K 7

```
RRRRRRRRRRRR     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
RRRRRRRRRRRR     UUU         UUU  NNN   NNN   NNN  000         000  FFFFFFFFFFFF     FFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN   NNN   NNN  000         000  FFFFFFFFFFFF     FFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN   NNN   NNN  000         000  FFFFFFFFFFFF     FFFFFFFFFFFF
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
```

```
_$2

Syw
---
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
STK
STR
STR
STR
STR
STR
STR
STR
STR
STR
```

```
**FILE**ID**PACK                                  I 15




PPPPPPPP      AAAAAA      CCCCCCCC  KK      KK
PPPPPPPP      AAAAAA      CCCCCCCC  KK      KK
PP      PP  AA      AA  CC          KK      KK
PP      PP  AA      AA  CC          KK      KK
PP      PP  AA      AA  CC          KK    KK
PP      PP  AA      AA  CC          KK    KK
PPPPPPPP    AA      AA  CC          KKKKKK
PPPPPPPP    AA      AA  CC          KKKKKK
PP          AAAAAAAAAA  CC          KK    KK
PP          AAAAAAAAAA  CC          KK    KK
PP          AA      AA  CC          KK      KK      ....
PP          AA      AA  CC          KK      KK      ....
PP          AA      AA    CCCCCCCC  KK      KK      ....
PP          AA      AA    CCCCCCCC  KK      KK      ....

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 %TITLE 'Number outputting routines'
;    2          0002  0 MODULE PACK (           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 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 ! FACILITY:      DSR (Digital Standard RUNOFF) / DSRPLUS
;   34          0034  1 !
;   35          0035  1 ! ABSTRACT:
;   36          0036  1 !
;   37          0037  1 !       Routines to output numbers, a RUNOFF style page number,
;   38          0038  1 !       RUNOFF style section numbers, and ordinary strings.
;   39          0039  1 !
;   40          0040  1 ! ENVIRONMENT:  Transportable
;   41          0041  1 !
;   42          0042  1 ! AUTHOR:  D. Knight    , CREATION DATE:  June 1978
;   43          0043  1 !
;    45          0044  1 %SBTTL 'Revision History'
;    46          0045  1 !
;    47          0046  1 ! MODIFIED BY:
;    48          0047  1 !
;    49          0048  1 !        008     REM00008        Ray Marshall    21-Mar-1984
;    50          0049  1 !                Implemented the foreign language conditionals for compiling
;    51          0050  1 !                  fixed output words.  At this time, we only have the German
;    52          0051  1 !                  translations available.  But, since the German word for INDEX
;    53          0052  1 !                  is the same as in English, that conditional isn't used.
;    54          0053  1 !
;    55          0054  1 !        007     KAD00007        Keith Dawson    07-Mar-1983
;    56          0055  1 !                Global edit of all modules. Updated module names, idents,
;    57          0056  1 !                copyright dates. Changed require files to BLISS library.
;    58          0057  1 !
;    59          0058  1 !--
;     61           0059   1 %SBTTL 'Module Level Declarations'
;     62           0060   1
;     63           0061   1 !
;     64           0062   1 ! TABLE OF CONTENTS:
;     65           0063   1 !
;     66           0064   1
;     67           0065   1 FORWARD ROUTINE
;     68           0066   1         PACBAS,                          !Convert to specified base and pack
;     69           0067   1         PACLET,                          !Convert to letters and pack
;     70           0068   1         PACPAG,                          !Convert page number to ASCII
;     71           0069   1         PACROM,                          !Convert string to Roman numerals.
;     72           0070   1         PACSTR,                          !Pack string into print line
;     73           0071   1         PACSEC,                          !Convert section number to ASCII
;     74           0072   1         PACXXX;                          !Convert any counter to whatever XXX indicates.
;     75           0073   1
;     76           0074   1 !
;     77           0075   1 ! INCLUDE FILES:
;     78           0076   1 !
;     79           0077   1
;     80           0078   1 LIBRARY 'NXPORT:XPORT';                  ! XPORT Library
;     81           0079   1 REQUIRE 'REQ:RNODEF';                    ! RUNOFF variant definitions
;     82           0210   1
;     83         U 0211   1 %IF DSRPLUS %THEN
;     84         U 0212   1 LIBRARY 'REQ:DPLLIB';                    ! DSRPLUS BLISS Library
;     85           0213   1 %ELSE
;     86           0214   1 LIBRARY 'REQ:DSRLIB';                    ! DSR BLISS Library
;     87           0215   1 %FI
;     88           0216   1
;     89           0217   1 !
;     90           0218   1 ! MACROS:
;     91           0219   1 !
;     92           0220   1 !
;     93           0221   1 ! EQUATED SYMBOLS:
;     94           0222   1 !
;     95           0223   1 !
;     96           0224   1 ! OWN STORAGE:
;     97           0225   1 !
;     98           0226   1 !
;     99           0227   1 ! EXTERNAL REFERENCES:
;    100           0228   1 !
;    101           0229   1
;    102           0230   1 EXTERNAL ROUTINE
;    103           0231   1     CONVBB,                                      !Convert Binary to specified base
;    104           0232   1     CONVBL,                                      !Convert binary to letters.
;    105           0233   1     CONVBR;                                      !Convert binary to Roman.
;    106           0234   1
;  108          0235  1 %SBTTL 'Routine PACBAS'
;  109          0236  1 GLOBAL ROUTINE PACBAS (VALUE, LINE_PTR, BASE) =          !
;  110          0237  1
;  111          0238  1 !++
;  112          0239  1 ! FUNCTIONAL DESCRIPTION:
;  113          0240  1 !
;  114          0241  1 !       Convert a decimal number to ASCII and place it in the
;  115          0242  1 !       output text.
;  116          0243  1 !
;  117          0244  1 ! FORMAL PARAMETERS:
;  118          0245  1 !
;  119          0246  1 !       VALUE - number to be converted (no negative numbers allowed)
;  120          0247  1 !       LINE_PTR - string pointer into output line for text
;  121          0248  1 !       BASE - base in which number is to be expressed.
;  122          0249  1 !
;  123          0250  1 ! IMPLICIT INPUTS:      None
;  124          0251  1 !
;  125          0252  1 ! IMPLICIT OUTPUTS:     None
;  126          0253  1 !
;  127          0254  1 ! ROUTINE VALUE:
;  128          0255  1 ! COMPLETION CODES:
