```
RRRRRRRRRRR      UUU          UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF   FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF   FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF   FFFFFFFFFFFFFFF
RRR          RRR UUU          UUU  NNN         NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNN         NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNN         NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNNNNN      NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNNNNN      NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNNNNN      NNN  000          000 FFF               FFF
RRRRRRRRRRR      UUU          UUU  NNN   NNN   NNN  000          000 FFFFFFFFFFF       FFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN   NNN   NNN  000          000 FFFFFFFFFFF       FFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN   NNN   NNN  000          000 FFFFFFFFFFF       FFFFFFFFFFF
RRR   RRR        UUU          UUU  NNN       NNNNNN 000          000 FFF               FFF
RRR   RRR        UUU          UUU  NNN       NNNNNN 000          000 FFF               FFF
RRR   RRR        UUU          UUU  NNN       NNNNNN 000          000 FFF               FFF
RRR      RRR     UUU          UUU  NNN          NNN 000          000 FFF               FFF
RRR      RRR     UUU          UUU  NNN          NNN 000          000 FFF               FFF
RRR      RRR     UUU          UUU  NNN          NNN 000          000 FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
```

```
_$2

Sym
---
NDX
NDX
NUM
NUM
OPE
OUT
PAC
PAC
PAC
PAC
PAC
PAC
PAC
PAD
PAG
PAG
PAG
PAG
PAG
PAG
PAG
PER
PUT
RCO
RIN

RLI
RNO
RNO
RTY
SAV
STR
STR
STR
STR
STR
STR

STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
```

