```
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        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      CCCCCCCC  UU      UU    SSSSSSSS  TTTTTTTTTT    OOOOOO    MM      MM
LL              IIIIII    BBBBBBBB      CCCCCCCC  UU      UU    SSSSSSSS  TTTTTTTTTT    OOOOOO    MM      MM
LL                II      BB      BB  CC          UU      UU  SS              TT      OO      OO  MMMM  MMMM
LL                II      BB      BB  CC          UU      UU  SS              TT      OO      OO  MMMM  MMMM
LL                II      BB      BB  CC          UU      UU  SS              TT      OO      OO  MM  MM  MM
LL                II      BB      BB  CC          UU      UU  SS              TT      OO      OO  MM  MM  MM
LL                II      BBBBBBBB    CC          UU      UU    SSSSSS        TT      OO      OO  MM      MM
LL                II      BBBBBBBB    CC          UU      UU    SSSSSS        TT      OO      OO  MM      MM
LL                II      BB      BB  CC          UU      UU          SS      TT      OO      OO  MM      MM
LL                II      BB      BB  CC          UU      UU          SS      TT      OO      OO  MM      MM
LL                II      BB      BB  CC          UU      UU          SS      TT      OO      OO  MM      MM
LL                II      BB      BB  CC          UU      UU          SS      TT      OO      OO  MM      MM
LLLLLLLLLL      IIIIII    BBBBBBBB      CCCCCCCC  UUUUUUUUUU  SSSSSSSS        TT        OOOOOO    MM      MM
LLLLLLLLLL      IIIIII    BBBBBBBB      CCCCCCCC  UUUUUUUUUU  SSSSSSSS        TT        OOOOOO    MM      MM


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$CUSTOMIZE_OUTPUT (%TITLE 'Customize printed output'
;     2          0002  0         IDENT = '2-002'         ! File: LIBCURREN.B32 Edit: DG2002
;     3          0003  0                 ) =
;     4          0004  1 BEGIN
;     5          0005  1 !
;     6          0006  1 !**********************************************************************
;     7          0007  1 !*                                                                    *
;     8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                          *
;     9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.           *
;    10          0010  1 !*  ALL RIGHTS RESERVED.                                           *
;    11          0011  1 !*                                                                    *
;    12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
;    14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    17          0017  1 !*  TRANSFERRED.                                                       *
;    18          0018  1 !*                                                                    *
;    19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    21          0021  1 !*  CORPORATION.                                                       *
;    22          0022  1 !*                                                                    *
;    23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.            *
;    25          0025  1 !*                                                                    *
;    26          0026  1 !*                                                                    *
;    27          0027  1 !**********************************************************************
;    28          0028  1 !
;    29          0029  1
;    30          0030  1 !++
;    31          0031  1 ! FACILITY: General Purpose Library
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT:
;    34          0034  1 !
;    35          0035  1 !       This module contains the procedures LIB$CURRENCY, LIB$DIGIT_SEP
;    36          0036  1 !       and LIB$RADIX_POINT.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT: User mode - AST reentrant
;    39          0039  1 !
;    40          0040  1 ! AUTHOR: John Sauter, CREATION DATE: 18-OCT-1979
;    41          0041  1 !
;    42          0042  1 ! EDIT HISTORY:
;    43          0043  1 !
;    44          0044  1 ! 1-001 - Original.
;    45          0045  1 ! 1-002 - Return signals as status.  JBS 08-JAN-1980
;    46          0046  1 ! 1-003 - Change STR$ codes to LIB$.  JBS 22-JAN-1980
;    47          0047  1 ! 1-004 - Enhance to recognize additional classes of string descriptors
;    48          0048  1 !         by invoking LIB$ANALYZE_SDESC_R3 to extract length and address
;    49          0049  1 !         of 1st data byte from descriptor.
;    50          0050  1 !         Change calls from STR$COPY_DX to LIB$SCOPY_DXDX6.
;    51          0051  1 !                           STR$COPY_R  to LIB$SCOPY_R_DX.
;    52          0052  1 !                           STR$FREE1_DX to LIB$SFREE1_DD.
;    53          0053  1 !         Eliminate need for handler and to convert STR$ statuses to
;    54          0054  1 !         LIB$ statuses.