;  129          0256  1 !
;  130          0257  1 !       The length of the converted string is returned
;  131          0258  1 !
;  132          0259  1 ! SIDE EFFECTS: None
;  133          0260  1 !
;  134          0261  1 !--
;  135          0262  1
;  136          0263  2     BEGIN
;  137          0264  2
;  138          0265  2     LOCAL
;  139          0266  2         COUNT,
;  140          0267  2         TEXT : VECTOR [10];
;  141          0268  2
;  142          0269  2     !Convert to decimal
;  143          0270  2     CONVBB (.VALUE, TEXT, COUNT, .BASE);
;  144          0271  2     !Now pack it into the output line
;  145          0272  2
;  146          0273  2     DECR I FROM .COUNT TO 1 DO
;  147          0274  2         CH$WCHAR_A (.TEXT [.I - 1], .LINE_PTR);
;  148          0275  2
;  149          0276  2     .COUNT
;  150          0277  1     END;                                        !End of PACBAS


                                              .TITLE  PACK Number outputting routines
                                              .IDENT  \V04-000\

                                              .EXTRN  CONVBB, CONVBL, CONVBR

                                              .PSECT  $CODE$,NOWRT,2

                              0000 00000      .ENTRY  PACBAS, Save nothing                 ; 0236
            5E            2C   C2 00002      SUBL2   #44, SP                               ;
                      0C  AC   DD 00005      PUSHL   BASE                                  ; 0270
                      04  AE   9F 00008      PUSHAB  COUNT                                 ;
                      0C  AE   9F 0000B      PUSHAB  TEXT                                  ;
```

```
                                          04    AC  DD 0000E          PUSHL   VALUE                                  ;
                       00000000G  EF            04  FB 00011          CALLS   #4, CONVBB                             ;
                  50              6E            01  C1 00018          ADDL3   #1, COUNT, I                           ; 0274
                                                0B  11 0001C          BRB     2$                                     ;
                                  51      08    AC  D0 0001E 1$:      MOVL    LINE_PTR, R1                           ;
                            00    B1          6E40  F6 00022          CVTLB   TEXT=4[I], @0(R1)                      ;
                                                61  D6 00027          INCL    (R1)                                   ;
                                  F2            50  F5 00029 2$:      SOBGTR  I, 1$                                  ;
                                  50            6E  D0 0002C          MOVL    COUNT, R0                              ; 0277
                                                    04 0002F          RET                                            ;

; Routine Size:  48 bytes,    Routine Base:  $CODE$ + 0000


;  151          0278  1
;  153       0279  1 %SBTTL 'Routine PACLET'
;  154       0280  1 GLOBAL ROUTINE PACLET (VALUE,LINE_PTR,ULM) =    !
;  155       0281  1
;  156       0282  1 !++
;  157       0283  1 ! FUNCTIONAL DESCRIPTION:
;  158       0284  1 !
;  159       0285  1 !       Convert a decimal number to letters and place it in the
;  160       0286  1 !       output text.
;  161       0287  1 !
;  162       0288  1 ! FORMAL PARAMETERS:
;  163       0289  1 !
;  164       0290  1 !       VALUE - number to be converted (no negative numbers allowed)
;  165       0291  1 !       LINE_PTR - string pointer into output line for text
;  166       0292  1 !       ULM - indicates whether result is to be in upper, lower, or mixed case.
;  167       0293  1 !
;  168       0294  1 ! IMPLICIT INPUTS:      None
;  169       0295  1 !
;  170       0296  1 ! IMPLICIT OUTPUTS:     None
;  171       0297  1 !
;  172       0298  1 ! ROUTINE VALUE:
;  173       0299  1 ! COMPLETION CODES:
;  174       0300  1 !
;  175       0301  1 !       The length of the converted string is returned
;  176       0302  1 !
;  177       0303  1 ! SIDE EFFECTS: None
;  178       0304  1 !
;  179       0305  1 !--
;  180       0306  1
;  181       0307  2     BEGIN
;  182       0308  2
;  183       0309  2     LOCAL
;  184       0310  2         COUNT,
;  185       0311  2         TEXT:   VECTOR[10];
;  186       0312  2
;  187       0313  2     !Convert to letters
;  188       0314  2     CONVBL(.VALUE,TEXT,COUNT,.ULM);
;  189       0315  2
;  190       0316  2     !Now pack it into the output line
;  191       0317  2     DECR I FROM .COUNT TO 1 DO
;  192       0318  2         CH$WCHAR_A(.TEXT[.I-1],.LINE_PTR);
;  193       0319  2
;  194       0320  2     .COUNT
;  195       0321  2
;  196       0322  1     END;                              !End of PACLET



                                        0000 00000      .ENTRY  PACLET, Save nothing                        ; 0280
                     5E               2C  C2 00002      SUBL2   #44, SP                                     ;
                               0C    AC  DD 00005      PUSHL   ULM                                         ; 0314
                               04    AE  9F 00008      PUSHAB  COUNT                                       ;
                               0C    AE  9F 0000B      PUSHAB  TEXT                                        ;
                               04    AC  DD 0000E      PUSHL   VALUE                                       ;
          00000000G  EF               04  FB 00011      CALLS   #4, CONVBL                                  ;
      50             6E               01  C1 00018      ADDL3   #1, COUNT, I                                ; 0318
```

```
                              0B  11 0001C        BRB     2$                                         ;
              51        08    AC  D0 0001E 1$:    MOVL    LINE_PTR, R1                               ;
        00    B1            6E40  F6 00022        CVTLB   TEXT-4[I], @0(R1)                          ;
                              61  D6 00027        INCL    (R1)                                       ;
              F2              50  F5 00029 2$:    SOBGTR  I, 1$                                      ;
              50              6E  D0 0002C        MOVL    COUNT, R0                                  ; 0322
                                  04 0002F        RET                                                ;

; Routine Size:  48 bytes,    Routine Base:  $CODE$ + 0030
```

```
:   198         0323   1 %SBTTL 'Routine PACPAG'
