```
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```


LL IIIIII BBBBBBBB PPPPPPPP UU UU TTTTTTTTT 000000 UU UU TTTTTTTTTT
LL IIIIII BBBBBBBB PPPPPPPP UU UU TTTTTTTTTT 000000 UU UU TTTTTTTTTT
LL II BB BB PP PP UU UU TT 00 00 UU UU TT
LL II BB BB PP PP UU UU TT 00 00 UU UU TT
LL II BB BB PP PP UU UU TT 00 00 UU UU TT
LL II BB BB PP PP UU UU TT 00 00 UU UU TT
LL II BBBBBBBB PPPPPPPP UU UU TT 00 00 UU UU TT
LL II BBBBBBBB PPPPPPPP UU UU TT 00 00 UU UU TT
LL II BB BB PP UU UU TT 00 00 UU UU TT
LL II BB BB PP UU UU TT 00 00 UU UU TT
LL II BB BB PP UU UU TT 00 00 UU UU TT
LL II BB BB PP UU UU TT 00 00 UU UU TT
LLLLLLLLLL IIIIII BBBBBBBB PP UUUUUUUUUU TT 000000 UUUUUUUUUU TT
LLLLLLLLLL IIIIII BBBBBBBB PP UUUUUUUUUU TT 000000 UUUUUUUUUU TT
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 LIB$PUT_OUTPUT (%TITLE'Library $PUT on device SYS$OUTPUT'
;     2          0002  0         IDENT = '1-006' ! File: LIBPUTOUT.B32 EDIT: SBL1006
;     3          0003  0                       ) =
;     4          0004  1 BEGIN
;     5          0005  1
;     6          0006  1 !
;     7          0007  1 !*****************************************************************************
;     8          0008  1 !*                                                                           *
;     9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    11          0011  1 !*  ALL RIGHTS RESERVED.                                                      *
;    12          0012  1 !*                                                                           *
;    13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    18          0018  1 !*  TRANSFERRED.                                                              *
;    19          0019  1 !*                                                                           *
;    20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    22          0022  1 !*  CORPORATION.                                                              *
;    23          0023  1 !*                                                                           *
;    24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    26          0026  1 !*                                                                           *
;    27          0027  1 !*                                                                           *
;    28          0028  1 !*****************************************************************************
;    29          0029  1 !
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 ! FACILITY:  General Uitlity Library
;    33          0033  1 !
;    34          0034  1 ! ABSTRACT:
;    35          0035  1 !
;    36          0036  1 !       Output a string as a record on device SYS$OUTPUT.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:  User Mode - AST re-entrant
;    39          0039  1 !
;    40          0040  1 ! AUTHOR:  Thomas N. Hastings, CREATION DATE:  8-Aug-1977
;    41          0041  1 !
;    42          0042  1 ! MODIFIED BY:
;    43          0043  1 !
;    44          0044  1 !       Thomas N. Hastings, 8-Aug-1977: VERSION 0
;    45          0045  1 ! 01    - original
;    46          0046  1 ! 04    - change to SYS$OUTPUT
;    47          0047  1 ! 05    - change to do OPEN at first time
;    48          0048  1 ! 06    - change to set up RAB for message
;    49          0049  1 ! 0-7   - fix comment
;    50          0050  1 ! 0-9   - Put in carriage control.  TNH 28-Oct-77
;    51          0051  1 ! 0-11  - Change to STARLET library.  DGP 20-Apr-78
;    52          0052  1 ! 0-12  - Change REQUIRE files for VAX system build.  DGP 28-Apr-78
;    53          0053  1 ! 0-13  - Change STARLET to RTLSTARLE to avoid conflicts.  DGP 1-May-78
;    54          0054  1 ! 0-14  - Make wait if stream active, so AST re-entrant.  TNH 29-July-78
;    55          0055  1 ! 0-15  - Change file name to LIBPUTOUT.B32, and change the name of
;    56          0056  1 !           the REQUIRE file similarly.  JBS 14-NOV-78
;    57          0057  1 ! 1-001 - Update version number and copyright notice.  JBS 16-NOV-78
```

```
;    58          0058  1 ! 1-002 - Change REQUIRE file names from FOR... to OTS...  JBS 07-DEC-78
;    59          0059  1 ! 1-003 - Enhance to recognize additional classes of string descriptors
;    60          0060  1 !         by invoking LIB$ANALYZE_SDESC_R3 to extract length and
;    61          0061  1 !         address of 1st data byte from descriptor.
;    62          0062  1 !         Remove reference to OTSMAC.REQ.  RKR 27-MAY-1981.
;    63          0063  1 ! 1-004 - Add special-case code to process string descriptors that
;    64          0064  1 !         "read" like fixed string descriptors.  RKR 7-OCT-1981.
;    65          0065  1 ! 1-005 - Redirect jsb's from LIB$ANALYZE_SDESC_R3 to
