BYTE(
;  2975     3093  3                  %ASCIC 'source directory search list for all modules:'));
;  2976     3094  3          DBG$NEWLINE();
;  2977     3095  3          OUTPUT_DIR_LIST(.DIRPTR[SDSL$L_LIST_PTR]);
;  2978     3096  3          RETURN;
;  2979     3097  2          END;
;  2980     3098  2
;  2981     3099  2
;  2982     3100  2      ! If we reach this point, there is at least one directory list for a
;  2983     3101  2      ! specific module. Loop through all the Source Directory Search List
;  2984     3102  2      ! Header Blocks and print the directory list for each corresponding
;  2985     3103  2      ! module.
;  2986     3104  2      !
;  2987     3105  2      DEF_DIRPTR = 0;
;  2988     3106  2      WHILE .DIRPTR NEQ 0 DO
;  2989     3107  3          BEGIN
;  2990     3108  3
;  2991     3109  3
;  2992     3110  3          ! Check for default directory list (SET SOURCE with no
;  2993     3111  3          ! module specified). Store this away in DEF_DIRPTR since
;  2994     3112  3          ! it will be printed last.
;  2995     3113  3          !
;  2996     3114  3          IF .DIRPTR[SDSL$L_MODPTR] EQL 0
;  2997     3115  3          THEN
;  2998     3116  3              DEF_DIRPTR = .DIRPTR
;  2999     3117  3
;  3000     3118  3
;  3001     3119  3          ! Else this SDSL Header Block is for a specific module. Print a
;  3002     3120  3          ! header message with the module name and print the corresponding
;  3003     3121  3          ! directory names.
;  3004     3122  3          !
;  3005     3123  3          ELSE
;  3006     3124  4              BEGIN
;  3007     3125  4              DBG$STA_SYMNAME (.DIRPTR[SDSL$L_MODPTR], MODNAMEPTR);
;  3008     3126  4              DBG$PRINT (UPLIT BYTE(
;  3009     3127  4                          %ASCIC 'source directory search list for !AC:'),
;  3010     3128  4                          .MODNAMEPTR);
;  3011     3129  4              DBG$NEWLINE();
;  3012     3130  4              OUTPUT_DIR_LIST (.DIRPTR[SDSL$L_LIST_PTR]);
;  3013     3131  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
```

```
;  3037          3154  1 ROUTINE DBG$SRC_TYPE_LINE(LINE_NUM, STMT_NUM, SRC_FILE, REC_NUM, STEP_FLAG): =
;  3038          3155  1 !
;  3039          3156  1 ! FUNCTION
;  3040          3157  1 !       This routine accepts the line number and statement number of a source
;  3041          3158  1 !       line, a pointer to the Declare Source File DST command for the corre-
;  3042          3159  1 !       sponding source file, and the record number of the line within that
;  3043          3160  1 !       file.  It then reads in the text of the source line and causes it to be
;  3044          3161  1 !       typed out to the user.  It is called by DBG$SRC_TYPE_LNUM_SOURCE each
;  3045          3162  1 !       time that routine finds a desired line number in the current module's
;  3046          3163  1 !       Source Line Correlation DST Record.
;  3047          3164  1 !
;  3048          3165  1 !       This routine calls DBG$SRC_OPEN_FILE with the pointer to the Declare
;  3049          3166  1 !       Source File DST command (where the full file declaration is found).
;  3050          3167  1 !       DBG$SRC_OPEN_FILE opens the file (unless it is already opened), and
;  3051          3168  1 !       returns a pointer to the file's Source File Control Block (SFCB).  This
;  3052          3169  1 !       pointer is then passed to DBG$SRC_READ_FILE to read in the actual source
;  3053          3170  1 !       image.  If SEARCH command is active then calls SRC$SRC_SEARCH to locate
;  3054          3171  1 !       the desired string.  If matched string is not found, exit from the
;  3055          3172  1 !       routine and return status back to caller.  Finally, DBG$SRC_OUTPUT_LINE is
;  3056          3173  1 !       is called to print the source line.
;  3057          3174  1 !
;  3058          3175  1 ! INPUTS
;  3059          3176  1 !       LINE_NUM - The line number of the line to be read and printed.  If the
;  3060          3177  1 !                  line is unnumbered (has no line number as is true for comments
;  3061          3178  1 !                  in some languages), LINE_NUM should be negative.
;  3062          3179  1 !
;  3063          3180  1 !       STMT_NUM - The statement number of the line to be read and printed.  If
;  3064          3181  1 !                  a statement number is not applicable (which is the case in
;  3065          3182  1 !                  most languages), STMT_NUM should be zero.
;  3066          3183  1 !
;  3067          3184  1 !       SRC_FILE - A pointer to the Declare Source File command for the desired
;  3068          3185  1 !                  source file in the Source Line Correlation DST Record.  This
;  3069          3186  1 !                  pointer defines which file the desired source line should be
;  3070          3187  1 !                  read from.
;  3071          3188  1 !
;  3072          3189  1 !       REC_NUM - The record number in the SRC_FILE source file of the source
;  3073          3190  1 !                 line to be read and printed.  This record number also comes
;  3074          3191  1 !                 from the Source Line Correlation DST Record.
;  3075          3192  1 !
;  3076          3193  1 !       STEP_FLAG - Flag set to TRUE to indicate this is called from STEP.
;  3077          3194  1 !
;  3078          3195  1 ! OUTPUTS
;  3079          3196  1 !       NONE
;  3080          3197  1 !
;  3081          3198  1
;  3082          3199  2     BEGIN
;  3083          3200  2
;  3084          3201  2     MAP
;  3085          3202  2         SRC_FILE: REF DST$SRC_COMMAND;  ! Pointer to Define Source File command
;  3086          3203  2                                         !       in Source Correlation DST Record
;  3087          3204  2
;  3088          3205  2     LOCAL
;  3089          3206  2         BUFLEN,                         ! The length in characters of the input
;  3090          3207  2                                         !       line
;  3091          3208  2         BUFPTR: REF VECTOR[,BYTE],      ! Pointer to the input buffer
;  3092          3209  2         SEARCH_STATUS,                  ! Flag set to TRUE if the search string
;  3093          3210  2                                         !       is found, for other conditions,
```

