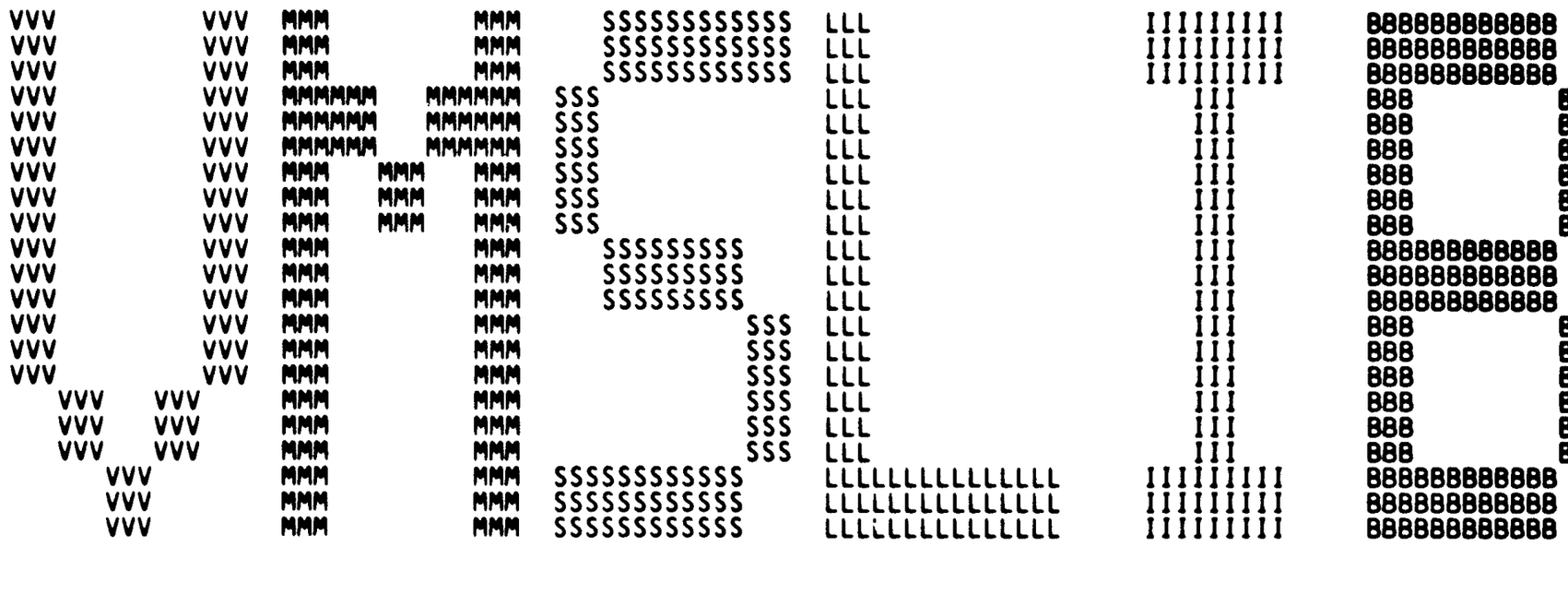
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB
VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB

FILEID**REPORTIOE

```
RRRRRRRR    EEEEEEEEEE  PPPPPPPP      000000    RRRRRRRR    TTTTTTTTTT    IIIIII      000000    EEEEEEEEEE
RRRRRRRR    EEEEEEEEEE  PPPPPPPP      000000    RRRRRRRR    TTTTTTTTTT    IIIIII      000000    EEEEEEEEEE
RR      RR  EE          PP      PP  00      00  RR      RR      TT        II      00      00  EE
RR      RR  EE          PP      PP  00      00  RR      RR      TT        II      00      00  EE
RR      RR  EE          PP      PP  00      00  RR      RR      TT        II      00      00  EE
RR      RR  EE          PP      PP  00      00  RR      RR      TT        II      00      00  EE
RRRRRRRR    EEEEEEEE    PPPPPPPP    00      00  RRRRRRRR        TT        II      00      00  EEEEEEEE
RRRRRRRR    EEEEEEEE    PPPPPPPP    00      00  RRRRRRRR        TT        II      00      00  EEEEEEEE
RR  RR      EE          PP          00      00  RR  RR          TT        II      00      00  EE
RR  RR      EE          PP          00      00  RR  RR          TT        II      00      00  EE
RR    RR    EE          PP          00      00  RR    RR        TT        II      00      00  EE              ....
RR    RR    EE          PP          00      00  RR    RR        TT        II      00      00  EE              ....
RR      RR  EEEEEEEEEE  PP            000000    RR      RR      TT      IIIIII      000000    EEEEEEEEEE      ....
RR      RR  EEEEEEEEEE  PP            000000    RR      RR      TT      IIIIII      000000    EEEEEEEEEE      ....

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

