```
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR        AAAAAAAAA     RRRRRRRRRRRR
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR        AAAAAAAAA     RRRRRRRRRRRR
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR        AAAAAAAAA     RRRRRRRRRRRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR     AAA         AAA  RRRRRRRRRRRR
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR     AAA         AAA  RRRRRRRRRRRR
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR     AAA         AAA  RRRRRRRRRRRR
LLL                    III        BBB         BBB  RRR   RRR        AAAAAAAAAAAAAAA  RRR   RRR
LLL                    III        BBB         BBB  RRR   RRR        AAAAAAAAAAAAAAA  RRR   RRR
LLL                    III        BBB         BBB  RRR   RRR        AAAAAAAAAAAAAAA  RRR   RRR
LLL                    III        BBB         BBB  RRR      RRR     AAA         AAA  RRR      RRR
LLL                    III        BBB         BBB  RRR      RRR     AAA         AAA  RRR      RRR
LLL                    III        BBB         BBB  RRR      RRR     AAA         AAA  RRR      RRR
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR  AAA         AAA  RRR         RRR
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR  AAA         AAA  RRR         RRR
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR  AAA         AAA  RRR         RRR
```

FILEID**SUBS

F 7

```
  SSSSSSSS  UU      UU  BBBBBBBB      SSSSSSSS
  SSSSSSSS  UU      UU  BBBBBBBB      SSSSSSSS
SS          UU      UU  BB      BB  SS
SS          UU      UU  BB      BB  SS
SS          UU      UU  BB      BB  SS
SS          UU      UU  BB      BB  SS
  SSSSSS    UU      UU  BBBBBBBB      SSSSSS
  SSSSSS    UU      UU  BBBBBBBB      SSSSSS
        SS  UU      UU  BB      BB        SS
        SS  UU      UU  BB      BB        SS
        SS  UU      UU  BB      BB        SS      ....
        SS  UU      UU  BB      BB        SS      ....
SSSSSSSS    UUUUUUUUUU  BBBBBBBB    SSSSSSSS        ....
SSSSSSSS    UUUUUUUUUU  BBBBBBBB    SSSSSSSS        ....

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

```
EXE

Mod
---
LIB
LIB
LIB
LIB
LIB

LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB

LIB
LIB
LIB
LIB
LIB

LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB
LIB

LIB
LIB
LIB
```

```
;     1          0001  0 MODULE LIB_SUBS (                        ! General subroutines for LIBRARIAN
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 IDENT = 'V04-000'
;     4          0004  0                 ) =
;     5          0005  1 BEGIN
;     6          0006  1
;     7          0007  1 !
;     8          0008  1 !**************************************************************************
;     9          0009  1 !*                                                                        *
;    10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;    11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    12          0012  1 !*  ALL RIGHTS RESERVED.                                                   *
;    13          0013  1 !*                                                                        *
;    14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    19          0019  1 !*  TRANSFERRED.                                                           *
;    20          0020  1 !*                                                                        *
;    21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    23          0023  1 !*  CORPORATION.                                                           *
;    24          0024  1 !*                                                                        *
;    25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    27          0027  1 !*                                                                        *
;    28          0028  1 !*                                                                        *
;    29          0029  1 !**************************************************************************
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 !
;    33          0033  1 ! FACILITY:  Library command processor
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       The VAX/VMS librarian is invoked by DCL to process the LIBRARY
;    38          0038  1 !       command.  It utilizes the librarian procedure set to perform
;    39          0039  1 !       the actual modifications to the library.
;    40          0040  1 !
;    41          0041  1 ! ENVIRONMENT:
;    42          0042  1 !
;    43          0043  1 !       VAX native, user mode.
;    44          0044  1 !
;    45          0045  1 !--
;    46          0046  1 !
;    47          0047  1 !
;    48          0048  1 ! AUTHOR:  Benn Schreiber,       CREATION DATE:  21-June-1979
;    49          0049  1 !
;    50          0050  1 ! MODIFIED BY:
;    51          0051  1 !
;    52          0052  1 !       V01.01          Benn Schreiber          8-Nov-1979
;    53          0053  1 !               Add routine getfilnamdesc
;    54          0054  1 !--
;    55          0055  1
;    56          0056  1
```