;    55          0055  1 !         RKR 27-MAY-1981
;    56          0056  1 ! 1-005 - Add special-case code to process strings that "read" like
;    57          0057  1 !         fixed strings.  RKR 7-OCT-1981.
```

```
;   58          0058  1 ;  1-006 - Redirect all jsb's from LIB$ANALYZE_SDESC_R3 to
;   59          0059  1 ;           LIB$ANALYZE_SDESC_R2.  Do all copying with LIB$SCOPY_R_DX6.
;   60          0060  1 ;           RKR 18-NOV-1981.
;   61          0061  1 ;  1-007 - Use $TRNLOG instead of LIB$SYS_TRNLOG.  Other minor improvements.
;   62          0062  1 ;           SBL 8-Oct-1982
;   63          0063  1 ;  2-001 - Rewrite using common inner routine.  SBL 8-Feb-1983
;   64          0064  1 ;  2-002 - Use LNM$C_NAMLENGTH for maximum size of equivalence string.
;   65          0065  1 ;           DG 31-Oct-1983
```

```
;    67          0066  1 !
;    68          0067  1 ! PROLOGUE FILE:
;    69          0068  1 !
;    70          0069  1
;    71          0070  1 REQUIRE 'RTLIN:LIBPROLOG';                      LIB$ definitions
;    72          0141  1
;    73          0142  1 !
;    74          0143  1 ! TABLE OF CONTENTS:
;    75          0144  1 !
;    76          0145  1
;    77          0146  1 FORWARD ROUTINE
;    78          0147  1     LIB$CURRENCY,                               ! Get currency symbol
;    79          0148  1     LIB$DIGIT_SEP,                              ! Get digit separator
;    80          0149  1     LIB$RADIX_POINT,                            ! Get radix point
;    81          0150  1     GET_CUSTOM_SYMBOL;                          ! Inner routine
;    82          0151  1
;    83          0152  1 !
;    84          0153  1 ! MACROS:
;    85          0154  1 !
;    86          0155  1 !       NONE
;    87          0156  1 !
;    88          0157  1 ! EQUATED SYMBOLS:
;    89          0158  1 !
;    90          0159  1 !       NONE
;    91          0160  1 !
;    92          0161  1 ! OWN STORAGE:
;    93          0162  1 !
;    94          0163  1 !       NONE
;    95          0164  1 !
;    96          0165  1 ! EXTERNAL REFERENCES:
;    97          0166  1 !
;    98          0167  1
;    99          0168  1 EXTERNAL ROUTINE
;   100          0169  1     LIB$ANALYZE_SDESC_R2 : LIB$ANALYZE_SDESC_R2$LINKAGE,
;   101          0170  1                                                 ! Get length and address
;   102          0171  1     LIB$SCOPY_R_DX6 : LIB$SCOPY_R_DX6$LINKAGE;  ! Copy a string by reference
;   103          0172  1
```

```
;  105          0173  1 GLOBAL ROUTINE LIB$CURRENCY (%SBTTL 'Get currency symbol'
;  106          0174  1         CURRENCY_STR: REF BLOCK [, BYTE],               . Where to put symbol
;  107          0175  1         OUT_LEN: REF VECTOR [, WORD]                    . Optional length of symbol
;  108          0176  1                         ) =
;  109          0177  1
;  110          0178  1 !++
;  111          0179  1 ! FUNCTIONAL DESCRIPTION:
;  112          0180  1 !
;  113          0181  1 !       This routine returns the system's currency symbol.  It should be
;  114          0182  1 !       used before a number to indicate that the number represents
;  115          0183  1 !       money in the country the system is running in.  This routine
;  116          0184  1 !       works by translating the logical name SYS$CURRENCY.  If the
;  117          0185  1 !       translation fails, this routine returns "$", the US money
;  118          0186  1 !       symbol.
;  119          0187  1 !       If the translation succeeds the text produced is returned.
;  120          0188  1 !       Thus a system manager can define SYS$CURRENCY as a system-wide
;  121          0189  1 !       logical name to provide a default for all users, and an
;  122          0190  1 !       individual user with a special need can define SYS$CURRENCY as
;  123          0191  1 !       a user logical name to override the default.
;  124          0192  1 !
;  125          0193  1 ! CALLING SEQUENCE:
;  126          0194  1 !