;    66          0066  1 !         LIB$ANALYZE_SDESC_R2.  RKR 18-NOV-1981
;    67          0067  1 ! 1-006 - Use prologue file.  SBL 24-June-1983
;    68          0068  1 !--
```

```
;    70          0069  1 !
;    71          0070  1 ! PROLOGUE FILE:
;    72          0071  1 !
;    73          0072  1
;    74          0073  1 REQUIRE 'RTLIN:LIBPROLOG';              ! LIB$ definitions
;    75          0144  1
;    76          0145  1 !
;    77          0146  1 ! TABLE OF CONTENTS:
;    78          0147  1 !
;    79          0148  1
;    80          0149  1 FORWARD ROUTINE
;    81          0150  1         LIB$PUT_OUTPUT;         ! Output string on device SYS$OUTPUT
;    82          0151  1
;    83          0152  1 !
;    84          0153  1 ! MACROS:
;    85          0154  1 !
;    86          0155  1
;    87          0156  1 !
;    88          0157  1 ! EQUATED SYMBOLS:
;    89          0158  1 !
;    90          0159  1
;    91          0160  1 !
;    92          0161  1 ! OWN STORAGE:
;    93          0162  1 !
;    94          0163  1
;    95          0164  1 OWN
;    96          0165  1     SYS_OUTPUT_ISI: WORD INITIAL (0); ! ISI for SYS$OUTPUT
;    97          0166  1
;    98          0167  1 !
;    99          0168  1 ! EXTERNAL REFERENCES:
;   100          0169  1 !
;   101          0170  1 EXTERNAL ROUTINE
;   102          0171  1     LIB$ANALYZE_SDESC_R2 : LIB$ANALYZE_SDESC_R2$LINKAGE;
;   103          0172  1                                     ! To extract length and address of 1st
;   104          0173  1                                     ! data byte from descriptor.
;   105          0174  1 !
;   106          0175  1
```

```
;  108          0176  1 GLOBAL ROUTINE LIB$PUT_OUTPUT ( ! Output string to SYS$OUTPUT
;  109          0177  1
;  110          0178  1         MESSAGE          ! Adr. of string descriptor
;  111          0179  1
;  112          0180  1                              ) = ! Value returned is RMS completion
;  113          0181  1                                  ! code
;  114          0182  1 !++
;  115          0183  1 ! FUNCTIONAL DESCRIPTION:
;  116          0184  1 !
;  117          0185  1 !       Outputs a record on device SYS$OUTPUT using RMS $PUT.
;  118          0186  1 !       On first call, device SYS$OUTPUT is opened
;  119          0187  1 !       (or created if it doesn't exist yet).  Thus the logical
;  120          0188  1 !       name SYS$OUTPUT can be assigned to any file name in order
;  121          0189  1 !       to redirect I/O.
;  122          0190  1 !
;  123          0191  1 ! FORMAL PARAMETERS:
;  124          0192  1 !
;  125          0193  1 !       MESSAGE.rt.dx           Adr. of string descriptor of string
;  126          0194  1 !                               to be output.
;  127          0195  1 !
;  128          0196  1 ! IMPLICIT INPUTS:
;  129          0197  1 !
;  130          0198  1 !       NONE
;  131          0199  1 !
;  132          0200  1 ! IMPLICIT OUTPUTS:
;  133          0201  1 !
;  134          0202  1 !       SYS_OUTPUT_ISI  RMS internal stream id for all but first call
;  135          0203  1 !
;  136          0204  1 ! COMPLETION CODES:
;  137          0205  1 !
;  138          0206  1 !       RMS completion code
;  139          0207  1 !       or LIB$_INVARG if descriptor is bad.
;  140          0208  1 !
;  141          0209  1 ! SIDE EFFECTS:
;  142          0210  1 !
;  143          0211  1 !       Opens (creates if not existent) file SYS$OUTPUT on first call.
;  144          0212  1 !--
;  145          0213  1
;  146          0214  2     BEGIN
;  147          0215  2
;  148          0216  2     LOCAL
;  149          0217  2         RMS_STATUS,                     ! RMS status
;  150          0218  2         FAB: $FAB_DECL,                 ! FAB
;  151          0219  2         RAB: $RAB_DECL;                 ! RAB
;  152          0220  2     MAP
;  153          0221  2         MESSAGE: REF BLOCK [, BYTE];    ! String descriptor
;  154          0222  2
;  155          0223  2     IF .SYS_OUTPUT_ISI EQL 0
;  156          0224  2     THEN
;  157          0225  2
;  158          0226  2         !+
;  159          0227  2         ! First call, initialize FAB
;  160          0228  2         !-
;  161          0229  2
;  162          0230  3         BEGIN
;  163        P 0231  3         $FAB_INIT (
;  164        P 0232  3             FAB = FAB,
```

```
;  165        P 0233  3            FAC = PUT,                      ! file access: PUT
;  166        P 0234  3            FNA = UPLIT ('SYS$OUTPUT'),     ! file name: SYS$output
;  167        P 0235  3            FNS = 10,                       ! file name size: 10 bytes
;  168        P 0236  3            RAT = CR,                       ! carriage control - each