```
;  3094     3211  2                                    !       flag set to FALSE
;  3095     3212  2          SRC_FILE_CNTRL_BLK:        ! Pointer to the Source File
;  3096     3213  2                      REF SFCB$BLOCK; !       Control Block
;  3097     3214  2
;  3098     3215  2
;  3099     3216  2
;  3100     3217  2      ! Signal an error if SRC_FILE is not declared or set in Source File
;  3101     3218  2      ! Correlation DST Record.
;  3102     3219  2      !
;  3103     3220  2      IF .SRC_FILE EQL 0 THEN SIGNAL(DBG$_INVDSTREC);
;  3104     3221  2
;  3105     3222  2
;  3106     3223  2      ! Abort the source output if the user interrupted it with Control-Y DEBUG.
;  3107     3224  2      !
;  3108     3225  2      $ABORT_ON_CONTROL_Y;
;  3109     3226  2
;  3110     3227  2
;  3111     3228  2      ! Initialize SEARCH_STATUS to FALSE for all non-search commands, and default
;  3112     3229  2      ! to no matches for SEARCH command.
;  3113     3230  2      !
;  3114     3231  2      SEARCH_STATUS = FALSE;
;  3115     3232  2
;  3116     3233  2
;  3117     3234  2      ! Calls OPEN LINE routine to open the Source File pointed by Declare Source
;  3118     3235  2      ! File DST Command in the given directory.
;  3119     3236  2      !
;  3120     3237  2      DBG$SRC_OPEN_FILE(.SRC_FILE,SRC_FILE_CNTRL_BLK,.STEP_FLAG);
;  3121     3238  2
;  3122     3239  2
;  3123     3240  2      ! Calls READ SOURCE FILE routine to read in Source File Records.
;  3124     3241  2      !
;  3125     3242  2      DBG$SRC_READ_FILE(.SRC_FILE_CNTRL_BLK,.REC_NUM,BUFPTR,BUFLEN);
;  3126     3243  2
;  3127     3244  2
;  3128     3245  2      ! Call SEARCH routine to search the desried string if SEARCH command is in
;  3129     3246  2      ! action.  If we did not find the matched string, return the status to
;  3130     3247  2      ! TYPE LNUM SOURCE, otherwise call OUTPUT LINE routine to output the line.
;  3131     3248  2      !
;  3132     3249  2      IF .DBG$SRC_SEARCH_FLAG NEQ 0
;  3133     3250  2      THEN
;  3134     3251  3          BEGIN
;  3135     3252  3          SEARCH_STATUS = DBG$SRC_SEARCH(.BUFPTR,.BUFLEN);
;  3136     3253  3          IF NOT .SEARCH_STATUS THEN RETURN .SEARCH_STATUS;
;  3137     3254  2          END;
;  3138     3255  2
;  3139     3256  2
;  3140     3257  2      ! Call the OUTPUT_LINE routine to print the source line.  However, if a
;  3141     3258  2      ! screen display for source lines is active, we call DBG$SCR_SOURCE_LINE
;  3142     3259  2      ! instead as the output routine.
;  3143     3260  2      !
;  3144     3261  2      IF .SOURCE_TO_SCREEN_FLAG
;  3145     3262  2      THEN
;  3146     3263  2          DBG$SCR_SOURCE_LINE(.LINE_NUM,
;  3147     3264  2                  .SRC_FILE[DST$W_SRC_DF_FILEID], .REC_NUM, .BUFLEN, .BUFPTR)
;  3148     3265  2
;  3149     3266  2      ELSE
;  3150     3267  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
```

```
;3160     3276  1 GLOBAL ROUTINE DBG$SRC_TYPE_LNUM_SOURCE(MODRSTPTR, LOWLNUM, LOWSTMT,
;3161     3277  1                                         HIGHLNUM, HIGHSTMT,
;3162     3278  1                                         SEARCH_FLAG, STEP_FLAG): NOVALUE =
;3163     3279  1 !
;3164     3280  1 ! FUNCTION
;3165     3281  1 !       This routine accepts a Module RST Entry pointer and a line number range
;3166     3282  1 !       and types out all source lines, or, ALL matched strings or NEXT matched
;3167     3283  1 !       string for SEARCH command in that range from that module.  it is
;3168     3284  1 !       called to handle source line display in response to the TYPE command,
;3169     3285  1 !       the EXAMINE/SOURCE command, the SEARCH command, breakpoints, watchpoints,
;3170     3286  1 !       and stepping by line.  In fact, all source line displays go through
;3171     3287  1 !       this routine.
;3172     3288  1 !
;3173     3289  1 !       More specifically, this routine scans through all the Source Line Corre-
;3174     3290  1 !       lation DST Records for the specified module to find the desired line
;3175     3291  1 !       numbers.  It executes all the commands in these DST records as it goes
;3176     3292  1 !       along, and is thus able to determine the appropriate File ID and record
;3177     3293  1 !       number within the file for each desired line number.  For each such line
;3178     3294  1 !       it then calls DBG$SRC_TYPE_LINE to actually retrieve, or search, and/or
;3179     3295  1 !       display the text of the line.
;3180     3296  1 !
;3181     3297  1 !       It handles Declare Source File commands in the DST records by building a
;3182     3298  1 !       File ID Table for the current module.  This table consists of a singly
;3183     3299  1 !       linked list where each entry corresponds to one source file contributing
;3184     3300  1 !       to the current module.  The file's File ID and a pointer to its Declare
;3185     3301  1 !       Source File command block are stored in each entry.  SRC_FILEID_TBL
;3186     3302  1 !       points to this linked list.  DBG$SRC_MODRSTPTR points to the current
;3187     3303  1 !       module's Module RST Entry, this variable is set by this routine and
;3188     3304  1 !       is later used by DBG$SRC_TYPE_LINE.
;3189     3305  1 !
;3190     3306  1 ! INPUTS
;3191     3307  1 !       MODRSTPTR - A pointer to the Module RST Entry of the module in which
;3192     3308  1 !                   the source lines are to be looked up.
;3193     3309  1 !
;3194     3310  1 !       LOWLNUM - The line number of the first source line to be typed out.
;3195     3311  1 !                 This is assumed to be a line number in the MODRSTPTR module.
;3196     3312  1 !
;3197     3313  1 !       LOWSTMT - The statement number, if any, associated with the first
;3198     3314  1 !                 source line to be typed out.  If no statement number should
;3199     3315  1 !                 be associated with the LOWLNUM line, LOWSTMT must be zero.
;3200     3316  1 !
;3201     3317  1 !       HIGHLNUM - The line number of the last source line to be type out.
;3202     3318  1 !                  This too is assumed to be a line in the MODRSTPTR module.
;3203     3319  1 !                  HIGHLNUM must be greater than or equal to LOWLNUM.
;3204     3320  1 !
;3205     3321  1 !       HIGHSTMT - The statement number, if any, associated with the last
;3206     3322  1 !                  source line to be typed out.  If no statement number should
;3207     3323  1 !                  be associated with the HIGHLNUM line, HIGHSTMT must be zero.
;3208     3324  1 !                  If HIGHLNUM equals LOWLNUM, HIGHSTMT must be greater than
;3209     3325  1 !                  or equal to LOWSTMT.
;3210     3326  1 !
;3211     3327  1 !       SEARCH_FLAG - Flag set to non-zero value to indicate the SEARCH command
;3212     3328  1 !                     is active, set to zero to indicate other commands are active.
;3213     3329  1 !
;3214     3330  1 !       STEP_FLAG - Flag set to TRUE to indicate the call is from STEP command.
;3215     3331  1 !                 If the source file is not available we only give the message
;3216     3332  1 !                 once, and suppress it the rest of the time.
```