;  127          0195  1 !       ret_status.wlc.v = LIB$CURRENCY (currency_str.wt.dx
;  128          0196  1 !                                        [,out_len.ww.r])
;  129          0197  1 !
;  130          0198  1 !
;  131          0199  1 ! FORMAL PARAMETERS:
;  132          0200  1 !
;  133          0201  1 !       currency_str    String to receive the currency string
;  134          0202  1 !       out_len         Optional length of the currency string.
;  135          0203  1 !                       This is valuable if currency_str
;  136          0204  1 !                       is a fixed-length string, since it does not
;  137          0205  1 !                       include the padding.
;  138          0206  1 !
;  139          0207  1 ! IMPLICIT INPUTS:
;  140          0208  1 !
;  141          0209  1 !       System-wide (or user-defined) logical name SYS$CURRENCY
;  142          0210  1 !
;  143          0211  1 ! IMPLICIT OUTPUTS:
;  144          0212  1 !
;  145          0213  1 !       NONE
;  146          0214  1 !
;  147          0215  1 ! CONDITION CODES:
;  148          0216  1 !
;  149          0217  1 !       LIB$_INVSTRDES  Invalid string descriptor
;  150          0218  1 !       LIB$_STRTRU     String truncated to fit into output
;  151          0219  1 !
;  152          0220  1 ! SIDE EFFECTS:
;  153          0221  1 !
;  154          0222  1 !       NONE
;  155          0223  1 !
;  156          0224  1 !--
;  157          0225  1
;  158          0226  2     BEGIN
;  159          0227  2
;  160          0228  2     BUILTIN
;  161          0229  2         ACTUALCOUNT;
```

```
;  162          0230  2
;  163          0231  2      !+
;  164          0232  2      ! Call GET_CUSTOM_SYMBOL to do the work.
;  165          0233  2      !-
;  166          0234  2
;  167          0235  3      RETURN ( GET_CUSTOM_SYMBOL (
;  168          0236  3          CURRENCY_STR [0,0,0,0],
;  169          0237  3          (IF ACTUALCOUNT () GEQU 2 THEN OUT_LEN [0] ELSE 0),
;  170          0238  3          UPLIT BYTE (%ASCIC'SYS$CURRENCY'),
;  171          0239  2          %C'$'));
;  172          0240  2
;  173          0241  1      END;                                ! End of routine LIB$CURRENCY


                                                                .TITLE  LIB$CUSTOMIZE_OUTPUT Customize printed output
                                                                .IDENT  \2-002\

                                                                .PSECT  _LIB$CODE,NOWRT,  SHR,  PIC,2

      59  43  4E  45  52  52  55  43  24  53  59  53  0C  00000 P.AAA:  .ASCII  <12>\SYS$CURRENCY\                          ;

                                                                .EXTRN  LIB$ANALYZE_SDESC_R2
                                                                .EXTRN  LIB$SCOPY_R_DX6

                                          0000 00000            .ENTRY  LIB$CURRENCY, Save nothing                      ; 0173
                                      24  DD 00002            PUSHL   #36                                             ; 0236
                                  EC  AF  9F 00004            PUSHAB  P.AAA                                           ; 0238
                              02      6C  91 00007            CMPB    (AP), #2                                        ; 0237
                                      05  1F 0000A            BLSSU   1$                                              ;
                                  08  AC  DD 0000C            PUSHL   OUT_LEN                                         ;
                                      02  11 0000F            BRB     2$                                              ;
                                      7E  D4 00011 1$:        CLRL    -(SP)                                           ;
                                  04  AC  DD 00013 2$:        PUSHL   CURRENCY_STR                                    ; 0236
                          0000V   CF      04  FB 00016        CALLS   #4, GET_CUSTOM_SYMBOL                           ;
                                          04 0001B            RET                                                     ; 0241

; Routine Size:  28 bytes,    Routine Base:  _LIB$CODE + 000D
```

```
;  175     0242  1 GLOBAL ROUTINE LIB$DIGIT_SEP (%SBTTL 'Get digit separator'
;  176     0243  1         DIGIT_SEP_STR: REF BLOCK [, BYTE],              ! Where to put symbol
;  177     0244  1         OUT_LEN: REF VECTOR [, WORD]                    ! Optional length of symbol
;  178     0245  1                         ) =
;  179     0246  1
;  180     0247  1 !++
;  181     0248  1 ! FUNCTIONAL DESCRIPTION:
;  182     0249  1 !