;  169        P 0237  3                                            ! record on separate line
;  170          0238  3            FOP = CIF);                     ! file options: create if file
;  171          0239  3                                            ! not exist
;  172          0240  3
;  173          0241  3        !+
;  174          0242  3        ! Create SYS$OUTPUT, open if exist and position to end-of-file,
;  175          0243  3        ! remember ISI
;  176          0244  3        !-
;  177          0245  3
;  178          0246  3        RMS_STATUS = $CREATE (FAB = FAB);                   ! fab addr : FAB
;  179          0247  3        IF NOT .RMS_STATUS THEN RETURN .RMS_STATUS;         ! if create fail
;  180          0248  3                                                            ! then return
;  181          0249  3                                                            ! with RMS
;  182          0250  3                                                            ! status code
;  183        P 0251  3        $RAB_INIT (
;  184        P 0252  3            FAB = FAB,          ! FAB address
;  185        P 0253  3            RAB = RAB,          ! RAB address
;  186          0254  3            ROP = EOF);         ! position at end-of-file if file exists
;  187          0255  3
;  188          0256  3        RMS_STATUS = $CONNECT (RAB = RAB);  ! connect RAB to the file
;  189          0257  3        IF NOT .RMS_STATUS THEN RETURN .RMS_STATUS;
;  190          0258  3        SYS_OUTPUT_ISI = .RAB[RAB$W_ISI];       ! remember ISI
;  191          0259  3        END
;  192          0260  3
;  193          0261  3
;  194          0262  2    ELSE
;  195          0263  2
;  196          0264  2        !+
;  197          0265  2        ! file already exist, just initialize RAB
;  198          0266  2        ! including internal stream identifier returned from first $OPEN
;  199          0267  2        !-
;  200          0268  3        BEGIN
;  201        P 0269  3        $RAB_INIT (
;  202        P 0270  3            FAB = FAB,          ! FAB address
;  203        P 0271  3            RAB = RAB,          ! RAB address
;  204          0272  3            ROP = EOF);         ! position at end-of-file if file exists
;  205          0273  3        RAB[RAB$W_ISI] = .SYS_OUTPUT_ISI;
;  206          0274  2        END;
;  207          0275  2
;  208          0276  2    !+
;  209          0277  2    ! Setup buffer address and length on first and subsequent $PUTs
;  210          0278  2    ! If descriptor is bad, return status from LIB$ANALYZE_SDESC_R2.
;  211          0279  2    !-
;  212          0280  2
;  213          0281  2        IF .MESSAGE [DSC$B_CLASS] GTRU DSC$K_CLASS_D
;  214          0282  2        THEN                    ! Use generalized extract
;  215          0283  3            BEGIN
;  216          0284  3            LOCAL RET_STATUS ;
;  217          0285  3            RET_STATUS = LIB$ANALYZE_SDESC_R2 ( .MESSAGE ;
;  218          0286  3                                                RAB [RAB$W_RSZ],   ! length
;  219          0287  3                                                RAB [RAB$L_RBF]);  ! address
;  220          0288  3
;  221          0289  3            IF NOT .RET_STATUS THEN RETURN (.RET_STATUS) ;
```

```
;  222          0290  3
;  223          0291  3          END
;  224          0292  3
;  225          0293  2      ELSE                    ! Fetch length and address directly
;  226          0294  2
;  227          0295  3          BEGIN
;  228          0296  3          RAB [RAB$W_RSZ] = .MESSAGE [DSC$W_LENGTH] ;
;  229          0297  3          RAB [RAB$L_RBF] = .MESSAGE [DSC$A_POINTER] ;
;  230          0298  2          END ;
;  231          0299  2      !+
;  232          0300  2      ! Output the string as a single record and return RMS completion status
;  233          0301  2      ! If error and it is RECORD STREAM ACTIVE, wait and try again, thus
;  234          0302  2      ! making routine AST re-entrant.  Return SS$_NORMAL (00000001) if
;  235          0303  2      ! success, rather than LIB$_NORMAL (00010001).
;  236          0304  2      !-
;  237          0305  2
;  238          0306  3      IF NOT $PUT (RAB = RAB)
;  239          0307  2      THEN
;  240          0308  2          WHILE .RAB[RAB$L_STS] EQL RMS$_RSA DO
;  241          0309  3              BEGIN
;  242          0310  3              $WAIT (RAB=RAB);
;  243          0311  3              $PUT (RAB=RAB);
;  244          0312  2              END;
;  245          0313  2
;  246          0314  2      RETURN (IF .RAB[RAB$L_STS] THEN SS$_NORMAL ELSE .RAB[RAB$L_STS]);
;  247          0315  2
;  248          0316  1      END;                        ! End of routine LIB$PUT_OUTPUT


                                                        .TITLE  LIB$PUT_OUTPUT Library $PUT on device SYS$OUTPU