```
;     1          0001  0 MODULE util$report_io_err(
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 ADDRESSING_MODE(EXTERNAL=GENERAL,NONEXTERNAL=GENERAL),
;     4          0004  0                 IDENT = 'V04-000'
;     5          0005  0                 ) =
;     6          0006  1 BEGIN
;     7          0007  1 %TITLE  'Report I/O error on FAB or RAB';
;     8          0008  1 !
;     9          0009  1 !*****************************************************************************
;    10          0010  1 !*                                                                           *
;    11          0011  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    12          0012  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    13          0013  1 !*  ALL RIGHTS RESERVED.                                                      *
;    14          0014  1 !*                                                                           *
;    15          0015  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    16          0016  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    17          0017  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    18          0018  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    19          0019  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    20          0020  1 !*  TRANSFERRED.                                                              *
;    21          0021  1 !*                                                                           *
;    22          0022  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    23          0023  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    24          0024  1 !*  CORPORATION.                                                              *
;    25          0025  1 !*                                                                           *
;    26          0026  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    27          0027  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    28          0028  1 !*                                                                           *
;    29          0029  1 !*                                                                           *
;    30          0030  1 !*****************************************************************************
;    31          0031  1
;    32          0032  1 !++
;    33          0033  1 !
;    34          0034  1 ! FACILITY:  Run time library
;    35          0035  1 !
;    36          0036  1 ! ABSTRACT:
;    37          0037  1 !
;    38          0038  1 !       Signal an I/O error on either the FAB or the RAB
;    39          0039  1 !
;    40          0040  1 ! ENVIRONMENT:
;    41          0041  1 !
;    42          0042  1 !       VAX native, user mode.
;    43          0043  1 !
;    44          0044  1 !--
;    45          0045  1 !
;    46          0046  1 !
;    47          0047  1 ! AUTHOR:  Benn Schreiber
;    48          0048  1 !
;    49          0049  1 ! CREATION DATE: 7-Feb-1983
;    50          0050  1 !
;    51          0051  1 ! MODIFIED BY:
;    52          0052  1 !
;    53          0053  1 !--
```

```
;    55          0054  1 %SBTTL  'Declarations';
;    56          0055  1
;    57          0056  1 LIBRARY
;    58          0057  1         'SYS$LIBRARY:STARLET';
;    59          0058  1 !
;    60          0059  1 ! Define UTIL$ psects
;    61          0060  1 !
;    62          0061  1 PSECT
;    63          0062  1     CODE = _UTIL$CODE,
;    64          0063  1     GLOBAL = _UTIL$DATA,
;    65          0064  1     OWN = _UTIL$DATA,
;    66          0065  1     PLIT = _UTIL$CODE;
```