```
;  3217     3333  1 !
;  3218     3334  1 ! OUTPUTS
;  3219     3335  1 !       NONE
;  3220     3336  1 !
;  3221     3337  1
;  3222     3338  2     BEGIN
;  3223     3339  2
;  3224     3340  2     MAP
;  3225     3341  2         MODRSTPTR: REF RST$ENTRY;        ! Pointer to Module RST Entry of module
;  3226     3342  2                                          !      containing desired source lines
;  3227     3343  2
;  3228     3344  2     LOCAL
;  3229     3345  2         ALL_DONE_FLAG,                   ! Flag set to TRUE if source operation
;  3230     3346  2                                          !      is all done in screen display
;  3231     3347  2         BUFFER: VECTOR[140,BYTE],        ! Output buffer for signaling
;  3232     3348  2         BUF_DESC: VECTOR[2,LONG],        ! Output buffer string descriptor
;  3233     3349  2         DSTPTR: REF DST$SRC_COMMAND,     ! Pointer to the Source File Correlation
;  3234     3350  2                                          !      Command
;  3235     3351  2         DSTREC_PTR: REF DST$RECORD,      ! Pointer to the Source File Correlation
;  3236     3352  2                                          !      DST record
;  3237     3353  2         HIGH_LINE_NUM,                   ! Place holder for Ending Line Number
;  3238     3354  2         HIGH_LNUM,                       ! Highest line number of range
;  3239     3355  2         HIGH_STMT,                       ! Statement number of of highest line
;  3240     3356  2         LENGTH,                          ! Length of current Source File Correlation
;  3241     3357  2                                          !      DST Record
;  3242     3358  2         LINE_NUM,                        ! The current listing line number
;  3243     3359  2         LOW_LNUM,                        ! Lowest line number of range
;  3244     3360  2         LOW_STMT,                        ! Statement number of lowest line
;  3245     3361  2         MODNAMEPTR: REF VECTOR[,BYTE],   ! Pointer to counted ASCII name of current
;  3246     3362  2                                          !      module
;  3247     3363  2         MODSRCTBL: REF VECTOR[,LONG],    ! Pointer to a list of Source File
;  3248     3364  2                                          !      Correlation DST Record pointers
;  3249     3365  2                                          !      for the current module
;  3250     3366  2         SEARCH_FND,                      ! Used for SEARCH command only.  Flag
;  3251     3367  2                                          !      set to TRUE if the search string
;  3252     3368  2                                          !      is found
;  3253     3369  2         SFIT_PTR: REF SFIT$ENTRY,        ! Pointer to the Source File ID Table
;  3254     3370  2         SRC_FILEID_TBL,                  ! Pointer to a linked list of File ID
;  3255     3371  2                                          !      Table entries for current module
;  3256     3372  2         SRC_DSTPTR,                      ! Pointer to Declare Source File Command
;  3257     3373  2                                          !      in DST Record
;  3258     3374  2         SRC_REC,                         ! The record number in the current source
;  3259     3375  2                                          !      file of the current source line
;  3260     3376  2         STATEMENT_MODE,                  ! Flag set to TRUE if we are currently
;  3261     3377  2                                          !      in Statement Mode
;  3262     3378  2         STMT_NUM,                        ! The current statement number
;  3263     3379  2         TYPE_FLAG;                       ! Flag set to TRUE if TYPE LINE routine
;  3264     3380  2                                          !      has been called at least once
;  3265     3381  2
;  3266     3382  2
;  3267     3383  2
;  3268     3384  2     ! Make sure Module RST pointer is not null.
;  3269     3385  2     !
;  3270     3386  2     IF .MODRSTPTR EQL 0
;  3271     3387  2     THEN
;  3272     3388  2         $DBG_ERROR('DBGSOURCE\DBG$SRC_TYPE_LNUM_SOURCE 10');
;  3273     3389  2
```

```
;  3274     3390  2
;  3275     3391  2      ! Start by doing some consistency checks to ensure that the line number
;  3276     3392  2      ! of the last source line is greater than or equal to the line number
;  3277     3393  2      ! of the first source line.  If no statement number should be associated
;  3278     3394  2      ! with line number, statement number must be zero.  If HIGHLNUM equals
;  3279     3395  2      ! LOWLNUM, HIGHSTMT must be greater than or equal to LOWSTMT.  If
;  3280     3396  2      ! either is wrong, signal an error and give up.
;  3281     3397  2      !
;  3282     3398  2      IF (.LOWLNUM GTR .HIGHLNUM) THEN SIGNAL(DBG$_INVSRCLIN);
;  3283     3399  3      IF (.LOWLNUM EQL .HIGHLNUM) AND (.LOWSTMT GTR .HIGHSTMT)
;  3284     3400  2      THEN
;  3285     3401  2          SIGNAL(DBG$_INVSRCLIN);
;  3286     3402  2
;  3287     3403  2
;  3288     3404  2      ! Determine whether the source output from this call should be redirected
;  3289     3405  2      ! to a screen display.
;  3290     3406  2      !
;  3291     3407  2      SOURCE_TO_SCREEN_FLAG = FALSE;
;  3292     3408  2      IF (.DBG$GL_SCREEN_SOURCE NEQ 0) AND
;  3293     3409  2         (.SEARCH_FLAG EQL 0)          AND
;  3294     3410  3         (NOT .STEP_FLAG)
;  3295     3411  2      THEN
;  3296     3412  2          SOURCE_TO_SCREEN_FLAG = TRUE;
;  3297     3413  2
;  3298     3414  2
;  3299     3415  2      ! Copy the line number range into local variables.  Then, if this is part
;  3300     3416  2      ! of a screen output operation, initialize the screen output and adjust
;  3301     3417  2      ! the line number range according to the size of the screen display to
;  3302     3418  2      ! receive the source output.
;  3303     3419  2      !
;  3304     3420  2      LOW_LNUM = .LOWLNUM;
;  3305     3421  2      LOW_STMT = .LOWSTMT;
;  3306     3422  2      HIGH_LNUM = .HIGHLNUM;
;  3307     3423  2      HIGH_STMT = .HIGHSTMT;
;  3308     3424  2      IF .SOURCE_TO_SCREEN_FLAG
;  3309     3425  2      THEN
;  3310     3426  3          BEGIN
;  3311     3427  3          DBG$SCR_SOURCE_BEGIN(.LOWLNUM, .HIGHLNUM,
;  3312     3428  3                               LOW_LNUM, HIGH_LNUM, .MODRSTPTR, ALL_DONE_FLAG);
;  3313     3429  3          IF .ALL_DONE_FLAG THEN RETURN;
;  3314     3430  2          END;
;  3315     3431  2
;  3316     3432  2
;  3317     3433  2      ! Initialization.  Assume that Line Number is set to the beginning of the
;  3318     3434  2      ! listing, Source File Record Number is set to the beginning of the
;  3319     3435  2      ! source file, no Source File ID is found, Statement Mode is not set,
;  3320     3436  2      ! no line is yet typed out, and statement number is set to 0.
;  3321     3437  2      !
;  3322     3438  2      IF .SEARCH_FLAG EQL 0 THEN DBG$SRC_NEXT_MODRSTPTR = 0;
;  3323     3439  2      DBG$SRC_SEARCH_FLAG = .SEARCH_FLAG;
;  3324     3440  2      DBG$SRC_FORMFEED_FLAG = FALSE;
;  3325     3441  2      LINE_NUM = 1;
;  3326     3442  2      STMT_NUM = 0;
;  3327     3443  2      SEARCH_FLAG = FALSE;
;  3328     3444  2      SEARCH_FND = FALSE;
;  3329     3445  2      SFIT_PTR = 0;
;  3330     3446  2      SRC_DSTPTR = 0;
```