                                                                                                              T
;
                                                        .IDENT  \1-006\
                                                        .PSECT  _LIB$DATA,NOEXE,  PIC,2
                                                0000  00000 SYS_OUTPUT_ISI:
                                                        .WORD   0                                             ;
                                                        .PSECT  _LIB$CODE,NOWRT,  SHR,  PIC,2
        00  00  54  55  50  54  55  4F  24  53  59  53  00000 P.AAA:   .ASCII  \SYS$OUTPUT\<0><0>                ;
                                                        .EXTRN  LIB$ANALYZE_SDESC_R2
                                                        .EXTRN  SYS$CREATE, SYS$CONNECT
                                                        .EXTRN  SYS$PUT, SYS$WAIT
                                                          01FC 00000        .ENTRY  LIB$PUT_OUTPUT, Save R2,R3,R4,R5,R6,R7,R8   ; 0176
                                        58 00000000'  EF   9E 00002         MOVAB   SYS_OUTPUT_ISI, R8                          ;
                                        57 00000000G  00   9E 00009         MOVAB   SYS$PUT, R7                                 ;
                                        5E     FF6C   CE   9E 00010         MOVAB   -148(SP), SP                                ;
                                        56            68   3C 00015         MOVZWL  SYS_OUTPUT_ISI, R6                          ; 0223
                                                      6B   12 00018         BNEQ    3$                                          ;
  0050  8F          00            6E                00   2C 0001A         MOVC5   #0, (SP), #0, #80, $RMS_PTR                 ; 0238
                                             44       AE      00021                                                            ;
                                  44   AE      5003   8F   B0 00023         MOVW    #20483, $RMS_PTR                            ;
                                  48   AE 02000000   8F   D0 00029         MOVL    #33554432, $RMS_PTR+4                       ;
                                  5A   AE            01   90 00031         MOVB    #1, $RMS_PTR+22                             ;
```

```
                        62  AE      0202  8F  B0 00035          MOVW    #514, $RMS_PTR+30                     :
                        70  AE        B6  AF  9E 0003B          MOVAB   P.AAA, $RMS_PTR+44                    :
                        78  AE            0A  90 00040          MOVB    #10, $RMS_PTR+52                      :
                                      44  AE  9F 00044          PUSHAB  FAB                                   : 0246
               00000000G    00            01  FB 00047          CALLS   #1, SYS$CREATE                        :
                            56            50  D0 0004E          MOVL    R0, RMS_STATUS                        :
                            27            56  E9 00051          BLBC    RMS_STATUS, 1$                        : 0247
  0044  8F          00      6E            00  2C 00054          MOVC5   #0, (SP), #0, #68, $RMS_PTR           : 0254
                            6E               0005B                                                            :
                            6E      4401  8F  B0 0005C          MOVW    #17409, $RMS_PTR                      :
                        04  AE      0100  8F  3C 00061          MOVZWL  #256, $RMS_PTR+4                      :
                        3C  AE        44  AE  9E 00067          MOVAB   FAB, $RMS_PTR+60                      :
                                          5E  DD 0006C          PUSHL   SP                                    : 0256
               00000000G    00            01  FB 0006E          CALLS   #1, SYS$CONNECT                       :
                            56            50  D0 00075          MOVL    R0, RMS_STATUS                        :
                            04            56  E8 00078          BLBS    RMS_STATUS, 2$                        : 0257
                            50            56  D0 0007B 1$:      MOVL    RMS_STATUS, R0                        :
                                          04     0007E          RET                                           :
                            68        02  AE  B0 0007F 2$:      MOVW    RAB+2, SYS_OUTPUT_ISI                 : 0258
                                          1C  11 00083          BRB     4$                                    : 0223
  0044  8F          00      6E            00  2C 00085 3$:      MOVC5   #0, (SP), #0, #68, $RMS_PTR           : 0272