```
;    68          0066  1 %SBTTL  'util$getfilename - Get descriptor of file spec from FAB';
;    69          0067  1 GLOBAL ROUTINE util$getfilename (fab) =
;    70          0068  2 BEGIN
;    71          0069  2 !++
;    72          0070  2 ! FUNCTIONAL DESCRIPTION:
;    73          0071  2 !
;    74          0072  2 !       This routine returns a string descriptor for a file.
;    75          0073  2 !
;    76          0074  2 ! Inputs:
;    77          0075  2 !
;    78          0076  2 !       fab             Address of the fab
;    79          0077  2 !
;    80          0078  2 ! Outputs:
;    81          0079  2 !
;    82          0080  2 !       Routine value is address of string descriptor for file name
;    83          0081  2 !
;    84          0082  2 !--
;    85          0083  2
;    86          0084  2 MAP
;    87          0085  2     fab : REF $BBLOCK;
;    88          0086  2
;    89          0087  2 LOCAL
;    90          0088  2     nam : REF $BBLOCK;
;    91          0089  2
;    92          0090  2 OWN
;    93          0091  2     filedesc : $BBLOCK[dsc$c_s_bln];
;    94          0092  2
;    95          0093  2 nam = .fab[fab$l_nam];
;    96          0094  3 IF (.nam EQL 0)
;    97          0095  3     OR (IF (filedesc [dsc$w_length] = .nam [nam$b_rsl]) NEQ 0
;    98          0096  3             THEN filedesc [dsc$a_pointer] = .nam [nam$l_rsa]
;    99          0097  3             ELSE IF (filedesc [dsc$w_length] = .nam [nam$b_esl]) NEQ 0
;   100          0098  3                 THEN filedesc [dsc$a_pointer] = .nam [nam$l_esa];
;   101          0099  3         .filedesc[dsc$w_length] EQL 0)
;   102          0100  3 THEN BEGIN
;   103          0101  3     filedesc [dsc$w_length] = .fab [fab$b_fns]; !Use filename string
;   104          0102  3     filedesc [dsc$a_pointer] = .fab [fab$l_fna];! if all else fails
;   105          0103  2     END;
;   106          0104  2
;   107          0105  2 RETURN filedesc
;   108          0106  1 END;                                    !Of util$getfilename


                                                .TITLE  UTIL$REPORT_IO_ERR Report I/O error on FAB or R
                                                                           AB
;
                                                .IDENT  \V04-000\
                                                .PSECT  _UTIL$DATA,NOEXE,2
                                        00000 FILEDESC:
                                                .BLKB   8


                                                .PSECT  _UTIL$CODE,NOWRT,2                              ;
                                    0004 00000          .ENTRY  UTIL$GETFILENAME, Save R2                   ; 0067
```

```
                                52 00000000'  00  9E 00002          MOVAB   FILEDESC, R2                          ;
                                51        04  AC  D0 00009          MOVL    FAB, R1                             ; 0093
                                50        28  A1  D0 0000D          MOVL    40(R1), NAM                         ;
                                                1C  13 00011          BEQL    3$                                  ; 0094
                                62        03  A0  9B 00013          MOVZBW  3(NAM), FILEDESC                    ; 0095
                                                07  13 00017          BEQL    1$                                  ;
                             04 A2        04  A0  D0 00019          MOVL    4(NAM), FILEDESC+4                  ; 0096
                                                0B  11 0001E          BRB     2$                                  ;
                                62        0B  A0  9B 00020  1$:     MOVZBW  11(NAM), FILEDESC                   ; 0097
                                                05  13 00024          BEQL    2$                                  ;
                             04 A2        0C  A0  D0 00026          MOVL    12(NAM), FILEDESC+4                 ; 0098
                                                62  B5 0002B  2$:     TSTW    FILEDESC                            ; 0099
                                                09  12 0002D          BNEQ    4$                                  ;
                                62        34  A1  9B 0002F  3$:     MOVZBW  52(R1), FILEDESC                    ; 0101
                             04 A2        2C  A1  D0 00033          MOVL    44(R1), FILEDESC+4                  ; 0102
                                50            62  9E 00038  4$:     MOVAB   FILEDESC, R0                        ; 0105
                                                04  0003B          RET                                         ; 0106