```
;  3331     3447  2      SRC_REC = 1;
;  3332     3448  2      STATEMENT_MODE = FALSE;
;  3333     3449  2      SRC_FILEID_TBL = 0;
;  3334     3450  2      TYPE_FLAG = FALSE;
;  3335     3451  2
;  3336     3452  2
;  3337     3453  2      ! Points to the current modules's Module RST Entry, this is later used
;  3338     3454  2      ! by DBG$SRC_TYPE_LINE routine.  And also get the Module Name.
;  3339     3455  2      !
;  3340     3456  2      DBG$SRC_MODRSTPTR = .MODRSTPTR;
;  3341     3457  2      DBG$STA_SYMNAME(.MODRSTPTR,MODNAMEPTR);
;  3342     3458  2
;  3343     3459  2
;  3344     3460  2      ! Make sure Module Source Table pointer in module's Module RST Entry
;  3345     3461  2      ! points to a list of Source File DST Record Pointers.
;  3346     3462  2      !
;  3347     3463  2      IF .MODRSTPTR[RST$L_MODSRCTBL] EQL 0
;  3348     3464  2      THEN
;  3349     3465  3          BEGIN
;  3350     3466  3          SIGNAL(DBG$_SRCLINNOT,1,.MODNAMEPTR);
;  3351     3467  3          RETURN;
;  3352     3468  2          END;
;  3353     3469  2
;  3354     3470  2
;  3355     3471  2      ! Loop thru Module's Source Line Correlation DST Records.
;  3356     3472  2      !
;  3357     3473  2      MODSRCTBL = .MODRSTPTR[RST$L_MODSRCTBL];
;  3358     3474  2      INCR COUNT FROM 1 TO .MODSRCTBL[0] DO
;  3359     3475  3          BEGIN
;  3360     3476  4          IF (.LINE_NUM GTR .HIGH_LNUM) AND (.STMT_NUM GEQ .HIGH_STMT)
;  3361     3477  3          THEN
;  3362     3478  3              EXITLOOP;
;  3363     3479  3
;  3364     3480  3
;  3365     3481  3          ! We have found the 1st match string for the SEARCH/IDENT/NEXT or
;  3366     3482  3          ! SEARCH/STRING/NEXT command.  Exit the loop.
;  3367     3483  3          !
;  3368     3484  3          IF .SEARCH_FND AND
;  3369     3485  4             ((.DBG$SRC_SEARCH_FLAG EQL SEARCH_IDENT_NEXT) OR
;  3370     3486  4              (.DBG$SRC_SEARCH_FLAG EQL SEARCH_STRING_NEXT))
;  3371     3487  3          THEN
;  3372     3488  3              EXITLOOP;
;  3373     3489  3
;  3374     3490  3
;  3375     3491  3          ! Get Source Line Correlation DST Record pointer.
;  3376     3492  3          !
;  3377     3493  3          DSTREC_PTR = .MODSRCTBL[.COUNT];
;  3378     3494  3          IF .DSTREC_PTR[DST$B_TYPE] NEQ DST$K_SOURCE
;  3379     3495  3          THEN
;  3380     3496  3              $DBG_ERROR('DBGSOURCE\DBG$SRC_TYPE_LNUM_SOURCE 20');
;  3381     3497  3
;  3382     3498  3
;  3383     3499  3          ! Scans thru Source File DST Record to decode the Source File Correlation
;  3384     3500  3          ! Commands.  Pick up the length of this DST record.  Set pointer to the
;  3385     3501  3          ! first command entry.
;  3386     3502  3          !
;  3387     3503  3          LENGTH = .DSTREC_PTR[DST$B_LENGTH];
```

```
; 3388          3504  3          DSTPTR = DSTREC_PTR[DST$A_SRC_FIRST_CMD];
; 3389          3505  3          WHILE (.DSTPTR = .DSTREC_PTR = 1) NEQ .LENGTH DO
; 3390          3506  4              BEGIN
; 3391          3507  5              IF (.LINE_NUM GTR .HIGH_LNUM) AND (.STMT_NUM GEQ .HIGH_STMT)
; 3392          3508  4              THEN
; 3393          3509  4                  EXITLOOP;
; 3394          3510  4
; 3395          3511  4
; 3396          3512  4              ! We have found the 1st match string for SEARCH/IDENT/NEXT or
; 3397          3513  4              ! SEARCH/STRING/NEXT command.  Exit the loop.
; 3398          3514  4              !
; 3399          3515  4              IF .SEARCH_FND AND
; 3400          3516  5                 ((.DBG$SRC_SEARCH_FLAG EQL SEARCH_IDENT_NEXT) OR
; 3401          3517  5                  (.DBG$SRC_SEARCH_FLAG EQL SEARCH_STRING_NEXT))
; 3402          3518  4              THEN
; 3403          3519  4                  EXITLOOP;
; 3404          3520  4
; 3405          3521  4
; 3406          3522  4              ! Pick up the command code defined in Source File DST Record
; 3407          3523  4              ! and use it as a CASE index so each significant command code
; 3408          3524  4              ! can be interpreted individually.
; 3409          3525  4              !
; 3410          3526  4              CASE .DSTPTR[DST$B_SRC_COMMAND] FROM DST$K_SRC_MIN_CMD TO
; 3411          3527  4                                                  DST$K_SRC_MAX_CMD OF
; 3412          3528  4                  SET
; 3413          3529  4
; 3414          3530  4
; 3415          3531  4                  ! Declare Source File Command.  Allocate some space to build
; 3416          3532  4                  ! the File ID Table.
; 3417          3533  4                  !
; 3418          3534  4                  [DST$K_SRC_DECLFILE]:
; 3419          3535  5                      BEGIN
; 3420          3536  5                      IF .SFIT_PTR NEQ 0 THEN SFIT_PTR[SFIT$L_CURRECNUM] = .SRC_REC;
; 3421          3537  5                      SRC_REC = 1;
; 3422          3538  5                      SFIT_PTR = DBG$GET_TEMPMEM(SFIT$K_SIZE);
; 3423          3539  5                      SFIT_PTR[SFIT$L_FLINK] = .SRC_FILEID_TBL;
; 3424          3540  5                      SFIT_PTR[SFIT$L_FILE_ID] = .DSTPTR[DST$W_SRC_DF_FILEID];
; 3425          3541  5                      SFIT_PTR[SFIT$L_DSTPTR] = .DSTPTR;
; 3426          3542  5                      SFIT_PTR[SFIT$L_CURRECNUM] = .SRC_REC;
; 3427          3543  5                      SRC_FILEID_TBL = .SFIT_PTR;
; 3428          3544  5                      DSTPTR = .DSTPTR + .DSTPTR[DST$B_SRC_DF_LENGTH] + 2;
; 3429          3545  4                      END;
; 3430          3546  4
; 3431          3547  4
; 3432          3548  4                  ! Set Source File Command.  This command sets the current
; 3433          3549  4                  ! source file to the file denoted by the File ID given in
; 3434          3550  4                  ! the command.  This File ID must be pre-defined by the
; 3435          3551  4                  ! Declare Source File Command.
; 3436          3552  4                  !
; 3437          3553  4                  [DST$K_SRC_SETFILE]:
; 3438          3554  5                      BEGIN
; 3439          3555  5                      IF .SFIT_PTR NEQ 0 THEN SFIT_PTR[SFIT$L_CURRECNUM] = .SRC_REC;
; 3440          3556  5                      SRC_DSTPTR = 0;
; 3441          3557  5                      SFIT_PTR = .SRC_FILEID_TBL;
; 3442          3558  5                      WHILE .SFIT_PTR NEQ 0 DO
; 3443          3559  6                          BEGIN
; 3444          3560  6                          IF .DSTPTR[DST$W_SRC_UNSWORD] EQL .SFIT_PTR[SFIT$L_FILE_ID]
```