                            6E               0008C                                                            :
                            6E      4401  8F  B0 0008D          MOVW    #17409, $RMS_PTR                      :
                        04  AE      0100  8F  3C 00092          MOVZWL  #256, $RMS_PTR+4                      :
                        3C  AE        44  AE  9E 00098          MOVAB   FAB, $RMS_PTR+60                      :
                        02  AE            56  B0 0009D          MOVW    R6, RAB+2                             : 0273
                            53        04  AC  D0 000A1 4$:      MOVL    MESSAGE, R3                           : 0281
                            02        03  A3  91 000A5          CMPB    3(R3), #2                             :
                                          15  1B 000A9          BLEQU   5$                                    :
                            50            53  D0 000AB          MOVL    R3, R0                                : 0286
                                00000000G 00  16 000AE          JSB     LIB$ANALYZE_SDESC_R2                  :
                        22  AE            51  B0 000B4          MOVW    R1, RAB+34                            :
                        28  AE            52  D0 000B8          MOVL    R2, RAB+40                            : 0287
                            0A            50  E8 000BC          BLBS    RET_STATUS, 6$                        : 0289
                                          04     000BF          RET                                           :
                        22  AE            63  B0 000C0 5$:      MOVW    (R3), RAB+34                          : 0296
                        28  AE        04  A3  D0 000C4          MOVL    4(R3), RAB+40                         : 0297
                                          5E  DD 000C9 6$:      PUSHL   SP                                    : 0306
                            67            01  FB 000CB          CALLS   #1, SYS$PUT                           :
                            1A            50  E8 000CE          BLBS    R0, 8$                                :
               000182DA     8F        08  AE  D1 000D1 7$:      CMPL    RAB+8, #99034                         : 0308
                                          10  12 000D9          BNEQ    8$                                    :
                                          5E  DD 000DB          PUSHL   SP                                    : 0310
               00000000G    00            01  FB 000DD          CALLS   #1, SYS$WAIT                          :
                                          5E  DD 000E4          PUSHL   SP                                    : 0311
                            67            01  FB 000E6          CALLS   #1, SYS$PUT                           :
                                          E6  11 000E9          BRB     7$                                    : 0308
                            04        08  AE  E9 000EB 8$:      BLBC    RAB+8, 9$                             : 0314
                            50            01  D0 000EF          MOVL    #1, R0                                :
                                          04     000F2          RET                                           :
                            50        08  AE  D0 000F3 9$:      MOVL    RAB+8, R0                             :
                                          04     000F7          RET                                           : 0316

; Routine Size:  248 bytes,    Routine Base:  _LIB$CODE + 000C

;  249        0317  1 END                               ! End of module LIB$PUT_OUTPUT
```

```
;  250          0318  0 ELUDOM
```

```
;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  _LIB$DATA                       2  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  _LIB$CODE                     260  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
```

```
;                                 Library Statistics
;
;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1      9776        78         0         581       00:00.7
;  _$255$DUA28:[LIBRTL.OBJ]RTLLIB.L32;1     36         1         2           8       00:00.1
```

```
;                                   COMMAND QUALIFIERS

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

; Size:          248 code + 14 data bytes
; Run Time:        00:06.1
; Elapsed Time:    00:28.9
; Lines/CPU Min:    3107
; Lexemes/CPU-Min: 54068
; Memory Used:  118 pages
; Compilation Complete
```

0209 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LIBPOLYG
LIS

LIBREMQHI
LIS

LIBRENAME
LIS

LIBSCANC
LIS

LIBSIGSTO
LIS

LIBROOBJ
LIS

LIBRUNPRO
LIS

LIBSIGNAL
LIS

LIBPUTOUT
LIS

LIBREMQTI
LIS

LIBSIGRET
LIS

LIBSIMTRA
LIS

LIBSCOPY
LIS

LIBPOLYH
LIS

LIBREVERT
LIS