:   199         0324   1 GLOBAL ROUTINE PACPAG (LINE_NO,LINE_PTR) =       !
:   200         0325   1
:   201         0326   1 !++
:   202         0327   1 ! FUNCTIONAL DESCRIPTION:
:   203         0328   1 !
:   204         0329   1 !      Pack multiple part page number into print line.
:   205         0330   1 !
:   206         0331   1 ! FORMAL PARAMETERS:
:   207         0332   1 !
:   208         0333   1 !      LINE_NO  - Address of line number to be packed
:   209         0334   1 !      LINE_PTR - String pointer to print line.
:   210         0335   1 !
:   211         0336   1 ! IMPLICIT INPUTS:       None.
:   212         0337   1 !
:   213         0338   1 ! IMPLICIT OUTPUTS:      None
:   214         0339   1 !
:   215         0340   1 ! ROUTINE VALUE:
:   216         0341   1 ! COMPLETION CODES:
:   217         0342   1 !
:   218         0343   1 !      The number of characters generated is returned.
:   219         0344   1 !
:   220         0345   1 ! SIDE EFFECTS: None
:   221         0346   1 !
:   222         0347   1 !--
:   223         0348   1
:   224         0349   2     BEGIN
:   225         0350   2     MAP
:   226         0351   2         LINE_NO: REF BLOCK;
:   227         0352   2     LOCAL
:   228         0353   2         CHAR_COUNT;
:   229         0354   2
:   230         0355   2     CHAR_COUNT=0;
:   231         0356   2
:   232         0357   2     !Pack section into line
:   233         0358   2     IF .LINE_NO[SCT_TYP] NEQ 0
:   234         0359   2     THEN
:   235         0360   3         BEGIN
:   236         0361   3
:   237         0362   3         CHAR_COUNT=.CHAR_COUNT+PACSEC(.LINE_NO,.LINE_PTR);
:   238         0363   3
:   239         0364   3         !Now put in the "-" for separator
:   240         0365   3         CH$WCHAR_A(%C'-',.LINE_PTR);
:   241         0366   3         CHAR_COUNT=.CHAR_COUNT+1;
:   242         0367   3
:   243         0368   2         END;
:   244         0369   2
:   245         0370   2     !Pack page number into line
:   246         0371   2     CHAR_COUNT=.CHAR_COUNT+PACXXX(.LINE_NO[SCT_PAGE],.LINE_PTR,.LINE_NO[SCT_PAGE_D]);
:   247         0372   2
:   248         0373   2     !Place sub-page into line
:   249         0374   2     IF .LINE_NO[SCT_SUB_PAGE] NEQ 0
:   250         0375   2     THEN
:   251         0376   3         BEGIN
:   252         0377   3         CHAR_COUNT=.CHAR_COUNT+PACXXX(.LINE_NO[SCT_SUB_PAGE],.LINE_PTR,.LINE_NO[SCT_SUBPG_D]);
:   253         0378   2         END;
:   254         0379   2
```

```
;  255          0380  2     .CHAR_COUNT
;  256          0381  2
;  257          0382  1     END;                         .End of PACPAG




                                                       001C 00000         .ENTRY  PACPAG, Save R2,R3,R4                        ; 0324
                                       54 00000000V  EF  9E 00002         MOVAB   PACXXX, R4                                   ;
                                                     53  D4 00009         CLRL    CHAR_COUNT                                   ; 0355
                                       52       04   AC  D0 0000B         MOVL    LINE_NO, R2                                  ; 0358
                                       0F            62  93 0000F         BITB    (R2), #15                                    ;
                                                     1B  13 00012         BEQL    1$                                           ;
                                                08   AC  DD 00014         PUSHL   LINE_PTR                                     ; 0362
                                                     52  DD 00017         PUSHL   R2                                           ;
                            00000000V  EF            02  FB 00019         CALLS   #2, PACSEC                                   ;
                                       53            50  C0 00020         ADDL2   R0, CHAR_COUNT                               ;
                                       50       08   BC  D0 00023         MOVL    @LINE_PTR, R0                                ; 0365
                                       60            2D  90 00027         MOVB    #45, (R0)                                    ;
                                                08   BC  D6 0002A         INCL    @LINE_PTR                                    ;
                                                     53  D6 0002D         INCL    CHAR_COUNT                                   ; 0366
           7E            62            04            04  EF 0002F 1$:     EXTZV   #4, #4, (R2), -(SP)                          ; 0371
                                                08   AC  DD 00034         PUSHL   LINE_PTR                                     ;
                                                08   A2  DD 00037         PUSHL   8(R2)                                        ;
                                       64            03  FB 0003A         CALLS   #3, PACXXX                                   ;
                                       53            50  C0 0003D         ADDL2   R0, CHAR_COUNT                               ;
                                                02   A2  B5 00040         TSTW    2(R2)                                        ; 0374
                                                     13  13 00043         BEQL    2$                                           ;
           7E       0C   A2            04            00  EF 00045         EXTZV   #0, #4, 12(R2), -(SP)                        ; 0377
                                                08   AC  DD 0004B         PUSHL   LINE_PTR                                     ;
                                       7E       02   A2  3C 0004E         MOVZWL  2(R2), -(SP)                                 ;
                                       64            03  FB 00052         CALLS   #3, PACXXX                                   ;
                                       53            50  C0 00055         ADDL2   R0, CHAR_COUNT                               ;
                                       50            53  D0 00058 2$:     MOVL    CHAR_COUNT, R0                               ; 0382
                                                         04 0005B         RET                                                  ;

; Routine Size:  92 bytes,    Routine Base:  $CODE$ + 0060
;  259        0383  1 %SBTTL 'Routine PACROM'
;  260        0384  1 GLOBAL ROUTINE PACROM (VALUE,LINE_PTR,ULM) =    !
;  261        0385  1
;  262        0386  1 !++
;  263        0387  1 ! FUNCTIONAL DESCRIPTION:
;  264        0388  1 !