```
;  3445     3561  6                    THEN
;  3446     3562  7                        BEGIN
;  3447     3563  7                        SRC_DSTPTR = .SFIT_PTR[SFIT$L_DSTPTR];
;  3448     3564  7                        SRC_REC = .SFIT_PTR[SFIT$L_CURRECNUM];
;  3449     3565  7                        EXITLOOP;
;  3450     3566  6                        END;
;  3451     3567  6
;  3452     3568  6                    SFIT_PTR = .SFIT_PTR[SFIT$L_FLINK];
;  3453     3569  5                    END;
;  3454     3570  5
;  3455     3571  5
;  3456     3572  5                ! If this File ID has not yet been declared, signal an
;  3457     3573  5                ! invalid DST Record.
;  3458     3574  5                !
;  3459     3575  5                IF .SRC_DSTPTR EQL 0 THEN SIGNAL(DBG$_INVDSTREC);
;  3460     3576  5                DSTPTR = .DSTPTR + 3;
;  3461     3577  4                END;
;  3462     3578  4
;  3463     3579  4
;  3464     3580  4            ! Set Source Record Number Long Command.  Set the current source
;  3465     3581  4            ! file record number.  Advance DSTPTR.
;  3466     3582  4            !
;  3467     3583  4            [DST$K_SRC_SETREC_L]:
;  3468     3584  5                BEGIN
;  3469     3585  5                SRC_REC = .DSTPTR[DST$L_SRC_UNSLONG];
;  3470     3586  5                DSTPTR = .DSTPTR + 5;
;  3471     3587  4                END;
;  3472     3588  4
;  3473     3589  4
;  3474     3590  4            ! Set Source Record Number Word Command. (More compact form).
;  3475     3591  4            !
;  3476     3592  4            [DST$K_SRC_SETREC_W]:
;  3477     3593  5                BEGIN
;  3478     3594  5                SRC_REC = .DSTPTR[DST$W_SRC_UNSWORD];
;  3479     3595  5                DSTPTR = .DSTPTR + 3;
;  3480     3596  4                END;
;  3481     3597  4
;  3482     3598  4
;  3483     3599  4            ! Set Line Number Long Command.  Set the current listing line
;  3484     3600  4            ! number.  Advance DSTPTR.
;  3485     3601  4            !
;  3486     3602  4            [DST$K_SRC_SETLNUM_L]:
;  3487     3603  5                BEGIN
;  3488     3604  5                LINE_NUM = .DSTPTR[DST$L_SRC_UNSLONG];
;  3489     3605  5                DSTPTR = .DSTPTR + 5;
;  3490     3606  4                END;
;  3491     3607  4
;  3492     3608  4
;  3493     3609  4            ! Set Line Number Word Command. (More compact form).
;  3494     3610  4            !
;  3495     3611  4            [DST$K_SRC_SETLNUM_W]:
;  3496     3612  5                BEGIN
;  3497     3613  5                LINE_NUM = .DSTPTR[DST$W_SRC_UNSWORD];
;  3498     3614  5                DSTPTR = .DSTPTR + 3;
;  3499     3615  4                END;
;  3500     3616  4
;  3501     3617  4
```

```
;  3502     3618  4              ! Increment Line Number Byte Command.  Increment the current
;  3503     3619  4              ! listing line number by a byte value given in this command.
;  3504     3620  4              ! Advance DSTPTR.
;  3505     3621  4              !
;  3506     3622  4              [DST$K_SRC_INCRLNUM_B]:
;  3507     3623  5                  BEGIN
;  3508     3624  5                  LINE_NUM = .LINE_NUM + .DSTPTR[DST$B_SRC_UNSBYTE];
;  3509     3625  5                  DSTPTR = .DSTPTR + 2;
;  3510     3626  4                  END;
;  3511     3627  4
;  3512     3628  4
;  3513     3629  4              ! Count Form-Feeds as Source Records Command.  This command says
;  3514     3630  4              ! that form-feeds in the source should be counted as separate
;  3515     3631  4              ! source records; the absence of this command says that such
;  3516     3632  4              ! records should be ignored as control information and should
;  3517     3633  4              ! not be assigned line nubmers.
;  3518     3634  4              !
;  3519     3635  4              [DST$K_SRC_FORMFEED]:
;  3520     3636  5                  BEGIN
;  3521     3637  5                  DBG$SRC_FORMFEED_FLAG = TRUE;
;  3522     3638  5                  DSTPTR = .DSTPTR + 1;
;  3523     3639  4                  END;
;  3524     3640  4
;  3525     3641  4
;  3526     3642  4              ! Define N Separate Lines Word Command, and Define N Separate
;  3527     3643  4              ! Lines Byte Command (More Compact Form).  If a Source File is not
;  3528     3644  4              ! set, signal and give up.  Defines the Source File and Source
;  3529     3645  4              ! Record Numbers for a specified number of Listing Line Numbers.
;  3530     3646  4              ! Call TYPE LINE routine, if LOW_LNUM <= LINE_NUM <= HIGH_LNUM.
;  3531     3647  4              !
;  3532     3648  4              [DST$K_SRC_DEFLINES_W, DST$K_SRC_DEFLINES_B]:
;  3533     3649  5                  BEGIN
;  3534     3650  5                  IF .SRC_DSTPTR EQL 0 THEN SIGNAL(DBG$_INVDSTREC);
;  3535     3651  5
;  3536     3652  5
;  3537     3653  5                  ! Calculate the listing line number range.
;  3538     3654  5                  !
;  3539     3655  5                  IF .DSTPTR[DST$B_SRC_COMMAND] EQL DST$K_SRC_DEFLINES_W
;  3540     3656  5                  THEN
;  3541     3657  5                      HIGH_LINE_NUM = .LINE_NUM + .DSTPTR[DST$W_SRC_UNSWORD] - 1
;  3542     3658  5                  ELSE
;  3543     3659  5                      HIGH_LINE_NUM = .LINE_NUM + .DSTPTR[DST$B_SRC_UNSBYTE] - 1;
;  3544     3660  5
;  3545     3661  5
;  3546     3662  5                  ! Test to see if the listing line number exceeds the type
;  3547     3663  5                  ! line range.  If it is, increments the source record number
;  3548     3664  5                  ! listing line number and DST pointer, then returns.
;  3549     3665  5                  !
;  3550     3666  5                  IF (.LINE_NUM GTR .HIGH_LNUM) OR
;  3551     3667  6                     (.HIGH_LINE_NUM LSS .LOW_LNUM)
;  3552     3668  5                  THEN
;  3553     3669  6                      BEGIN
;  3554     3670  6                      IF .DSTPTR[DST$B_SRC_COMMAND] EQL DST$K_SRC_DEFLINES_W
;  3555     3671  6                      THEN
;  3556     3672  6                          SRC_REC = .SRC_REC + .DSTPTR[DST$W_SRC_UNSWORD]
;  3557     3673  6                      ELSE
;  3558     3674  6                          SRC_REC = .SRC_REC + .DSTPTR[DST$B_SRC_UNSBYTE];
```