```
;    58          0057  1 LIBRARY
;    59          0058  1         'SYS$LIBRARY:STARLET';
;    60          0059  1 REQUIRE
;    61          0060  1         'PREFIX';
;    62          0244  1
;    63          0245  1 EXTERNAL ROUTINE
;    64          0246  1         lib$get_vm : ADDRESSING_MODE (GENERAL), !Allocate virtual memory
;    65          0247  1         lib$free_vm : ADDRESSING_MODE (GENERAL);!Deallocate virtual memory
```

EXE

```
Mod
---
LIB
LIB
LIB
LIB

OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
OTS
STR
STR

STR
STR
STR
STR

STR
STR
STR
STR

STR
STR
STR
STR
STR
STR

STR
STR
```

```
;    67          0248  1 GLOBAL ROUTINE lib_get_mem (bytes, retadr) =
;    68          0249  1 !
;    69          0250  1 ! allocate virtual memory
;    70          0251  1 !
;    71          0252  1 lib$get_vm (bytes, .retadr);

                                                        .TITLE  LIB_SUBS
                                                        .IDENT  \V04-000\

                                                        .EXTRN  LIB$GET_VM, LIB$FREE_VM

                                                        .PSECT  $CODE$,NOWRT,2

                                          0000 00000    .ENTRY  LIB_GET_MEM, Save nothing                ; 0248
                                    08    AC  DD 00002  PUSHL   RETADR                                   ; 0252
                                    04    AC  9F 00005  PUSHAB  BYTES                                    ;
                         00000000G  00        02  FB 00008  CALLS   #2, LIB$GET_VM                      ;
                                              04 0000F    RET                                          ;

; Routine Size:  16 bytes,    Routine Base:  $CODE$ + 0000

;    72          0253  1 GLOBAL ROUTINE lib_free_mem (bytes, address) =
;    73          0254  1 !
;    74          0255  1 ! deallocate virtual memory
;    75          0256  1 !
;    76          0257  1 lib$free_vm (bytes, address);

                                          0000 00000    .ENTRY  LIB_FREE_MEM, Save nothing               ; 0253
                                    08    AC  9F 00002  PUSHAB  ADDRESS                                  ; 0257
                                    04    AC  9F 00005  PUSHAB  BYTES                                    ;
                         00000000G  00        02  FB 00008  CALLS   #2, LIB$FREE_VM                     ;
                                              04 0000F    RET                                          ;

; Routine Size:  16 bytes,    Routine Base:  $CODE$ + 0010

;    77          0258  1 GLOBAL ROUTINE lib_get_zmem (bytes, retadr) =
;    78          0259  2 BEGIN
;    79          0260  2 !
;    80          0261  2 ! Get zeroed memory
;    81          0262  2 !
;    82          0263  2 perform (lib_get_mem (.bytes, .retadr));
;    83          0264  2 CH$FILL (0, .bytes, ..retadr);
;    84          0265  2 RETURN true
;    85          0266  1 END;

                                          003C 00000    .ENTRY  LIB_GET_ZMEM, Save R2,R3,R4,R5           ; 0258
```

```
                        7E      04      AC  7D 00002           MOVQ    BYTES, -(SP)                            ; 0263
                    D6  AF              02  FB 00006           CALLS   #2, LIB_GET_MEM                         ;
                        0E              50  E9 0000A           BLBC    STATUS, 1$                              ;
                        50      08      BC  D0 0000D           MOVL    @RETADR, R0                             ; 0264
    04  AC                  00          00  2C 00011           MOVC5   #0, (SP), #0, BYTES, (R0)               ;
                                        60      00017                                                       ;
                        50              01  D0 00018           MOVL    #1, R0                                  ; 0265
                                            04 0001B 1$:       RET                                             ; 0266