;  183     0250  1 !       This routine returns the system's digit separator symbol.  It
;  184     0251  1 !       should be used to separate groups of three digits in the
;  185     0252  1 !       integer part of a number, for readability, using the customary
;  186     0253  1 !       symbol.  This routine works by translating the logical name
;  187     0254  1 !       SYS$DIGIT_SEP.  If the translation fails, this routine returns
;  188     0255  1 !       ",", the OS digit separator.  If the translation succeeds the
;  189     0256  1 !       text produced is returned.  Thus a system manager can define
;  190     0257  1 !       SYS$DIGIT_SEP as a system-wide logical name to provide a
;  191     0258  1 !       default for all users, and an individual user with a special
;  192     0259  1 !       need can define SYS$DIGIT_SEP as a user logical name to
;  193     0260  1 !       override the default.
;  194     0261  1 !
;  195     0262  1 ! CALLING SEQUENCE:
;  196     0263  1 !
;  197     0264  1 !       ret_status.wlc.v = LIB$DIGIT_SEP (digit_sep_str.wt.dx
;  198     0265  1 !                                         [,out_len.ww.r])
;  199     0266  1 !
;  200     0267  1 ! FORMAL PARAMETERS:
;  201     0268  1 !
;  202     0269  1 !       digit_sep_str   String to receive the digit_sep string
;  203     0270  1 !       out_len         Optional length of the digit separator string.
;  204     0271  1 !                       This is valuable if digit_sep_str is a
;  205     0272  1 !                       fixed-length string, since it does not include
;  206     0273  1 !                       the padding.
;  207     0274  1 !
;  208     0275  1 ! IMPLICIT INPUTS:
;  209     0276  1 !
;  210     0277  1 !       System-wide (or user-defined) logical name SYS$DIGIT_SEP
;  211     0278  1 !
;  212     0279  1 ! IMPLICIT OUTPUTS:
;  213     0280  1 !
;  214     0281  1 !       NONE
;  215     0282  1 !
;  216     0283  1 ! CONDITION CODES:
;  217     0284  1 !
;  218     0285  1 !       SS$_NORMAL      Normal successful completion
;  219     0286  1 !       LIB$_INVSTRDES  Invalid string descriptor
;  220     0287  1 !       LIB$_STRTRU     Result did not fit in caller's string;
;  221     0288  1 !                       truncated on right
;  222     0289  1 !
;  223     0290  1 ! SIDE EFFECTS:
;  224     0291  1 !
;  225     0292  1 !       NONE
;  226     0293  1 !
;  227     0294  1 !--
;  228     0295  1
;  229     0296  2     BEGIN
;  230     0297  2
;  231     0298  2     BUILTIN
```

```
;  232          0299  2          ACTUALCOUNT;
;  233          0300  2
;  234          0301  2      !+
;  235          0302  2      ! Call GET_CUSTOM_SYMBOL to do the work.
;  236          0303  2      !-
;  237          0304  2
;  238          0305  3      RETURN ( GET_CUSTOM_SYMBOL (
;  239          0306  3          DIGIT_SEP_STR [0,0,0,0],
;  240          0307  3          (IF ACTUALCOUNT () GEQU 2 THEN OUT_LEN [0] ELSE 0),
;  241          0308  3          UPLIT BYTE (%ASCIC'SYS$DIGIT_SEP'),
;  242          0309  2          %C','));
;  243          0310  2
;  244          0311  1      END;                                    ! End of routine LIB$DIGIT_SEP


    50  45  53  5F  54  49  47  49  44  24  53  59  53  0D  00029 P.AAB:  .ASCII  <13>\SYS$DIGIT_SEP\                          ;


                                                  0000 00000          .ENTRY  LIB$DIGIT_SEP, Save nothing                 ; 0242
                                              2C  DD 00002            PUSHL   #44                                         ; 0306
                                          EB  AF  9F 00004            PUSHAB  P.AAB                                       ; 0308
                                    02            6C  91 00007            CMPB    (AP), #2                                ; 0307
                                                  05  1F 0000A            BLSSU   1$                                      ;
                                          08      AC  DD 0000C            PUSHL   OUT_LEN                                 ;
                                                  02  11 0000F            BRB     2$                                      ;
                                                  7E  D4 00011 1$:        CLRL    -(SP)                                   ;
                                          04      AC  DD 00013 2$:        PUSHL   DIGIT_SEP_STR                           ; 0306
                                0000V     CF            04  FB 00016            CALLS   #4, GET_CUSTOM_SYMBOL           ;
                                                        04  0001B            RET                                         ; 0311

; Routine Size:  28 bytes,    Routine Base:  _LIB$CODE + 0037
```

```
;  246          0312  1 GLOBAL ROUTINE LIB$RADIX_POINT (%SBTTL 'Get radix point symbol'
;  247          0313  1         RADIX_POINT_STR: REF BLOCK [, BYTE],    ! Where to put symbol
;  248          0314  1         OUT_LEN: REF VECTOR [, WORD]            ! Optional length of symbol
;  249          0315  1                                 ) =
;  250          0316  1
;  251          0317  1 !++
;  252          0318  1 ! FUNCTIONAL DESCRIPTION:
;  253          0319  1 !