```
;  3559     3675  6
;  3560     3676  6                  END
;  3561     3677  6
;  3562     3678  5              ELSE
;  3563     3679  6                  BEGIN
;  3564     3680  6                  TYPE_FLAG = TRUE;
;  3565     3681  6
;  3566     3682  6
;  3567     3683  6                  ! Synchronize the Source Record Number with the Listing Line
;  3568     3684  6                  ! Number.
;  3569     3685  6                  !
;  3570     3686  6                  IF (.LOW_LNUM - .LINE_NUM) GTR 0
;  3571     3687  6                  THEN
;  3572     3688  6                      SRC_REC = .SRC_REC + (.LOW_LNUM - .LINE_NUM);
;  3573     3689  6
;  3574     3690  6                  INCR I FROM MAX(.LOW_LNUM,.LINE_NUM) TO
;  3575     3691  6                             MIN(.HIGH_LNUM,.HIGH_LINE_NUM) DO
;  3576     3692  7                      BEGIN
;  3577     3693  7                      SEARCH_FND = DBG$SRC_TYPE_LINE(.I, .STMT_NUM,
;  3578     3694  7                                   .SRC_DSTPTR, .SRC_REC, .STEP_FLAG);
;  3579     3695  7                      IF .SEARCH_FND THEN SEARCH_FLAG = TRUE;
;  3580     3696  7                      SRC_REC = .SRC_REC + 1;
;  3581     3697  7                      IF .SEARCH_FND AND
;  3582     3698  8                         ((.DBG$SRC_SEARCH_FLAG EQL SEARCH_IDENT_NEXT) OR
;  3583     3699  8                          (.DBG$SRC_SEARCH_FLAG EQL SEARCH_STRING_NEXT))
;  3584     3700  7                      THEN
;  3585     3701  7                          EXITLOOP;
;  3586     3702  7
;  3587     3703  6                      END;
;  3588     3704  6
;  3589     3705  5                  END;
;  3590     3706  5
;  3591     3707  5                      LINE_NUM = .HIGH_LINE_NUM + 1;
;  3592     3708  5                      IF .DSTPTR[DST$B_SRC_COMMAND] EQL DST$K_SRC_DEFLINES_W
;  3593     3709  5                      THEN
;  3594     3710  5                          DSTPTR = .DSTPTR + 3
;  3595     3711  5                      ELSE
;  3596     3712  5                          DSTPTR = .DSTPTR + 2;
;  3597     3713  5
;  3598     3714  4                      END;
;  3599     3715  4
;  3600     3716  4
;  3601     3717  4                  ! Any other command code constitutes an error in the DST.
;  3602     3718  4                  !
;  3603     3719  4                  [INRANGE, OUTRANGE]:
;  3604     3720  4                      SIGNAL(DBG$_INVDSTREC);
;  3605     3721  4
;  3606     3722  4                  TES;
;  3607     3723  4
;  3608     3724  3              END;                        ! End of WHILE Loop through commands.
;  3609     3725  3
;  3610     3726  2          END;                            ! End of INCR Loop through DST records.
;  3611     3727  2
;  3612     3728  2
;  3613     3729  2      ! If we were outputting source lines to a screen display, close out that
;  3614     3730  2      ! screen display properly.
;  3615     3731  2      !
```

```
;  3616     3732  2      IF .SOURCE_TO_SCREEN_FLAG THEN DBG$SCR_SOURCE_END();
;  3617     3733  2
;  3618     3734  2
;  3619     3735  2      ! Issue a warning message if no lines were typed out.  This situation
;  3620     3736  2      ! occurs if the specified line numbers do not exist in this module.
;  3621     3737  2      ! If it is in the search mode, signal no matches instead.
;  3622     3738  2      !
;  3623     3739  2      IF NOT .TYPE_FLAG
;  3624     3740  2      THEN
;  3625     3741  2          IF .DBG$SRC_SEARCH_FLAG EQL 0
;  3626     3742  2          THEN
;  3627     3743  3              BEGIN
;  3628     3744  3              BUF_DESC[0] = 139;
;  3629     3745  3              BUF_DESC[1] = BUFFER[1];
;  3630     3746  4              IF (.LOWLNUM EQL .HIGHLNUM) AND (.LOWSTMT EQL .HIGHSTMT)
;  3631     3747  3              THEN
;  3632     3748  3                  DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC 'line '), 0)
;  3633     3749  3              ELSE
;  3634     3750  3                  DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC 'lines '), 0);
;  3635     3751  3
;  3636     3752  3              DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC '!UL'), .LOWLNUM);
;  3637     3753  3              IF .LOWSTMT NEQ 0
;  3638     3754  3              THEN
;  3639     3755  3                  DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC '.!UL'), .LOWSTMT);
;  3640     3756  3
;  3641     3757  4              IF (.LOWLNUM NEQ .HIGHLNUM) OR (.LOWSTMT NEQ .HIGHSTMT)
;  3642     3758  3              THEN
;  3643     3759  3                  DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC ':!UL'), .HIGHLNUM);
;  3644     3760  3
;  3645     3761  3              IF .HIGHSTMT NEQ 0
;  3646     3762  3              THEN
;  3647     3763  3                  DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC '.!UL'), .HIGHSTMT);
;  3648     3764  3
;  3649     3765  4              IF (.LOWLNUM EQL .HIGHLNUM) AND (.LOWSTMT EQL .HIGHSTMT)
;  3650     3766  3              THEN
;  3651     3767  3                  DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC ' does'), 0)
;  3652     3768  3              ELSE
;  3653     3769  3                  DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC ' do'), 0);
;  3654     3770  3
;  3655     3771  3              DBG$FORMAT_FAO_OUT(BUF_DESC, UPLIT BYTE(%ASCIC ' not exist in module !AC'),
;  3656     3772  3                      .MODNAMEPTR);
;  3657     3773  3              BUFFER[0] = 139 - .BUF_DESC[0];
;  3658     3774  3              SIGNAL(DBG$_NOLINXXX, 1, BUFFER);
;  3659     3775  3              END
;  3660     3776  3
;  3661     3777  3
;  3662     3778  3          ! If we were in a SEARCH command and output no lines, signal no matches.
;  3663     3779  3          !
;  3664     3780  2          ELSE
;  3665     3781  2              SIGNAL(DBG$_NOMATCH);
;  3666     3782  2
;  3667     3783  2
;  3668     3784  2      ! Check to see if any matches were found for the SEARCH command.
;  3669     3785  2      !
;  3670     3786  2      IF .TYPE_FLAG AND NOT .SEARCH_FLAG AND .DBG$SRC_SEARCH_FLAG NEQ 0
;  3671     3787  2      THEN
;  3672     3788  2          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  00081          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
```

```
                        98   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
```