; Routine Size:  28 bytes,    Routine Base:  $CODE$ + 0020
```

```
;    87          0267  1 GLOBAL ROUTINE find_list_width (fab, listingwidth) =
;    88          0268  2 BEGIN
;    89          0269  2 !
;    90          0270  2 ! Determine the width of the listing line
;    91          0271  2 ! FAB is the fab of the open file, width returned in listingwidth.
;    92          0272  2 !
;    93          0273  2 MAP
;    94          0274  2     fab : REF BBLOCK;
;    95          0275  2
;    96          0276  2 BIND
;    97          0277  2     namblk = .fab [fab$l_nam] : BBLOCK;          !NAM block
;    98          0278  2
;    99          0279  2 LOCAL
;   100          0280  2     devnamdesc : BBLOCK [dsc$c_s_bln],
;   101          0281  2     devnambuf : VECTOR [nam$c_maxrss, BYTE],
;   102          0282  2     devnambufdesc : BBLOCK [dsc$c_s_bln],
;   103          0283  2     devinfobuf : BBLOCK [dib$k_length],
;   104          0284  2     devinfodesc : BBLOCK [dsc$c_s_bln],
;   105          0285  2     devchan;
;   106          0286  2
;   107          0287  2 LITERAL
;   108          0288  2     ch_escape = %X '1B';                         !ASCII <ESC>
;   109          0289  2 .listingwidth = 80;                               !Assume default of 80
;   110          0290  2 IF (.fab [fab$l_dev] AND dev$m_spl) NEQ 0         !If device is spooled
;   111          0291  3 THEN BEGIN
;   112          0292  3     devnamdesc [dsc$w_length] = CH$FIND_CH (.namblk [nam$b_esl],
;   113          0293  3                                 .namblk [nam$l_esa], %ASCII ':')
;   114          0294  3                                 - .namblk [nam$l_esa];
;   115          0295  3     devnamdesc [dsc$a_pointer] = .namblk [nam$l_esa];
;   116          0296  3     END
;   117          0297  3 ELSE BEGIN                                        !Device is not spooled
;   118          0298  3     devnamdesc [dsc$w_length] = CH$FIND_CH (.namblk [nam$b_rsl],
;   119          0299  3                                 .namblk [nam$l_rsa], %ASCII ':')
;   120          0300  3                                 - .namblk [nam$l_rsa];
;   121          0301  3     devnamdesc [dsc$a_pointer] = .namblk [nam$l_esa];
;   122          0302  2     END;
;   123          0303  2 devnambufdesc [dsc$w_length] = nam$c_maxrss;
;   124          0304  2 devnambufdesc [dsc$a_pointer] = devnambuf;
;   125          0305  2 $TRNLOG (LOGNAM = devnamdesc, RSLLEN = devnambufdesc, RSLBUF = devnambufdesc);
;   126          0306  2 IF .devnambuf [0] EQL ch_escape                   !Check for process permanent file
;   127          0307  3 THEN BEGIN
;   128          0308  3     devnambufdesc [dsc$w_length] = .devnambufdesc [dsc$w_length] - 4;
;   129          0309  3     devnambufdesc [dsc$a_pointer] = .devnambufdesc [dsc$a_pointer] + 4;
;   130          0310  2     END;
;   131          0311  2 !
;   132          0312  2 ! Assign the device and then do a $GETCHN to get the width
;   133          0313  2 !
;   134          0314  3 IF $ASSIGN (DEVNAM = devnambufdesc, CHAN = devchan)
;   135          0315  3 THEN BEGIN
;   136          0316  3     devinfodesc [dsc$w_length] = dib$k_length;   !Set up descriptor for $GETCHN
;   137          0317  3     devinfodesc [dsc$a_pointer] = devinfobuf;
;   138          0318  4     IF $GETCHN (CHAN = .devchan, SCDBUF = devinfodesc)
;   139          0319  3         THEN .listingwidth = .devinfobuf [dib$w_devbufsiz];
;   140          0320  3     $DASSGN (CHAN = .devchan);                   !Deassign channel
;   141          0321  2     END;
;   142          0322  2 RETURN true;
;   143          0323  1 END;                                             !Of find_list_width
```

```
                                                        .EXTRN  SYS$TRNLOG, SYS$ASSIGN
                                                        .EXTRN  SYS$GETCHN, SYS$DASSGN