;  254          0320  1 !       This routine returns the system's radix point symbol.  It
;  255          0321  1 !       should be used inside a digit string to divide the integer
;  256          0322  1 !       part from the fraction part, using the customary symbol.
;  257          0323  1 !       This routine works by translating the logical name
;  258          0324  1 !       SYS$RADIX_POINT.  If the translation fails, this routine
;  259          0325  1 !       returns ".", the US radix point symbol.  If the translation
;  260          0326  1 !       succeeds the text produced is returned.  Thus a system manager
;  261          0327  1 !       can define SYS$RADIX_POINT as a system-wide logical name to
;  262          0328  1 !       provide a default for all users, and an individual user with a
;  263          0329  1 !       special need can define SYS$RADIX_POINT as a user logical name
;  264          0330  1 !       to override the default.
;  265          0331  1 !
;  266          0332  1 ! CALLING SEQUENCE:
;  267          0333  1 !
;  268          0334  1 !       ret_status.wlc.v = LIB$RADIX_POINT (radix_point_str.wt.dx
;  269          0335  1 !                                           [,out_len.ww.r])
;  270          0336  1 !
;  271          0337  1 ! FORMAL PARAMETERS:
;  272          0338  1 !
;  273          0339  1 !       radix_point_str String to receive the radix point string
;  274          0340  1 !       out_len         Optional length of the radix_point string.
;  275          0341  1 !                       This is valuable if radix_point_str
;  276          0342  1 !                       is a fixed-length string, since it does not
;  277          0343  1 !                       include the padding.
;  278          0344  1 !
;  279          0345  1 ! IMPLICIT INPUTS:
;  280          0346  1 !
;  281          0347  1 !       System-wide (or user-defined) logical name SYS$RADIX_POINT
;  282          0348  1 !
;  283          0349  1 ! IMPLICIT OUTPUTS:
;  284          0350  1 !
;  285          0351  1 !       NONE
;  286          0352  1 !
;  287          0353  1 ! CONDITION CODES:
;  288          0354  1 !
;  289          0355  1 !       SS$_NORMAL      Normal successful completion
;  290          0356  1 !       LIB$_INVSTRDES  Invalid string descriptor
;  291          0357  1 !       LIB$_STRTRU     String Truncated
;  292          0358  1 !
;  293          0359  1 ! SIDE EFFECTS:
;  294          0360  1 !
;  295          0361  1 !       NONE
;  296          0362  1 !
;  297          0363  1 !--
;  298          0364  1
;  299          0365  2     BEGIN
;  300          0366  2
;  301          0367  2     BUILTIN
;  302          0368  2         ACTUALCOUNT;
```

```
;  303          0369  2
;  304          0370  2      !+
;  305          0371  2      ! Call GET_CUSTOM_SYMBOL to do the work.