;  265        0389  1 !       Convert a decimal number to roman numerals and place it in the
;  266        0390  1 !       output text.
;  267        0391  1 !
;  268        0392  1 ! FORMAL PARAMETERS:
;  269        0393  1 !
;  270        0394  1 !       VALUE - number to be converted (no negative numbers allowed)
;  271        0395  1 !       LINE_PTR - string pointer into output line for text
;  272        0396  1 !       ULM = indicates whether result is to be in upper, lower, or mixed case.
;  273        0397  1 !
;  274        0398  1 ! IMPLICIT INPUTS:      None
;  275        0399  1 !
;  276        0400  1 ! IMPLICIT OUTPUTS:     None
;  277        0401  1 !
;  278        0402  1 ! ROUTINE VALUE:
;  279        0403  1 ! COMPLETION CODES:
;  280        0404  1 !
;  281        0405  1 !       The length of the converted string is returned
;  282        0406  1 !
;  283        0407  1 ! SIDE EFFECTS: None
;  284        0408  1 !
;  285        0409  1 !--
;  286        0410  1
;  287        0411  2     BEGIN
;  288        0412  2     LOCAL
;  289        0413  2         COUNT,
;  290        0414  2         TEXT:   VECTOR[10];
;  291        0415  2
;  292        0416  2     !Convert to Roman
;  293        0417  2     CONVBR(.VALUE,TEXT,COUNT,.ULM);
;  294        0418  2
;  295        0419  2     !Now pack it into the output line
;  296        0420  2     INCR I FROM 1 TO .COUNT DO
;  297        0421  2         CH$WCHAR_A(.TEXT[.I-1],.LINE_PTR);
;  298        0422  2
;  299        0423  2     .COUNT
;  300        0424  2
;  301        0425  1     END;                              !End of PACROM



                                   0000 00000        .ENTRY  PACROM, Save nothing                       ; 0384
                  5E           2C    C2 00002        SUBL2   #44, SP                                    ;
                         0C    AC    DD 00005        PUSHL   ULM                                        ; 0417
                         04    AE    9F 00008        PUSHAB  COUNT                                      ;
                         0C    AE    9F 0000B        PUSHAB  TEXT                                       ;
                         04    AC    DD 0000E        PUSHL   VALUE                                      ;
       00000000G  EF           04    FB 00011        CALLS   #4, CONVBR                                 ;
                               50    D4 00018        CLRL    I                                          ; 0421
                               0B    11 0001A        BRB     2$                                         ;
```

```
                                   51        08    AC  D0 0001C 1$:     MOVL    LINE_PTR, R1                                   ;
                              00   B1            6E40  F6 00020         CVTLB   TEXT=4[I], @0(R1)                              ;
                                                   61  D6 00025         INCL    (R1)                                           ;
                      F1           50              6E  F3 00027 2$:     AOBLEQ  COUNT, I, 1$                                   ;
                                   50              6E  D0 0002B         MOVL    COUNT, R0                                      ; 0425
                                                       04 0002E         RET                                                    ;

; Routine Size:  47 bytes,    Routine Base:  $CODE$ + 00BC
;  303          0426  1 %SBTTL 'Routine PACSEC'
;  304          0427  1 GLOBAL ROUTINE PACSEC (LINE_NO,STRING_PTR) =    !
;  305          0428  1
;  306          0429  1 !++
;  307          0430  1 ! FUNCTIONAL DESCRIPTION:
;  308          0431  1 !
;  309          0432  1 !       Pack section number into string.
;  310          0433  1 !
;  311          0434  1 ! FORMAL PARAMETERS:
;  312          0435  1 !
;  313          0436  1 !       LINE_NO  - Address of line number to be packed
;  314          0437  1 !       STRING_PTR - String pointer to output string.
;  315          0438  1 !
;  316          0439  1 ! IMPLICIT INPUTS:      None.
;  317          0440  1 !
;  318          0441  1 ! IMPLICIT OUTPUTS:     None
;  319          0442  1 !
;  320          0443  1 ! ROUTINE VALUE:
;  321          0444  1 ! COMPLETION CODES:
;  322          0445  1 !
;  323          0446  1 !       The number of characters generated is returned.
;  324          0447  1 !
;  325          0448  1 ! SIDE EFFECTS: None
;  326          0449  1 !
;  327          0450  1 !--
;  328          0451  1
;  329          0452  2     BEGIN
;  330          0453  2     MAP
;  331          0454  2         LINE_NO: REF BLOCK;
;  332          0455  2     LOCAL
;  333          0456  2         CHAR_COUNT;
;  334          0457  2
;  335          0458  2     CHAR_COUNT=0;
;  336          0459  2
;  337          0460  2     !Figure out which kind of section
;  338          0461  2     CASE .LINE_NO[SCT_TYP] FROM SCT_LOW TO SCT_HIGH OF
;  339          0462  2     SET
;  340          0463  2
;  341          0464  2     [SCT_CHAPT]:
;  342          0465  2             !Chapter
;  343          0466  2             CHAR_COUNT=PACXXX(.LINE_NO[SCT_NUMBER],.STRING_PTR,.LINE_NO[SCT_CHAPT_D]);
;  344          0467  2
;  345          0468  2     [SCT_INDEX]:
;  346          0469  2             !INDEX
;  347          0470  3             BEGIN
;  348          0471  3             LOCAL
;  349          0472  3                 PTR;
;  350          0473  3
;  351          0474  3             IF .LINE_NO[SCT_INDEX_D] EQL TCONVRT_LET_UPP
;  352          0475  3             THEN
;  353        U 0476  3 %IF french %THEN
;  354        U 0477  3                 PTR = CH$PTR(UPLIT('INDEX'))
;  355          0478  3 %ELSE
;  356        U 0479  3 %IF italian %THEN
;  357        U 0480  3                 PTR = CH$PTR(UPLIT('INDEX'))
;  358          0481  3 %ELSE   ! German and English are the same word here:
;  359          0482  3                 PTR = CH$PTR(UPLIT('INDEX'))
;  360      0483  3 %FI %FI
;  361      0484  3             ELSE
;  362      0485  3                 IF .LINE_NO[SCT_INDEX_D] EQL TCONVRT_LET_LOW
;  363      0486  3                 THEN
;  364    U 0487  3 %IF french %THEN
;  365    U 0488  3                     PTR = CH$PTR:UPLIT('index'))
;  366      0489  3 %ELSE
;  367    U 0490  3 %IF ital'^n %THEN
;  368    U 0491  3                     PTR = CH$PTR(UPLIT('index'))
;  369      0492  3 %ELSE   ! German and English are the same word here:
;  370      0493  3                     PTR = CH$PTR(UPLIT('index'))
;  371      0494  3 %FI %FI
;  372      0495  3                 ELSE