```
; 3678          3793  1 GLOBAL ROUTINE DBG$SRC_TYPE_PC_SOURCE(LOW_PC, HIGH_PC, MODPRTFLG, STEP_FLAG): NOVALUE =
; 3679          3794  1 !
; 3680          3795  1 ! FUNCTION
; 3681          3796  1 !       This routine accepts a range of Program Counter values and prints all
; 3682          3797  1 !       source lines which cover that PC range.  This assumes that both PC
; 3683          3798  1 !       values are located in the same module and that a line number is associ-
; 3684          3799  1 !       ated with each end of the PC range.  It also assums that the lower PC
; 3685          3800  1 !       value is associated with the lower line number of the range.  This
; 3686          3801  1 !       routine is called in the processing of the EXAMINE/SOURCE command and
; 3687          3802  1 !       does the source display in response to breakpoints, watchpoints, and
; 3688          3803  1 !       stepping by lines.
; 3689          3804  1 !
; 3690          3805  1 !       The routine calls DBG$PC_TO_LINE_LOOKUP for the two PC values to get the corre-
; 3691          3806  1 !       sponding line numbers.  If both PC values are the same (which is always
; 3692          3807  1 !       the case on breakpoints or when stepping), DBG$PC_TO_LINE_LOOKUP is only called
; 3693          3808  1 !       once.  The routine then checks that both PC values are in the same mod-
; 3694          3809  1 !       ule and signals an error if they are not.  It also signals an error if
; 3695          3810  1 !       one or the other PC value does not correspond to a line number or if the
; 3696          3811  1 !       line numbers they correspond to are not in the proper order.  Routine
; 3697          3812  1 !       DBG$SRC_TYPE_LNUM_SOURCE is then called to retrieve and type the source
; 3698          3813  1 !       images.
; 3699          3814  1 !
; 3700          3815  1 ! INPUTS
; 3701          3816  1 !       LOW_PC  - The lower PC value of the address range whose source lines
; 3702          3817  1 !                 are to be displayed.
; 3703          3818  1 !
; 3704          3819  1 !       HIGH_PC - The higher PC value of the address range whose source lines
; 3705          3820  1 !                 are to be displayed.  HIGH_PC must be greater than or equal
; 3706          3821  1 !                 to LOW_PC.
; 3707          3822  1 !
; 3708          3823  1 !       MODPRTFLG - Flag set to FALSE when the routine is called in reponse to
; 3709          3824  1 !                 breakpoints, watchpoints, and stepping by lines, so that the
; 3710          3825  1 !                 module name will not be printed.
; 3711          3826  1 !
; 3712          3827  1 !       STEP_FLAG - Flag set to TRUE to indicate the call is from STEP.
; 3713          3828  1 !
; 3714          3829  1 ! OUTPUTS
; 3715          3830  1 !       NONE
; 3716          3831  1 !
; 3717          3832  1
; 3718          3833  2     BEGIN
; 3719          3834  2
; 3720          3835  2     ENABLE
; 3721          3836  2         TYPE_PC_SOURCE_HANDLER;
; 3722          3837  2
; 3723          3838  2     LOCAL
; 3724          3839  2         LLINE_END,                          ! Low End PC of the selected line/stmt
; 3725          3840  2         LLINE_NUM,                          ! Low line number
; 3726          3841  2         LLINE_START,                        ! Low Start PC of the selected line/stmt
; 3727          3842  2         LMODNAMEPTR,                        ! Low PC module name pointer
; 3728          3843  2         LOW_MODPTR : REF RST$ENTRY,         ! Low PC module pointer
; 3729          3844  2         LSTMT_NUM,                          ! Low stmt number
; 3730          3845  2         HLINE_END,                          ! High End PC of the selected line/stmt
; 3731          3846  2         HLINE_NUM,                          ! High line number
; 3732          3847  2         HLINE_START,                        ! High Start PC of the selected line/stmt
; 3733          3848  2         HMODNAMEPTR,                        ! High PC module name pointer
; 3734          3849  2         HIGH_MODPTR,                        ! High PC module pointer
```

```
;3735     3850  2          HSTMT_NUM,                      ! High stmt number
;3736     3851  2          STATUS;                         ! return status from PC to LINE translation
;3737     3852  2
;3738     3853  2
;3739     3854  2
;3740     3855  2      ! Check to see if the PC is an entry point, if it is then PC = PC + 2.
;3741     3856  2      ! Separately handle the case where LOW_PC = HIGH_PC, so that we only
;3742     3857  2      ! have to call IS_IT_ENTRY once.
;3743     3858  2      !
;3744     3859  2      IF .HIGH_PC EQL .LOW_PC
;3745     3860  2      THEN
;3746     3861  2          BEGIN
;3747     3862  2          IF (DBG$IS_IT_ENTRY(.LOW_PC))
;3748     3863  2          THEN
;3749     3864  2              BEGIN
;3750     3865  2              LOW_PC = .LOW_PC + 2;
;3751     3866  2              HIGH_PC = .LOW_PC;
;3752     3867  2              END;
;3753     3868  2          END
;3754     3869  2
;3755     3870  2      ELSE
;3756     3871  2          BEGIN
;3757     3872  2          IF (DBG$IS_IT_ENTRY(.LOW_PC)) THEN LOW_PC = .LOW_PC + 2;
;3758     3873  2          IF (DBG$IS_IT_ENTRY(.HIGH_PC)) THEN HIGH_PC = .HIGH_PC + 2;
;3759     3874  2          END;
;3760     3875  2
;3761     3876  2
;3762     3877  2      ! Ensure that higher PC value is greater than or equal to lower PC value.
;3763     3878  2      ! If it is wrong, signal an error and give up.
;3764     3879  2      !
;3765     3880  2      IF (.LOW_PC GTR .HIGH_PC) THEN SIGNAL(DBG$_EXARANGE);
;3766     3881  2
;3767     3882  2
;3768     3883  2      ! Match an absolute PC address to a line number.
;3769     3884  2      !
;3770     3885  2      LOW_MODPTR = 0;
;3771     3886  2      STATUS = DBG$PC_TO_LINE_LOOKUP(.LOW_PC, LLINE_NUM, LSTMT_NUM, LLINE_START,
;3772     3887  2                                     LLINE_END, LOW_MODPTR);
;3773     3888  2
;3774     3889  2
;3775     3890  2      ! Check to see if any error or if match cannot be made in PC to LINE
;3776     3891  2      ! translation.
;3777     3892  2      !
;3778     3893  2      IF NOT .STATUS
;3779     3894  2      THEN
;3780     3895  3          BEGIN
;3781     3896  3          IF .MODPRTFLG
;3782     3897  3          THEN
;3783     3898  3              SIGNAL(DBG$_NOSRCLIN,1,.LOW_PC)
;3784     3899  3          ELSE
;3785     3900  3              RETURN;
;3786     3901  3
;3787     3902  2          END;
;3788     3903  2
;3789     3904  2
;3790     3905  2      ! Get Module Name for lower PC value.
;3791     3906  2      !
```