;  306          0372  2      !-
;  307          0373  2
;  308          0374  3      RETURN ( GET_CUSTOM_SYMBOL (
;  309          0375  3          RADIX_POINT_STR [0,0,0,0],
;  310          0376  3          (IF ACTUALCOUNT () GEQU 2 THEN OUT_LEN [0] ELSE 0),
;  311          0377  3          UPLIT BYTE (%ASCIC'SYS$RADIX_POINT'),
;  312          0378  2          %C'.'));
;  313          0379  2
;  314          0380  1      END;                                    ! End of routine LIB$RADIX_POINT


4E  49  4F  50  5F  58  49  44  41  52  24  53  59  53  0F  00053 P.AAC:  .ASCII  <15>\SYS$RADIX_POINT\                    ;
                                                        54  00062                                                       ;


                                                        0000 00000          .ENTRY  LIB$RADIX_POINT, Save nothing       ; 0312
                                                    2E  DD  00002          PUSHL   #46                                 ; 0375
                                                E9  AF  9F  00004          PUSHAB  P.AAC                               ; 0377
                                      02            6C  91  00007          CMPB    (AP), #2                            ; 0376
                                                    05  1F  0000A          BLSSU   1$                                  ;
                                                08  AC  DD  0000C          PUSHL   OUT_LEN                             ;
                                                    02  11  0000F          BRB     2$                                  ;
                                                    7E  D4  00011 1$:      CLRL    -(SP)                               ;
                                                04  AC  DD  00013 2$:      PUSHL   RADIX_POINT_STR                     ; 0375
                                  0000V CF          04  FB  00016          CALLS   #4, GET_CUSTOM_SYMBOL               ;
                                                    04  0001B            RET                                         ; 0380

; Routine Size:  28 bytes,    Routine Base:  _LIB$CODE + 0063
```

```
;  316          0381  1 ROUTINE GET_CUSTOM_SYMBOL (%SBTTL 'Inner routine'
;  317          0382  1         OUT_STR: REF BLOCK [, BYTE],          ! Where to put symbol
;  318          0383  1         OUT_LEN: REF VECTOR [, WORD],         ! Optional length of symbol
;  319          0384  1         LOGNAM: REF VECTOR [, BYTE],          ! ASCIC logical name
;  320          0385  1         DEFAULT_CHAR: BYTE                    ! Default translation character
;  321          0386  1                         ) =
;  322          0387  1
;  323          0388  1 !++
;  324          0389  1 ! FUNCTIONAL DESCRIPTION:
;  325          0390  1 !
;  326          0391  1 !       This routine is called by LIB$CURRENCY, LIB$DIGIT_SEP and
;  327          0392  1 !       LIB$RADIX_POINT to do the actual translation and string copying.
;  328          0393  1 !
;  329          0394  1 ! CALLING SEQUENCE:
;  330          0395  1 !
;  331          0396  1 !       ret_status.wlc.v = GET_CUSTOM_SYMBOL (out_str.wt.dx,
;  332          0397  1 !                                             [out_len.ww.r],
;  333          0398  1 !                                             lognam.rt.r,
;  334          0399  1 !                                             default_char.rbu.v)
;  335          0400  1 !
;  336          0401  1 !
;  337          0402  1 ! FORMAL PARAMETERS:
;  338          0403  1 !
;  339          0404  1 !       out_str         String to receive the currency string
;  340          0405  1 !       out_len         Optional length of the currency string.
;  341          0406  1 !                       This is valuable if currency_str
;  342          0407  1 !                       is a fixed-length string, since it does not
;  343          0408  1 !                       include the padding.
;  344          0409  1 !       lognam          The ASCIC string containing the logical name to
;  345          0410  1 !                       translate.
;  346          0411  1 !       default_char    The default character to use if lognam doesn't translate.
;  347          0412  1 !
;  348          0413  1 ! IMPLICIT INPUTS:
;  349          0414  1 !
;  350          0415  1 !       NONE
;  351          0416  1 !
;  352          0417  1 ! IMPLICIT OUTPUTS:
;  353          0418  1 !
;  354          0419  1 !       NONE
;  355          0420  1 !
;  356          0421  1 ! CONDITION CODES:
;  357          0422  1 !
;  358          0423  1 !       LIB$_INVSTRDES  Invalid string descriptor
;  359          0424  1 !       LIB$_STRTRU     String truncated to fit into output
;  360          0425  1 !
;  361          0426  1 ! SIDE EFFECTS:
;  362          0427  1 !
;  363          0428  1 !       NONE
;  364          0429  1 !