;  373      0496  3                 !Everything else is interpreted as "mixed"
;  374    U 0497  3 %IF french %THEN
;  375    U 0498  3                     PTR = CH$PTR(UPLIT('Index'));
;  376      0499  3 %ELSE
;  377    U 0500  3 %IF italian %THEN
;  378    U 0501  3                     PTR = CH$PTR(UPLIT('Index'));
;  379      0502  3 %ELSE   ! German and English are the same word here:
;  380      0503  3                     PTR = CH$PTR(UPLIT('Index'));
;  381      0504  3 %FI %FI
;  382      0505  3
;  383      0506  3             CHAR_COUNT=PAC3TR(   .PTR
;  384    U 0507  3 %IF french %THEN
;  385    U 0508  3                                 ,5
;  386      0509  3 %ELSE
;  387    U 0510  3 %IF italian %THEN
;  388    U 0511  3                                 ,5
;  389      0512  3 %ELSE   ! German and English are the same word here:
;  390      0513  3                                 ,5
;  391      0514  3 %FI %FI
;  392      0515  3                                 ,.STRING_PTR);
;  393      0516  2             END;
;  394      0517  2
;  395      0518  2     [SCT_APPEND]:
;  396      0519  2             !Appendix
;  397      0520  2             CHAR_COUNT=PACXXX(.LINE_NO[SCT_NUMBER],.STRING_PTR,.LINE_NO[SCT_APPEN_D]);
;  398      0521  2
;  399      0522  2     TES;
;  400      0523  2
;  401      0524  2     .CHAR_COUNT
;  402      0525  2
;  403      0526  1     END;                            !End of PACSEC



                                                              .PSECT  $PLIT$,NOWRT,NOEXE,2

                   00  00  00  58  45  44  4E  49  00000 P.AAA:   .ASCII  \INDEX\<0><0><0>                          ;
                   00  00  00  78  65  64  6E  69  00008 P.AAB:   .ASCII  \index\<0><0><0>                          ;
                   00  00  00  78  65  64  6E  49  00010 P.AAC:   .ASCII  \Index\<0><0><0>                          ;



                                                              .PSECT  $CODE$,NOWRT,2
```

```
                                          001C 00000        .ENTRY  PACSEC, Save R2,R3,R4                ; 0427
                         54 00000000'  EF   9E 00002        MOVAB   P.AAA, R4                             ;
                                       50   D4 00009        CLRL    CHAR_COUNT                            ; 0458
                         51       04   AC   D0 0000B        MOVL    LINE_NO, R1                           ; 0461
     53          61      04            00   EF 0000F        EXTZV   #0, #4, (R1), R3                      ;
                 02      01            53   CF 00014        CASEL   R3, #1, #2                            ;
               003C    000E       0006         00018 1$:    .WORD   2$-1$,-                               ;
                                                                    3$-1$,-                               ;
                                                                    7$-1$                                 ;
     7E     0C   A1      04            04   EF 0001E 2$:    EXTZV   #4, #4, 12(R1), -(SP)                 ; 0466
                                       34   11 00024        BRB     8$                                    ;
     02     0D   A1      04            04   ED 00026 3$:    CMPZV   #4, #4, 13(R1), #2                    ; 0474
                                       05   12 0002C        BNEQ    4$                                    ;
                         52            64   9E 0002E        MOVAB   P.AAA, PTR                            ; 0482
                                       12   11 00031        BRB     6$                                    ;
     03     0D   A1      04            04   ED 00033 4$:    CMPZV   #4, #4, 13(R1), #3                    ; 0485
                                       06   12 00039        BNEQ    5$                                    ;
                         52       08   A4   9E 0003B        MOVAB   P.AAB, PTR                            ; 0493
                                       04   11 0003F        BRB     6$                                    ;
                         52       10   A4   9E 00041 5$:    MOVAB   P.AAC, PTR                            ; 0503
                                  08   AC   DD 00045 6$:    PUSHL   STRING_PTR                            ; 0515
                                       05   DD 00048        PUSHL   #5                                    ; 0506
                                       52   DD 0004A        PUSHL   PTR                                   ;
              00000000V  EF            03   FB 0004C        CALLS   #3, PACSTR                            ;
                                            04 00053        RET                                           ; 0461
     7E     0D   A1      04            00   EF 00054 7$:    EXTZV   #0, #4, 13(R1), -(SP)                 ; 0520
                                  08   AC   DD 0005A 8$:    PUSHL   STRING_PTR                            ;
                                  04   A1   DD 0005D        PUSHL   4(R1)                                 ;
              00000000V  EF            03   FB 00060        CALLS   #3, PACXXX                            ;
                                            04 00067        RET                                           ; 0526

; Routine Size:  104 bytes,    Routine Base:  $CODE$ + 00EB
```

```
1 %SBTTL 'Routine PACSTR'
1 GLOBAL ROUTINE PACSTR (STG_PTR, STG_SIZE, LINE_PTR) =    !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Pack string into print line.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       STG_PTR - ch$ptr to string to be packed.
1 !       STG_SIZE - size of string to be packed.
1 !       LINE_PTR - string pointer to output print line.
1 !
1 ! IMPLICIT INPUTS:      None
1 !
1 ! IMPLICIT OUTPUTS:     None
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Number of characters placed in print line.
1 !
1 ! SIDE EFFECTS: None
1 !
1 !--
1
2     BEGIN
2     LOCAL
2         CHAR_COUNT,
2         C_PTR;
2
2     CHAR_COUNT = 0;
2     C_PTR = .STG_PTR;
2     !Now pack the string
2
2     INCR I FROM 1 TO .STG_SIZE DO
3         BEGIN
3         LOCAL
3             CHAR;
3
3         CHAR = CH$RCHAR_A (C_PTR);
3
4         IF (.CHAR LSS %C' '
4             OR .CHAR GTR %O'176')
3         THEN
3         !Substitute printables for control character
4             BEGIN
4             CH$WCHAR_A (%C'^', .LINE_PTR);
4             CHAR_COUNT = .CHAR_COUNT + 1;
4             CHAR = %C'@' + .CHAR
3             END;
3
3         CH$WCHAR_A (.CHAR, .LINE_PTR);
3         CHAR_COUNT = .CHAR_COUNT + 1
2         END;
2
```