FILEID**CONVBL

```
  CCCCCCCC    000000    NN      NN  VV      VV  BBBBBBBB    LL
  CCCCCCCC    000000    NN      NN  VV      VV  BBBBBBBB    LL
CC           00      00  NN      NN  VV      VV  BB      BB  LL
CC           00      00  NN      NN  VV      VV  BB      BB  LL
CC           00      00  NNNN    NN  VV      VV  BB      BB  LL
CC           00      00  NNNN    NN  VV      VV  BB      BB  LL
CC           00      00  NN  NN  NN  VV      VV  BBBBBBBB    LL
CC           00      00  NN  NN  NN  VV      VV  BBBBBBBB    LL
CC           00      00  NN    NNNN  VV      VV  BB      BB  LL
CC           00      00  NN    NNNN  VV      VV  BB      BB  LL
CC           00      00  NN      NN    VV  VV    BB      BB  LL              ....
CC           00      00  NN      NN    VV  VV    BB      BB  LL              ....
  CCCCCCCC    000000    NN      NN      VV      BBBBBBBB    LLLLLLLLLL      ....
  CCCCCCCC    000000    NN      NN      VV      BBBBBBBB    LLLLLLLLLL      ....


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 CONVBL (                                        !
;     2          0002  0                 IDENT = 'V04-000'
;     3        P 0003  0 %BLISS32[,
;     4        P 0004  0         ADDRESSING_MODE(EXTERNAL=LONG_RELATIVE,NONEXTERNAL=LONG_RELATIVE)
;     5          0005  0         ]
;     6          0006  0                 ) =
;     7          0007  1 BEGIN
;     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 COPYR     .OTICE. THIS SOFTWARE OR  ANY  OTHER    *
;    18          0018  1 !*  COPIES THEREOF MAY NOT BE PRO.. _D 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 ! FACILITY:      DSR (Digital Standard RUNOFF) / DSRPLUS
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT: Convert a binary number into a vector of characters and
;    36          0036  1 !           return the result and character count.
;    37          0037  1 !
;    38          0038  1 !
;    39          0039  1 ! ENVIRONMENT: Transportable
;    40          0040  1 !
;    41          0041  1 ! AUTHOR: R.W.Friday     CREATION DATE: April, 1979
;    42          0042  1 !
```

```
;    44          0043  1 %SBTTL 'Revision History'
;    45          0044  1 !
;    46          0045  1 ! MODIFIED BY:
;    47          0046  1 !
;    48          0047  1 !       003     KFA00003        Ken Alden       07-Mar-1983
;    49          0048  1 !               Global edit of all modules. Updated module names, idents,
;    50          0049  1 !               copyright dates. Changed require files to BLISS library.
;    51          0050  1 !
;    52          0051  1 !--
```

```
;    54          0052  1 %SBTTL 'Module Level Declarations'
;    55          0053  1
;    56          0054  1 !
;    57          0055  1
;    58          0056  1 ! OWN STORAGE:
;    59          0057  1 !
;    60          0058  1 OWN
;    61          0059  1     TABLE : INITIAL ( CH$PTR(UPLIT('0ABCDEFGHIJKLMNOPQRSTUVWXYZ')) );
```

```
;    63          0060  1 GLOBAL ROUTINE CONVBL (BINARY_NUMBER, KHARACTERS, KHARACTER_COUNT, ULM) : NOVALUE =
;    64          0061  1
;    65          0062  1 !++
;    66          0063  1 ! FUNCTIONAL DESCRIPTION:
;    67          0064  1 !
;    68          0065  1 !       Converts 'binary_number' to a vector of characters,
;    69          0066  1 !       returning them in 'kharacters'; kharacter_count is the
;    70          0067  1 !       number of characters that result.
;    71          0068  1 !       The absolute value of 'binary_number' is converted,
;    72          0069  1 !       so that the user is responsible for handling negative numbers.
;    73          0070  1 !       ULM determines what the string of letters looks like:
;    74          0071  1 !               -1 means return all in upper case.
;    75          0072  1 !                0 means return all in lower case.
;    76          0073  1 !               +1 means capitalize first letter.
;    77          0074  1 !
;    78          0075  1 ! FORMAL PARAMETERS:
;    79          0076  1 !
;    80          0077  1 !       See FUNCTIONAL DESCRIPTION
;    81          0078  1 !
;    82          0079  1 ! IMPLICIT INPUTS:      None
;    83          0080  1 !
;    84          0081  1 ! IMPLICIT OUTPUTS:     None
;    85          0082  1 !
;    86          0083  1 ! ROUTINE VALUE:
;    87          0084  1 ! COMPLETION CODES:     None
;    88          0085  1 !
;    89          0086  1 ! SIDE EFFECTS: None
;    90          0087  1 !
;    91          0088  1 !--
;    92          0089  1
;    93          0090  2     BEGIN
;    94          0091  2
;    95          0092  2     MAP
;    96          0093  2         KHARACTERS : REF VECTOR;
;    97          0094  2
;    98          0095  2     LOCAL
;    99          0096  2         LEFT_TO_CONVERT;
;   100          0097  2
;   101          0098  2     .KHARACTER_COUNT = 0;
;   102          0099  2     LEFT_TO_CONVERT = ABS (.BINARY_NUMBER);
;   103          0100  2
;   104          0101  2     !Catch the special case, when the number is zero.
;   105          0102  2     IF  .LEFT_TO_CONVERT EQL 0                      !
;   106          0103  2     THEN
;   107          0104  3         BEGIN
;   108          0105  3         KHARACTERS [0] = CH$RCHAR(.TABLE);      !First character is the special 'zero' character.
;   109          0106  3         .KHARACTER_COUNT = 1;
;   110          0107  3         RETURN;
;   111          0108  2         END;
;   112          0109  2
;   113          0110  2     !The normal case, when the number is not zero.
;   114          0111  2     DO
;   115          0112  3         BEGIN
;   116          0113  3         LOCAL
;   117          0114  3             GROUP,
;   118          0115  3             UNITS;
;   119          0116  3
```

```
;  120          0117  3          GROUP = (.LEFT_TO_CONVERT - 1)/26;
;  121          0118  3          UNITS = .LEFT_TO_CONVERT - (26*.GROUP);
;               0119  3          KHARACTERS [..KHARACTER_COUNT] = CH$RCHAR(CH$PLUS(.TABLE,.UNITS));