; Routine Size:  60 bytes,    Routine Base:  _UTIL$CODE + 0000
```

```
;  110          0107  1 %SBTTL  'util$report_io_error - Report I/O error on FAB or RAB';
;  111          0108  1 GLOBAL ROUTINE util$report_io_error (frab) =
;  112          0109  2 BEGIN
;  113          0110  2 !+++
;  114          0111  2 ! FUNCTIONAL DESCRIPTION:
;  115          0112  2 !
;  116          0113  2 !       This routine signals an I/O error.
;  117          0114  2 !
;  118          0115  2 ! Inputs:
;  119          0116  2 !
;  120          0117  2 !       frab            The FAB or the RAB which got the error
;  121          0118  2 !                       The $L_CTX field of the FAB/RAB must contain the
;  122          0119  2 !                       error code to signal
;  123          0120  2 !                        (SHR$_OPENIN/OPENOUT/READERR/WRITEERR/CLOSEIN/CLOSEOUT)
;  124          0121  2 !                       If frab is a RAB, then RAB$L_FAB must point to the FAB
;  125          0122  2 !                       In either case, the FAB must point to a valid NAM block
;  126          0123  2 !                        with both the expanded and resultant name strings in
;  127          0124  2 !                        order for consistent error reporting
;  128          0125  2 !
;  129          0126  2 ! Outputs:
;  130          0127  2 !
;  131          0128  2 !       The error is signalled.  RMS$_EOF is not signalled
;  132          0129  2 !
;  133          0130  2 ! Routine value:
;  134          0131  2 !
;  135          0132  2 !       The $L_STS field of frab is returned
;  136          0133  2 !
;  137          0134  2 !---
;  138          0135  2
;  139          0136  2 MAP
;  140          0137  2     frab : REF $BBLOCK;
;  141          0138  2
;  142          0139  2 IF .frab[rab$l_sts] NEQ rms$_eof
;  143          0140  2     THEN SIGNAL((.frab[fab$l_ctx],1,
;  144          0141  3         util$getfilename((IF .frab[fab$b_bid] EQL fab$c_bid
;  145          0142  3                          THEN .frab
;  146          0143  2                          ELSE .frab[rab$l_fab])),
;  147          0144  2          .frab[fab$l_sts],.frab[fab$l_stv]);
;  148          0145  2
;  149          0146  2 RETURN .frab[fab$l_sts]
;  150          0147  1 END;


                                        0004 00000          .ENTRY  UTIL$REPORT_IO_ERROR, Save R2                   ; 0108
                        52      04  AC  D0 00002          MOVL    FRAB, R2                                        ; 0139
            0001827A    8F      08  A2  D1 00006          CMPL    8(R2), #98938                                   ;
                                    22  13 0000E          BEQL    3$                                              ;
                        7E      08  A2  7D 00010          MOVQ    8(R2), -(SP)                                    ; 0144
                        03          62  91 00014          CMPB    (R2), #3                                        ; 0141
                                    04  12 00017          BNEQ    1$                                              ;
                                    52  DD 00019          PUSHL   R2                                              ; 0142
                                    03  11 0001B          BRB     2$                                              ;
                                3C  A2  DD 0001D 1$:      PUSHL   60(R2)                                          ; 0143
                  A0    AF              01  FB 00020 2$:  CALLS   #1, UTIL$GETFILENAME                            ; 0141
```

```
                                        50  DD 00024          PUSHL   R0                                      ;  0140
                                        01  DD 00026          PUSHL   #1                                      ;
                                    18  A2  DD 00028          PUSHL   24(R2)                                  ;
                        00000000G   00      05  FB 0002B      CALLS   #5, LIB$SIGNAL                          ;
                                    50  08  A2  D0 00032 3$:   MOVL    8(R2), R0                               ;  0146
                                            04 00036          RET                                             ;  0147

; Routine Size:  55 bytes,    Routine Base:  _UTIL$CODE + 003C

;  151          0148  0 END ELUDOM



                                                              .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  _UTIL$DATA                      8  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  _UTIL$CODE                    115  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        19         0         581        00:01.0



;                                   COMMAND QUALIFIERS

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

; Size:          115 code + 8 data bytes
; Run Time:        00:04.2
; Elapsed Time:    00:05.3
; Lines/CPU Min:    2119
; Lexemes/CPU-Min: 13890
; Memory Used:  47 pages
; Compilation Complete
```

READOBJ
LIS

SCREEN
LIS

LIBEXECLI
LIS

MOUNTMSG
LIS

PASMSG
LIS

LIBFILPRO
LIS

LIBNETCON
LIS

LIBEXTCON
LIS

LIBMERGE
LIS

LNKMSG
LIS

REPORTIOE
LIS

LIBUNLFIL
LIS

LIBFIDNAM
LIS

MATCHNAME
LIS

OPCMSG
LIS

PLIMSG
LIS