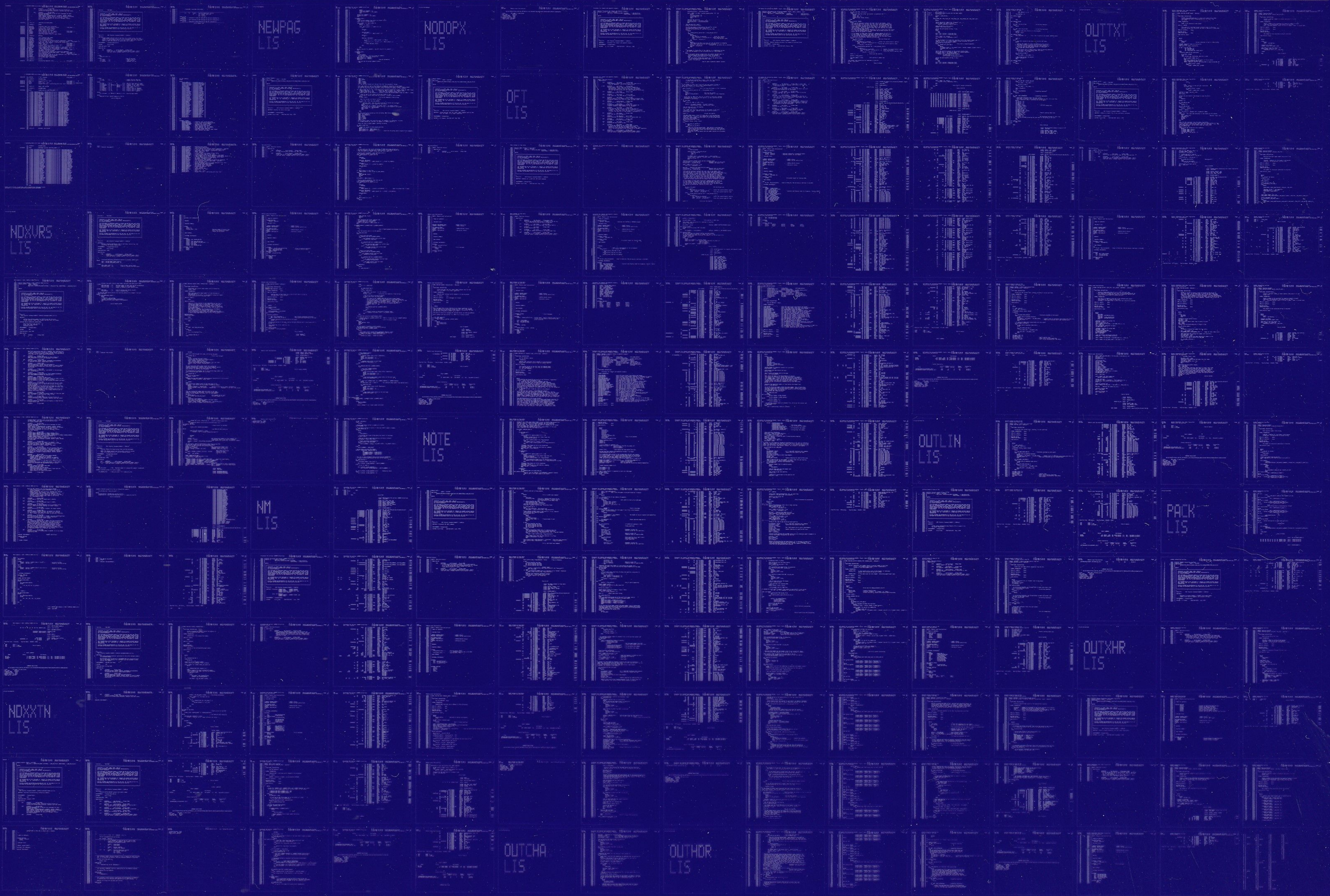
```
;  462            0584  2      .CHAR_COUNT
;  463            0585  1      END;                                       !End of PACSTR



                                         001C 00000          .ENTRY   PACSTR, Save R2,R3,R4                                  ; 0528
                                       50  D4 00002          CLRL     CHAR_COUNT                                             ; 0559
                   54        04        AC  D0 00004          MOVL     STG_PTR, C_PTR                                         ; 0560
                                       53  D4 00008          CLRL     I                                                      ; 0563
                                       2E  11 0000A          BRB      4$                                                     ;
                   52                  84  9A 0000C  1$:     MOVZBL   (C_PTR)+, CHAR                                         ; 0568
                   20                  52  D1 0000F          CMPL     CHAR, #32                                              ; 0570
                                       09  19 00012          BLSS     2$                                                     ;
         0000007E  8F                  52  D1 00014          CMPL     CHAR, #126                                             ; 0571
                                       11  15 0001B          BLEQ     3$                                                     ;
                   51        0C        AC  D0 0001D  2$:     MOVL     LINE_PTR, R1                                           ; 0575
              00   B1        5E        8F  90 00021          MOVB     #94, @0(R1)                                            ;
                                       61  D6 00026          INCL     (R1)                                                   ;
                                       50  D6 00028          INCL     CHAR_COUNT                                             ; 0576
                   52        40        A2  9E 0002A          MOVAB    64(R2), CHAR                                           ; 0577
                   51        0C        AC  D0 0002E  3$:     MOVL     LINE_PTR, R1                                           ; 0580
              00   B1                  52  90 00032          MOVB     CHAR, @0(R1)                                           ;
                                       61  D6 00036          INCL     (R1)                                                   ;
                                       50  D6 00038          INCL     CHAR_COUNT                                             ; 0581
       CD          53        08        AC  F3 0003A  4$:     AOBLEQ   STG_SIZE, I, 1$                                        ;
                                           04 0003F          RET                                                             ; 0585

; Routine Size:  64 bytes,    Routine Base:  $CODE$ + 0153
```

```
1 %SBTTL 'Routine PACXXX'
1 GLOBAL ROUTINE PACXXX (VALUE, LINE_PTR, DISPLAY_CODE) =          !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Convert a decimal number to whatever and place it in the
1 !       output text.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       VALUE - number to be converted (no negative numbers allowed)
1 !       LINE_PTR - string pointer into output line for text
1 !       DISPLAY_CODE - indicates type of conversion desired.
1 !
1 ! IMPLICIT INPUTS:      None
1 !
1 ! IMPLICIT OUTPUTS:     None
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       The length of the converted string is returned
1 !
1 ! SIDE EFFECTS: None
1 !
1 !--
1
2    BEGIN
2
2    CASE .DISPLAY_CODE FROM TCONVRT_LOW TO TCONVRT_HIGH OF
2        SET
2        [TCONVRT_DEC_NOZ, TCONVRT_DEC_ZER] :
2            PACBAS (.VALUE, .LINE_PTR, 10);
2
2        [TCONVRT_OCT_NOZ] :
2            PACBAS (.VALUE, .LINE_PTR, 8);
2
2        [TCONVRT_HEX_NOZ] :
2            PACBAS (.VALUE, .LINE_PTR, 16);
2
2        [TCONVRT_LET_UPP] :
2            PACLET (.VALUE, .LINE_PTR, -1);
2
2        [TCONVRT_LET_LOW] :
2            PACLET (.VALUE, .LINE_PTR, 0);
2
2        [TCONVRT_LET_MIX] :
2            PACLET (.VALUE, .LINE_PTR, +1);
2
2        [TCONVRT_ROM_UPP] :
2            PACROM (.VALUE, .LINE_PTR, -1);
2
2        [TCONVRT_ROM_LOW] :
2            PACROM (.VALUE, .LINE_PTR, 0);
2
2        [TCONVRT_ROM_MIX] :
```

```
;  522          0643  2             PACROM (.VALUE, .LINE_PTR, +1);
;  523          0644  2
;  524          0645  2          TES
;  525          0646  1      END;                                             .End of PACXXX



                                   0004 00000       .ENTRY  PACXXX, Save R2                       ; 0587
                         52  FE67 CF 9E 00002       MOVAB   PACBAS, R2                            :
                         50    04 AC 7D 00007       MOVQ    VALUE, R0                             ; 0619
