```
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
```

```
XX      XX  TTTTTTTTTT    AAAAAA    BBBBBBBB
XX      XX  TTTTTTTTTT    AAAAAA    BBBBBBBB
XX      XX      TT       AA    AA   BB      BB
XX      XX      TT       AA    AA   BB      BB
  XX  XX        TT       AA    AA   BB      BB
  XX  XX        TT       AA    AA   BB      BB
    XX          TT       AA    AA   BBBBBBBB
    XX          TT       AA    AA   BBBBBBBB
  XX  XX        TT       AAAAAAAAAA   BB      BB
  XX  XX        TT       AAAAAAAAAA   BB      BB
XX      XX      TT       AA    AA   BB      BB      ....
XX      XX      TT       AA    AA   BB      BB      ....
XX      XX      TT       AA    AA   BBBBBBBB       ....
XX      XX      TT       AA    AA   BBBBBBBB       ....

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 XTAB (                                        !
;     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 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: Expands a tab, according to tab stop settings.
;    36          0036  1 !
;    37          0037  1 ! ENVIRONMENT: Transportable
;    38          0038  1 !
;    39          0039  1 ! AUTHOR: R.W.Friday     CREATION DATE: July, 1978
;    40          0040  1 !
```

```
;    42          0041  1 %SBTTL 'Revision History'
;    43          0042  1 !
;    44          0043  1 ! MODIFIED BY:
;    45          0044  1 !
;    46          0045  1 !       006     KFA00006        Ken Alden       28-Jun-1983
;    47          0046  1 !               Tabs may now expand out to the right margin, which may
;    48          0047  1 !                 be larger than the page size.
;    49          0048  1 !
;    50          0049  1 !       005     KAD00005        Keith Dawson    07-Mar-1983
;    51          0050  1 !               Global edit of all modules. Updated module names, idents,
;    52          0051  1 !               copyright dates. Changed require files to BLISS library.
;    53          0052  1 !
;    54          0053  1 !--
```

```
;    56          0054  1 %SBTTL 'Module Level Declarations'
;    57          0055  1
;    58          0056  1 !
;    59          0057  1 ! TABLE OF CONTENTS:
;    60          0058  1 !
;    61          0059  1 !
;    62          0060  1 ! INCLUDE FILES:
;    63          0061  1 !
;    64          0062  1
;    65          0063  1 LIBRARY 'NXPORT:XPORT';                 ! XPORT Library
;    66          0064  1 REQUIRE  REQ:RNODEF';                   ! RUNOFF variant definitions
;    67          0195  1
;    68        U 0196  1 %IF DSRPLUS %THEN
;    69        U 0197  1 LIBRARY 'REQ:DPLLIB';                   ! DSRPLUS BLISS Library
;    70          0198  1 %ELSE
;    71          0199  1 LIBRARY 'REQ:DSRLIB';                   ! DSR BLISS Library
;    72          0200  1 %FI
;    73          0201  1
;    74          0202  1 !
;    75          0203  1 ! MACROS:
;    76          0204  1 !
;    77          0205  1 !
;    78          0206  1 ! EQUATED SYMBOLS:
;    79          0207  1 !
;    80          0208  1
;    81          0209  1 EXTERNAL LITERAL
;    82          0210  1     RINTES : UNSIGNED (8);
;    83          0211  1
;    84          0212  1 !
;    85          0213  1 ! OWN STORAGE:
;    86          0214  1 !
;    87          0215  1 !
;    88          0216  1 ! EXTERNAL REFERENCES:
;    89          0217  1 !
;    90          0218  1
;    91          0219  1 EXTERNAL
;    92          0220  1     mra : ref fixed_string,
;    93          0221  1     sca : sca_definition,
;    94          0222  1     tsf : tsf_definition,
;    95          0223  1     ttable : counted_list;
;    96          0224  1
;    97          0225  1 EXTERNAL ROUTINE
;    98          0226  1     endchr,     endwrd,         fcimra,         remneg;
;    99          0227  1
```

```
;  101     0228  1 GLOBAL ROUTINE xtab =
;  102     0229  1
;  103     0230  1 !++
;  104     0231  1 ! FUNCTIONAL DESCRIPTION:
;  105     0232  1 !
;  106     0233  1 !       Expands tabs.