;  365          0430  1 !--
;  366          0431  1
;  367          0432  2     BEGIN
;  368          0433  2
;  369          0434  2     LOCAL
;  370          0435  2         TRNLOG_STATUS,                  ! Status from $TRNLOG
;  371          0436  2         RET_STATUS,                     ! Return status
;  372          0437  2         EQUIV_LENGTH: WORD,             ! Length of equivalence string
```

```
;  373          0438  2          LOGDES : BLOCK [8, BYTE],       ! String descriptor for logical
;  374          0439  2                                          ! name
;  375          0440  2          NUMDES : BLOCK [8, BYTE],       ! String descriptor for
;  376          0441  2                                          ! translated string
;  377          0442  2          EQUIV_STRING: VECTOR [LNM$C_NAMLENGTH, BYTE];! Equivalence string
;  378          0443  2
;  379          0444  2      !+
;  380          0445  2      ! Initialize descriptors for logical name and translated string
;  381          0446  2      !-
;  382          0447  2          LOGDES [DSC$B_CLASS] = DSC$K_CLASS_S;
;  383          0448  2          LOGDES [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  384          0449  2          LOGDES [DSC$W_LENGTH] = .LOGNAM [0];
;  385          0450  2          LOGDES [DSC$A_POINTER] = LOGNAM [1];
;  386          0451  2
;  387          0452  2          NUMDES [DSC$B_CLASS] = DSC$K_CLASS_S;
;  388          0453  2          NUMDES [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  389          0454  2          NUMDES [DSC$W_LENGTH] = LNM$C_NAMLENGTH;
;  390          0455  2          NUMDES [DSC$A_POINTER] = EQUIV_STRING;
;  391          0456  2
;  392          0457  2      !+
;  393          0458  2      ! Translate and convert the logical name to determine
;  394          0459  2      ! the symbol.
;  395          0460  2      !-
;  396        P 0461  2          TRNLOG_STATUS = $TRNLOG (
;  397        P 0462  2                                  LOGNAM = LOGDES,
;  398        P 0463  2                                  RSLLEN = EQUIV_LENGTH,
;  399          0464  2                                  RSLBUF = NUMDES);
;  400          0465  2
;  401          0466  3          IF (( NOT .TRNLOG_STATUS) OR (.TRNLOG_STATUS EQL SS$_NOTRAN))
;  402          0467  2          THEN
;  403          0468  3              BEGIN
;  404          0469  3              EQUIV_STRING [0] = .DEFAULT_CHAR;       ! Use default
;  405          0470  3              EQUIV_LENGTH = 1;
;  406          0471  2              END;
;  407          0472  2
;  408          0473  2      !+
;  409          0474  2      ! EQUIV_STRING is now the string to return to the caller.
;  410          0475  2      !-
;  411          0476  2          RET_STATUS = LIB$SCOPY_R_DX6 ( .EQUIV_LENGTH,
;  412          0477  2                                         EQUIV_STRING,
;  413          0478  2                                         .OUT_STR);
;  414          0479  2
;  415          0480  2      !+
;  416          0481  2      ! If caller supplied optional 2nd parameter (OUT_LEN), fill it in.
;  417          0482  2      ! No need to check status from LIB$ANALYZE_SDESC_R2.  If OUT_STR
;  418          0483  2      ! was bad, call to LIB$SCOPY_R_DX6 will have caught it.