                                        0004 00000      .ENTRY  FIND_LIST_WIDTH, Save R2                        ; 0267
                          5E      FE70  CE   9E 00002   MOVAB   -400(SP), SP
                          50        04  AC   D0 00007   MOVL    FAB, R0                                         ; 0277
                          52        28  A0   D0 0000B   MOVL    40(R0), R2
                       08 BC        50  8F   9A 0000F   MOVZBL  #80, @LISTINGWIDTH                             ; 0289
                    15    40 A0       06  E1 00014   BBC     #6, 64(R0), 2$                                  ; 0290
                          50        0B  A2   9A 00019   MOVZBL  11(R2), R0                                      ; 0292
                 0C B2    50            3A   3A 0001D   LOCC    #58, R0, @12(R2)
                                        02   12 00022   BNEQ    1$
                                        51   D4 00024   CLRL    R1
                 F8 AD    51        0C  A2   A3 00026 1$:   SUBW3   12(R2), R1, DEVNAMDESC                    ; 0294
                                        13   11 0002C   BRB     4$                                              ; 0295
                          50        03  A2   9A 0002E 2$:   MOVZBL  3(R2), R0                                   ; 0298
                 04 B2    50            3A   3A 00032   LOCC    #58, R0, @4(R2)
                                        02   12 00037   BNEQ    3$
                                        51   D4 00039   CLRL    R1
                 F8 AD    51        04  A2   A3 0003B 3$:   SUBW3   4(R2), R1, DEVNAMDESC                     ; 0300
                       FC AD        0C  A2   D0 00041 4$:   MOVL    12(R2), DEVNAMDESC+4                    ; 0301
                  0080 CE          FF  8F   9B 00046   MOVZBW  #255, DEVNAMBUFDESC                           ; 0303
                  0084 CE        0088  CE   9E 0004C   MOVAB   DEVNAMBUF, DEVNAMBUFDESC+4                    ; 0304
                                        7E   7C 00053   CLRQ    -(SP)                                           ; 0305
                                        7E   D4 00055   CLRL    -(SP)
                                  008C  CE   9F 00057   PUSHAB  DEVNAMBUFDESC
                                  0090  CE   9F 0005B   PUSHAB  DEVNAMBUFDESC
                                    F8  AD   9F 0005F   PUSHAB  DEVNAMDESC
           00000000G   00                   06   FB 00062   CALLS   #6, SYS$TRNLOG
                          1B        0088  CE   91 00069   CMPB    DEVNAMBUF, #27                              ; 0306
                                        0A   12 0006E   BNEQ    5$
                0080 CE                 04   A2 00070   SUBW2   #4, DEVNAMBUFDESC                            ; 0308
                0084 CE                 04   C0 00075   ADDL2   #4, DEVNAMBUFDESC+4                          ; 0309
                                        7E   7C 0007A 5$:   CLRQ    -(SP)                                   ; 0314
                                    08  AE   9F 0007C   PUSHAB  DEVCHAN
                                  008C  CE   9F 0007F   PUSHAB  DEVNAMBUFDESC
           00000000G   00                   04   FB 00083   CALLS   #4, SYS$ASSIGN
                          2C                50   E9 0008A   BLBC    R0, 7$
                    04    AE        74  8F   9B 0008D   MOVZBW  #116, DEVINFODESC                            ; 0316
                    08    AE        0C  AE   9E 00092   MOVAB   DEVINFOBUF, DEVINFODESC+4                    ; 0317
                                    04  AE   9F 00097   PUSHAB  DEVINFODESC                                   ; 0318
                                        7E   7C 0009A   CLRQ    -(SP)
                                        7E   D4 0009C   CLRL    -(SP)
                                    10  AE   DD 0009E   PUSHL   DEVCHAN
           00000000G   00                   05   FB 000A1   CALLS   #5, SYS$GETCHN
                          05                50   E9 000A8   BLBC    R0, 6$
                    08    BC        12  AE   3C 000AB   MOVZWL  DEVINFOBUF+6, @LISTINGWIDTH                   ; 0319
                                        6E   DD 000B0 6$:   PUSHL   DEVCHAN                                     ; 0320
           00000000G   00                   01   FB 000B2   CALLS   #1, SYS$DASSGN
                          50                01   D0 000B9 7$:   MOVL    #1, R0                               ; 0322
                                            04   000BC       RET                                             ; 0323

; Routine Size:  189 bytes,    Routine Base:  $CODE$ + 003C
```

_$

Ps

_L

_L

_L

_O

_S