                09       00    0C AC CF 0000B       CASEL   DISPLAY_CODE, #0, #9                  ; 0616
     0029     0024     0014     0014    00010 1$:   .WORD   2$-1$,-                               :
     003F     003B     0036     002D    00018               2$-1$,-                               :
                       001C     0018    00020               6$-1$,-                               :
                                                            7$-1$,-                               :
                                                            8$-1$,-                               :
                                                            10$-1$,-                              :
                                                            11$-1$,-                              :
                                                            12$-1$,-                              :
                                                            3$-1$,-                               :
                                                            4$-1$                                 :
                                  0A DD 00024 2$:   PUSHL   #10                                   ; 0619
                                  06 11 00026       BRB     5$                                    :
                                  08 DD 00028 3$:   PUSHL   #8                                    ; 0622
                                  02 11 0002A       BRB     5$                                    :
                                  10 DD 0002C 4$:   PUSHL   #16                                   ; 0625
                                  03 BB 0002E 5$:   PUSHR   #^M<R0,R1>                            :
                         62       03 FB 00030       CALLS   #3, PACBAS                            :
                                     04 00033       RET                                           :
                         7E       01 CE 00034 6$:   MNEGL   #1, -(SP)                             ; 0628
                                  06 11 00037       BRB     9$                                    :
                                  7E D4 00039 7$:   CLRL    -(SP)                                 ; 0631
                                  02 11 0003B       BRB     9$                                    :
                                  01 DD 0003D 8$:   PUSHL   #1                                    ; 0634
                                  03 BB 0003F 9$:   PUSHR   #^M<R0,R1>                            :
                      30 A2       03 FB 00041       CALLS   #3, PACLET                            :
                                     04 00045       RET                                           :
                         7E       01 CE 00046 10$:  MNEGL   #1, -(SP)                             ; 0637
                                  06 11 00049       BRB     13$                                   :
                                  7E D4 0004B 11$:  CLRL    -(SP)                                 ; 0640
                                  02 11 0004D       BRB     13$                                   :
                                  01 DD 0004F 12$:  PUSHL   #1                                    ; 0643
                                  03 BB 00051 13$:  PUSHR   #^M<R0,R1>                            :
                    00BC C2       03 FB 00053       CALLS   #3, PACROM                            :
                                     04 00058       RET                                           ; 0646

; Routine Size:  89 bytes,    Routine Base:  $CODE$ + 0193


;  526          0647  1
;  527          0648  1 END                                                   !End of module
;  528          0649  0 ELUDOM
;                                          PSECT SUMMARY
;
;       Name                         Bytes                          Attributes
;
;  $CODE$                                492  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                                 24  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)




;                                  Library Statistics
;
;                                                 -------- Symbols --------      Pages       Processing
;       File                                      Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]XPORT.L32;1                 590         0         0         252         00:00.2
;  _$255$DUA28:[RUNOFF.SRC]DSRLIB.L32;1           1248        26         2          86         00:00.3





;                                        COMMAND QUALIFIERS

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

; Size:          492 code + 24 data bytes
; Run Time:         00:10.3
; Elapsed Time:     00:22.8
; Lines/CPU Min:     3795
; Lexemes/CPU-Min: 12304
; Memory Used:  58 pages
; Compilation Complete
```