;               0120  3          .KHARACTER_COUNT = ..KHARACTER_COUNT + 1;
;  124          0121  3          LEFT_TO_CONVERT = .GROUP;
;  125          0122  3          END
;  126          0123  2      UNTIL .LEFT_TO_CONVERT EQL 0;
;  127          0124  2
;  128          0125  2      !Apply case rules
;  129          0126  2      IF  .ULM EQL -1
;  130          0127  2      THEN
;  131          0128  2          !User is content with all in upper case.
;  132          0129  2          RETURN;
;  133          0130  2
;  134          0131  2      !Convert the necessary number of letters to upper case.
;  135          0132  2      INCR I FROM 0 TO (..KHARACTER_COUNT - .ULM - 1) DO
;  136          0133  2              KHARACTERS [.I] = .KHARACTERS [.I] - %C'A' + %C'a';
;  137          0134  2
;  138          0135  2      RETURN;
;  139          0136  1      END;                                    !End of CONVBL


                                                    .TITLE  CONVBL
                                                    .IDENT  \V04-000\

                                                    .PSECT  $PLIT$,NOWRT,NOEXE,2

4E  4D  4C  4B  4A  49  48  47  46  45  44  43  42  41  30  00000 P.AAA:  .ASCII  \0ABCDEFGHIJKLMNOPQRSTUVWXYZ\<0>                ;
        00  5A  59  58  57  56  55  54  53  52  51  50  4F  0000F                                                          ;

                                                    .PSECT  $OWN$,NOEXE,2

                                00000000' 00000 TABLE:  .ADDRESS P.AAA                                                   ;


                                                    .PSECT  $CODE$,NOWRT,2

                                    001C 00000          .ENTRY  CONVBL, Save R2,R3,R4                                ; 0060
                    54      0C  AC  D0 00002            MOVL    KHARACTER_COUNT, R4                                  ; 0098
                                64  D4 00006            CLRL    (R4)                                                 ;
                    50      04  AC  D0 00008            MOVL    BINARY_NUMBER, LEFT_TO_CONVERT                       ; 0099
                                03  18 0000C            BGEQ    1$                                                   ;
                    50          50  CE 0000E            MNEGL   LEFT_TO_CONVERT, LEFT_TO_CONVERT                     ;
                                0C  12 00011 1$:        BNEQ    2$                                                   ; 0102
                08  BC 00000000' FF  9A 00013            MOVZBL  @TABLE, @KHARACTERS                                  ; 0105
                    64          01  D0 0001B            MOVL    #1, (R4)                                             ; 0106
                                04  0001E              RET                                                          ; 0104
                    51      FF  A0  9E 0001F 2$:        MOVAB   -1(R0), R1                                           ; 0117
                    51          1A  C6 00023            DIVL2   #26, GROUP                                           ;
          52        51          1A  C5 00026            MULL3   #26, GROUP, R2                                       ; 0118
          53        50          52  C3 0002A            SUBL3   R2, LEFT_TO_CONVERT, UNITS                           ;
                    52          64  D0 0002E            MOVL    (R4), R2                                             ; 0119
                08 BC42 00000000'FF43  9A 00031         MOVZBL  @TABLE[UNITS], @KHARACTERS[R2]                       ;
                                64  D6 0003B            INCL    (R4)                                                 ; 0120
                    50          51  D0 0003D            MOVL    GROUP, LEFT_TO_CONVERT                               ; 0121
                                DD  12 00040            BNEQ    2$                                                   ; 0123
```

```
                    FFFFFFFF   8F        10   AC  D1 00042          CMPL    ULM, #-1                                ; 0126
                                             13  13 0004A          BEQL    5$                                      ;
                         54          64      10   AC  C3 0004C          SUBL3   ULM, (R4), R4                       ; 0132
                                     50           01  CE 00051          MNEGL   #1, I                               ;
                                                  05  11 00054          BRB     4$                                  ;
                               08 BC40            20  C0 00056 3$:      ADDL2   #32, @KHARACTERS[I]                 ; 0133
                         F7          50           54  F2 0005B 4$:      AOBLSS  R4, I, 3$                           ;
                                                  04 0005F 5$:      RET                                             ; 0136

; Routine Size:  96 bytes,    Routine Base:  $CODE$ + 0000

;  140          0137  1
;  141          0138  1 END                                                 !End of module
;  142          0139  0 ELUDOM




;                                   PSECT SUMMARY
;
;       Name                    Bytes                       Attributes
;
;  $PLIT$                         28  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                           4  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                         96  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)




;                                       COMMAND QUALIFIERS

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

; Size:          96 code + 32 data bytes
; Run Time:         00:02.6
; Elapsed Time:     00:06.4
; Lines/CPU Min:     3183
; Lexemes/CPU-Min: 10763
; Memory Used:  39 pages
; Compilation Complete
```

CNVDAT
LIS

CONVBB
LIS

CONVBR
LIS

CNTVMSMSG
LIS

CONTENTS
LIS

CNTVMS
LIS

CNTVRS
LIS

CONTRL
LIS

CONVBI
LIS