;  419          0484  2      !-
;  420          0485  3          IF ( OUT_LEN [0] NEQA 0)
;  421          0486  2          THEN
;  422          0487  3              BEGIN
;  423          0488  3              LOCAL
;  424          0489  3                  WRITTEN_LEN: WORD;
;  425          0490  3              LIB$ANALYZE_SDESC_R2 (.OUT_STR;
;  426          0491  3                                    WRITTEN_LEN);
;  427          0492  3              IF .WRITTEN_LEN LSSU .EQUIV_LENGTH
;  428          0493  3              THEN
;  429          0494  3                  EQUIV_LENGTH = .WRITTEN_LEN;
```

```
;  430          0495  3          OUT_LEN [0] = .EQUIV_LENGTH;
;  431          0496  2          END;
;  432          0497  2
;  433          0498  2      RETURN (.RET_STATUS);
;  434          0499  1      END;                                ! End of routine GET_CUSTOM_SYMBOL


                                                                .EXTRN  SYS$TRNLOG

                                                        007C 00000 GET_CUSTOM_SYMBOL:
                                                                .WORD   Save R2,R3,R4,R5,R6                     ; 0381
                                    5E      FEEC    CE  9E 00002        MOVAB   -276(SP), SP                    ;
                                FA  AD      010E    8F  B0 00007        MOVW    #270, LOGDES+2                  ; 0448
                                F8  AD        0C    BC  9B 0000D        MOVZBW  @LOGNAM, LOGDES                 ; 0449
                            FC  AD  0C      AC      01  C1 00012        ADDL3   #1, LOGNAM, LOGDES+4            ; 0450
                                F0  AD  010E00FF    8F  D0 00018        MOVL    #17694975, NUMDES               ; 0454
                                F4  AD        04    AE  9E 00020        MOVAB   EQUIV_STRING, NUMDES+4          ; 0455
                                                    7E  7C 00025        CLRQ    -(SP)                           ; 0464
                                                    7E  D4 00027        CLRL    -(SP)                           ;
                                          F0    AD  9F 00029        PUSHAB  NUMDES                          ;
                                          10    AE  9F 0002C        PUSHAB  EQUIV_LENGTH                    ;
                                          F8    AD  9F 0002F        PUSHAB  LOGDES                          ;
                                00000000G   00        06  FB 00032        CALLS   #6, SYS$TRNLOG                  ;
                                            09        50  E9 00039        BLBC    TRNLOG_STATUS, 1$               ; 0466
                                00000629    8F        50  D1 0003C        CMPL    TRNLOG_STATUS, #1577            ;
                                                      08  12 00043        BNEQ    2$                              ;
                                      04    AE        10  AC 90 00045 1$:     MOVB    DEFAULT_CHAR, EQUIV_STRING      ; 0469
                                            6E              01  B0 0004A        MOVW    #1, EQUIV_LENGTH                ; 0470
                                            51        04    AE  9E 0004D 2$:     MOVAB   EQUIV_STRING, R1        ; 0476
                                            52        04    AC  D0 00051        MOVL    OUT_STR, R2             ;
                                            50              6E  3C 00055        MOVZWL  EQUIV_LENGTH, R0        ;
                                               00000000G    00  16 00058        JSB     LIB$SCOPY_R_DX6         ;
                                            53              50  D0 0005E        MOVL    R0, RET_STATUS          ;
                                                    08      AC  D5 00061        TSTL    OUT_LEN                 ; 0485
                                                            19  13 00064        BEQL    4$                      ;
                                            50        04    AC  D0 00066        MOVL    OUT_STR, R0             ; 0490
                                               00000000G    00  16 0006A        JSB     LIB$ANALYZE_SDESC_R2    ;
                                            50              51  D0 00070        MOVL    R1, WRITTEN_LEN         ;
                                            6E              50  B1 00073        CMPW    WRITTEN_LEN, EQUIV_LENGTH       ; 0492
                                                            03  1E 00076        BGEQU   3$                      ;
                                            6E              50  B0 00078        MOVW    WRITTEN_LEN, EQUIV_LENGTH       ; 0494
                                      08    BC              6E  B0 0007B 3$:     MOVW    EQUIV_LENGTH, @OUT_LEN          ; 0495
                                            50              53  D0 0007F 4$:     MOVL    RET_STATUS, R0          ; 0498
                                                                04 00082        RET                             ; 0499

; Routine Size:  131 bytes,    Routine Base:  _LIB$CODE + 007F

;  435          0500  1
;  436          0501  1 END                                     ! End of module LIB$CUSTOMIZE_OUTPUT
;  437          0502  1
;  438          0503  0 ELUDOM
```

```
;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  _LIB$CODE                       258  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        11         0         581         00:00.7
;  _$255$DUA28:[LIBRTL.OBJ]RTLLIB.L32;1        36         2         5           8         00:00.1




;                                   COMMAND QUALIFIERS

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

; Size:         215 code + 43 data bytes
; Run Time:         00:05.1
; Elapsed Time:     00:21.2
; Lines/CPU Min:     5940
; Lexemes/CPU-Min: 19877
; Memory Used:  68 pages
; Compilation Complete
```

LIBCVTDX
LIS

LIBCVTMAC
LIS

LIBCRCTAB
LIS

LIBCVTDF
LIS

LIBCUSTOM
LIS

LIBCREDIR
LIS

LIBCUTATB
LIS

LIBCOMMON
LIS

LIBCRC
LIS