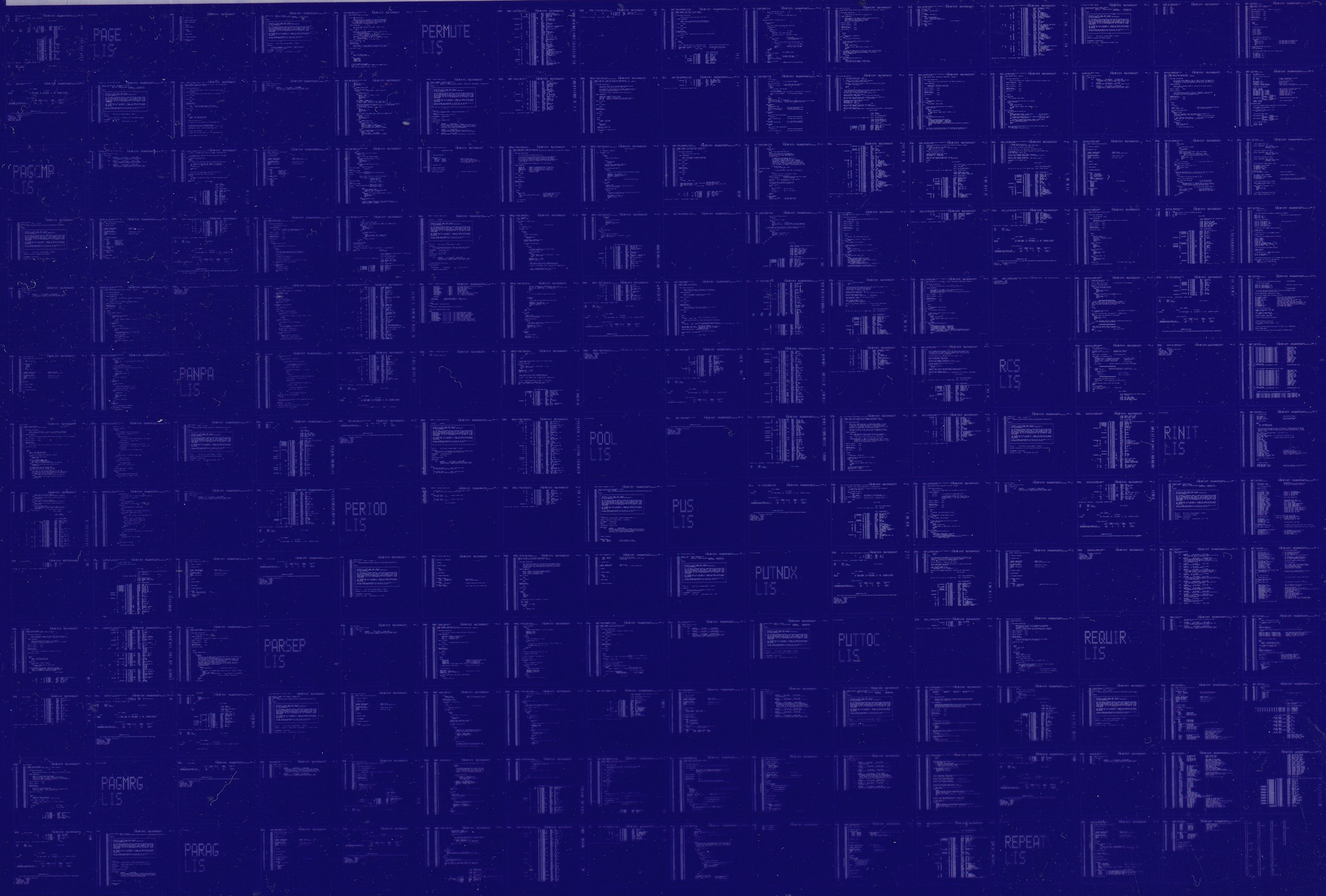
0346 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0347 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



PAGCMP
LIS

PAGE
LIS

PAGMRG
LIS

PANPA
LIS

PARAG
LIS

PARSEP
LIS

PERIOD
LIS

PERMUTE
LIS

POOL
LIS

PUS
LIS

PUTNDX
LIS

PUTTOC
LIS

RCS
LIS

REPEAT
LIS

REQUIR
LIS

RINIT
LIS