```
;  3792          3907  2      DBG$STA_SYMNAME(.LOW_MODPTR,LMODNAMEPTR);
;  3793          3908  2
;  3794          3909  2
;  3795          3910  2      ! If two PC values are not the same, calls PC to LINE routine for the
;  3796          3911  2      ! higher PC value to get the corresponding line number.  If both PC values
;  3797          3912  2      ! are the same, set up parameters for routine TYPE_LNUM_SOURCE.
;  3798          3913  2      !
;  3799          3914  2      IF .LOW_PC EQL .HIGH_PC
;  3800          3915  2      THEN
;  3801          3916  3          BEGIN
;  3802          3917  3          HLINE_NUM = .LLINE_NUM;
;  3803          3918  3          HSTMT_NUM = .LSTMT_NUM;
;  3804          3919  3          END
;  3805          3920  2      ELSE
;  3806          3921  3          BEGIN
;  3807          3922  3          HIGH_MODPTR = 0;
;  3808          3923  3          STATUS = DBG$PC_TO_LINE_LOOKUP(.HIGH_PC,HLINE_NUM,HSTMT_NUM,HLINE_START,
;  3809          3924  3                                         HLINE_END,HIGH_MODPTR);
;  3810          3925  3
;  3811          3926  3
;  3812          3927  3          ! Check to see if any error or if match cannot be made in PC to LINE
;  3813          3928  3          ! translation.
;  3814          3929  3          !
;  3815          3930  3          IF NOT .STATUS
;  3816          3931  3          THEN
;  3817          3932  4              BEGIN
;  3818          3933  4              IF .MODPRTFLG
;  3819          3934  4              THEN
;  3820          3935  4                  SIGNAL(DBG$_NOSRCLIN,1,.HIGH_PC)
;  3821          3936  4              ELSE
;  3822          3937  4                  RETURN;
;  3823          3938  4
;  3824          3939  3              END;
;  3825          3940  3
;  3826          3941  3
;  3827          3942  3          ! Get the Module Name for higher PC value.
;  3828          3943  3          !
;  3829          3944  3          DBG$STA_SYMNAME(.HIGH_MODPTR,HMODNAMEPTR);
;  3830          3945  3
;  3831          3946  3
;  3832          3947  3          ! Check that both PC values are in the same module.
;  3833          3948  3          !
;  3834          3949  3          IF .LOW_MODPTR NEQ .HIGH_MODPTR
;  3835          3950  3          THEN
;  3836          3951  4              BEGIN
;  3837          3952  4              IF .MODPRTFLG
;  3838          3953  4              THEN
;  3839          3954  4                  SIGNAL(DBG$_ADDRANCOV,4,.LOW_PC,.LMODNAMEPTR,.HIGH_PC,.HMODNAMEPTR)
;  3840          3955  4              ELSE
;  3841          3956  4                  RETURN;
;  3842          3957  4
;  3843          3958  3              END;
;  3844          3959  3
;  3845          3960  2          END;            ! End of matching high PC value to a line number.
;  3846          3961  2
;  3847          3962  2
;  3848          3963  2      ! Check for the entry point.
```

```
;  3849          3964  2     !
;  3850          3965  3     IF (.LLINE_NUM EQL 0) AND (.HLINE_NUM EQL 0)
;  3851          3966  2     THEN
;  3852          3967  3         BEGIN
;  3853          3968  3         IF .MODPRTFLG
;  3854          3969  3         THEN
;  3855          3970  3             SIGNAL(DBG$_NOSRCLIN,1,.LOW_PC)
;  3856          3971  3         ELSE
;  3857          3972  3             RETURN;
;  3858          3973  3
;  3859          3974  2         END;
;  3860          3975  2
;  3861          3976  2
;  3862          3977  2     ! Make sure Module Source Table pointer in module's Module RST Entry
;  3863          3978  2     ! points to a list of Source File DST Record Pointers.
;  3864          3979  2     ! If not, then check whether we have reached this point through a step
;  3865          3980  2     ! or breakpoint. In this case we do not want to signal an error, so
;  3866          3981  2     ! we just return.
;  3867          3982  2     !
;  3868          3983  3     IF (.LOW_MODPTR[RST$L_MODSRCTBL] EQL 0) AND (NOT .MODPRTFLG)
;  3869          3984  2     THEN
;  3870          3985  2         RETURN;
;  3871          3986  2
;  3872          3987  2
;  3873          3988  2     ! Print out Module name if the print-module-name flag is set to TRUE.
;  3874          3989  2     ! Note that the message is suppressed if the source line output is
;  3875          3990  2     ! being directed to a source screen display.
;  3876          3991  2     !
;  3877          3992  3     IF .MODPRTFLG AND (.DBG$GL_SCREEN_SOURCE EQL 0)
;  3878          3993  2     THEN
;  3879          3994  3         BEGIN
;  3880          3995  3         DBG$PRINT(UPLIT BYTE(%ASCIC 'module !AC'), .LMODNAMEPTR);
;  3881          3996  3         DBG$NEWLINE();
;  3882          3997  2         END;
;  3883          3998  2
;  3884          3999  2
;  3885          4000  2     ! Call TYPE_LNUM_SOURCE to type out the given source line number ranges.
;  3886          4001  2     !
;  3887          4002  2     DBG$SRC_TYPE_LNUM_SOURCE(.LOW_MODPTR, .LLINE_NUM, .LSTMT_NUM,
;  3888          4003  2                              .HLINE_NUM, .HSTMT_NUM, 0, .STEP_FLAG);
;  3889          4004  2
;  3890          4005  2     RETURN;
;  3891          4006  2
;  3892          4007  1     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
```

```
;  3894          4008  1 ROUTINE OUTPUT_DIR_LIST (DIRPTR) : NOVALUE =
;  3895          4009  1 !
;  3896          4010  1 ! FUNCTION
;  3897          4011  1 !       This routine outputs the Source Directory Search List
;  3898          4012  1 !       for one module (or the default list) on the user's terminal.
;  3899          4013  1 !       It prints each directory name on a separate line, indented
;  3900          4014  1 !       eight spaces. This routine is called during the SHOW SOURCE
;  3901          4015  1 !       command.
;  3902          4016  1 !
;  3903          4017  1 ! INPUTS
;  3904          4018  1 !       DIRPTR  -       The first of a linked list of Source Directory
;  3905          4019  1 !                       Search List Entries.
;  3906          4020  1 !
;  3907          4021  1 ! OUTPUTS
;  3908          4022  1 !       NONE
;  3909          4023  1
;  3910          4024  1
;  3911          4025  2     BEGIN
;  3912          4026  2     MAP
;  3913          4027  2         DIRPTR : REF SDSL$ENTRY;        ! A pointer to a Source Directory
;  3914          4028  2                                         !       Search List Entry. This is
;  3915          4029  2                                         !       used to walk through the
;  3916          4030  2                                         !       linked list.
;  3917          4031  2
;  3918          4032  2
;  3919          4033  2
;  3920          4034  2     ! Loop through the list, outputting the directory name in each Source
;  3921          4035  2     ! Directory Search List Entry.
;  3922          4036  2     !
;  3923          4037  2     WHILE .DIRPTR NEQ 0 DO
;  3924          4038  3         BEGIN
;  3925          4039  3         DBG$PRINT (UPLIT BYTE(
;  3926          4040  3             %ASCIC '        !AC'), DIRPTR[SDSL$B_ENT_DIRLEN]);
;  3927          4041  3         DBG$NEWLINE();
;  3928          4042  3         DIRPTR = .DIRPTR[SDSL$L_ENT_FLINK];
;  3929          4043  2         END;
;  3930          4044  2
;  3931          4045  2     RETURN;
;  3932          4046  2
;  3933          4047  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