```
;  145          0324  1 GLOBAL ROUTINE getfilnamdesc (fab, filedesc) =
;  146          0325  2 BEGIN
;  147          0326  2 !++
;  148          0327  2 ! FUNCTIONAL DESCRIPTION:
;  149          0328  2 !
;  150          0329  2 !       This routine returns a string descriptor for a file.
;  151          0330  2 !
;  152          0331  2 ! Inputs:
;  153          0332  2 !
;  154          0333  2 !       fab             Address of the fab
;  155          0334  2 !       filedesc        Address of string descriptor to store into
;  156          0335  2 !
;  157          0336  2 ! Outputs:
;  158          0337  2 !
;  159          0338  2 !       filedesc is filled in.  The strings tried are:
;  160          0339  2 !
;  161          0340  2 !               1) resultant name string
;  162          0341  2 !               2) expanded name string
;  163          0342  2 !               2) filename string
;  164          0343  2 !
;  165          0344  2 !--
;  166          0345  2
;  167          0346  2 MAP
;  168          0347  2     fab : REF BBLOCK,
;  169          0348  2     filedesc : REF BBLOCK;
;  170          0349  2
;  171          0350  2 BIND
;  172          0351  2     nam = .fab [fab$l_nam] : BBLOCK;
;  173          0352  2
;  174          0353  2 LOCAL
;  175          0354  2     nameptr;
;  176          0355  2
;  177          0356  2 IF (filedesc [dsc$w_length] = .nam [nam$b_rsl]) NEQ 0   !If resultant name present
;  178          0357  2 THEN filedesc [dsc$a_pointer] = .nam [nam$l_rsa]
;  179          0358  2 ELSE IF (filedesc [dsc$w_length] = .nam [nam$b_esl]) NEQ 0 !If expanded name present
;  180          0359  2     THEN filedesc [dsc$a_pointer] = .nam [nam$l_esa]
;  181          0360  3     ELSE BEGIN
;  182          0361  3         filedesc [dsc$w_length] = .fab [fab$b_fns];     !Use filename string
;  183          0362  3         filedesc [dsc$a_pointer] = .fab [fab$l_fna];    ! if all else fails
;  184          0363  2         END;
;  185          0364  2 !
;  186          0365  2 ! Allocate memory and copy the filename to it
;  187          0366  2 !
;  188          0367  2 lib$get_mem (.filedesc [dsc$w_length], nameptr);
;  189          0368  2 CH$MOVE (.filedesc [dsc$w_length], .filedesc [dsc$a_pointer], .nameptr);
;  190          0369  2 filedesc [dsc$a_pointer] = .nameptr;
;  191          0370  2 RETURN true
;  192          0371  1 END;                                                   !Of getfilnamdesc

                                007C 00000          .ENTRY  GETFILNAMDESC, Save R2,R3,R4,R5,R6          ; 0324
                     5E           04  C2 00002      SUBL2   #4, SP                                      ;
                     51      04   AC  7D 00005      MOVQ    FAB, R1                                     ; 0351
                     50      28   A1  D0 00009      MOVL    40(R1), R0                                  ;
```

```
                                56      04    A2  9E 0000D          MOVAB   4(R2), R6                                ; 0357
                                62      03    A0  9B 00011          MOVZBW  3(R0), (R2)                             ; 0356
                                              06  13 00015          BEQL    1$                                      ;
                                66      04    A0  D0 00017          MOVL    4(R0), (R6)                             ; 0357
                                              14  11 0001B          BRB     3$                                      ;
                                62      0B    A0  9B 0001D 1$:      MOVZBW  11(R0), (R2)                            ; 0358
                                              06  13 00021          BEQL    2$                                      ;
                                66      0C    A0  D0 00023          MOVL    12(R0), (R6)                            ; 0359
                                              08  11 00027          BRB     3$                                      ;
                                62      34    A1  9B 00029 2$:      MOVZBW  52(R1), (R2)                            ; 0361
                                66      2C    A1  D0 0002D          MOVL    44(R1), (R6)                            ; 0362
                                              5E  DD 00031 3$:      PUSHL   SP                                      ; 0367
                                7E            62  3C 00033          MOVZWL  (R2), -(SP)                             ;
                          FECC  CF            02  FB 00036          CALLS   #2, LIB_GET_MEM                         ;
                00    BE    00  B6            62  28 0003B          MOVC3   (R2), @0(R6), @NAMEPTR                  ; 0368
                                66            6E  D0 00041          MOVL    NAMEPTR, (R6)                           ; 0369
                                50            01  D0 00044          MOVL    #1, R0                                  ; 0370
                                                  04 00047          RET                                             ; 0371

; Routine Size:  72 bytes,    Routine Base:  $CODE$ + 00F9

;  193          0372  1 END
;  194          0373  0 ELUDOM


;                                   PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  $CODE$                         321  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                               Library Statistics

;                                                 -------- Symbols --------      Pages       Processing
;       File                                      Total    Loaded   Percent      Mapped      Time

;  _$255$DUA28:[SYSLIB]STARLET.L32;1               9776        37         0       581          00:01.0


;                                   COMMAND QUALIFIERS

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

; Size:         321 code + 0 data bytes
```

```
; Run Time:          00:08.7
; Elapsed Time:      00:18.3
; Lines/CPU Min:      2566
; Lexemes/CPU-Min: 28176
; Memory Used:  86 pages
; Compilation Complete
```