;  107     0234  1 !
;  108     0235  1 ! FORMAL PARAMETERS:    None
;  109     0236  1 !
;  110     0237  1 ! IMPLICIT INPUTS·      None
;  111     0238  1 !
;  112     0239  1 ! IMPLICIT OUTPUTS:     None
;  113     0240  1 !
;  114     0241  1 ! ROUTINE VALUE:
;  115     0242  1 ! COMPLETION CODES:
;  116     0243  1 !
;  117     0244  1 !       Returns TRUE if the tab could be expanded, otherwise FALSE.
;  118     0245  1 !
;  119     0246  1 ! SIDE EFFECTS:
;  120     0247  1 !
;  121     0248  1 !       Generally causes a reorganization of TSF, MRA, and SCA.
;  122     0249  1 !--
;  123     0250  2     BEGIN
;  124     0251  2
;  125     0252  2     LOCAL
;  126     0253  2         current_ext_hl,
;  127     0254  2         d_ptr,
;  128     0255  2         fill_count,
;  129     0256  2         hold_cnbits,
;  130     0257  2         s_ptr;
;  131     0258  2
;  132     0259  2 !+
;  133     0260  2 ! Compute current length of line.  When doing this count trailing spaces, etc.
;  134     0261  2 ! If the SCA_rent word is not yet finished then SCA_WRD_CPEND will count as a
;  135     0262  2 ! spacing character, otherwise as a non-spacer, since it will contain RINTES.
;  136     0263  2 ! Also, detect the fact that the tab may be the very first character to be
;  137     0264  2 ! generated on the output line.
;  138     0265  2 !-
;  139     0266  2
;  140     0267  2 !+
;  141     0268  2 ! If a negative indent is pending, do not cancel it. Instead, subtract it from
;  142     0269  2 ! the left margin before beginning the tab-expansion check. (V1.124f,
;  143     0270  2 ! 21-OCT-1980, kad.)
;  144     0271  2 !-
;  145     0272  3     IF (.tsf_ext_hl EQL 0)
;  146     0273  2         AND .sca_fc
;  147     0274  2     THEN
;  148     0275  3         BEGIN
;  149     0276  3         !Any positive pending indentation is canceled because tabs are, by
;  150     0277  3         !definition, relative to the left margin and do not include
;  151     0278  3         !indentation.
;  152     0279  3         IF .sca_indent GTR 0
;  153     0280  3          THEN
;  154     0281  3              sca_indent = 0;
;  155     0282  3         !Subtract the indent if it is negative.
;  156     0283  3         current_ext_hl = .sca_lm + .sca_indent;
;  157     0284  3         END
```

```
;  158     0285  2      ELSE
;  159     0286  2          current_ext_hl = .tsf_ext_hl;
;  160     0287  2
;  161     0288  2      IF .current_ext_hl LSS 0
;  162     0289  2      THEN
;  163     0290  3          BEGIN
;  164     0291  3          remneg ();                              !Attempted negative indent.
;  165     0292  3          current_ext_hl = 0;
;  166     0293  2          END;
;  167     0294  2
;  168     0295  2      current_ext_hl = .current_ext_hl + .sca_wrd_ext_l + .sca_wrd_lst_sp;
;  169     0296  2
;  170     0297  2      IF .sca_wrd_cpend GEQ %C' '
;  171     0298  2      THEN
;  172     0299  2          current_ext_hl = .current_ext_hl + 1;
;  173     0300  2
;  174     0301  2      INCR i FROM 1 TO .ttable [cl_index] DO
;  175     0302  3          BEGIN
;  176     0303  3          IF .ttable [.i] GTR .current_ext_hl
;  177     0304  3          THEN
;  178     0305  4              BEGIN                               !Found a suitable tab setting.
;  179     0306  4      !+
;  180     0307  4      ! Compute number of spaces needed to be added to line. Do not allow user to tab
;  181     0308  4      ! past the right edge of his page. If the user is in a literal, the right margin
;  182     0309  4      ! will be larger than the page-width. In this case, allow him to expand tabs out
;  183     0310  4      ! to the right margin. (V1.124f, 21-OCT-1980, kad.)
;  184     0311  4      !-
;  185     0312  4              IF .ttable [.i] GEQ .sca_rm
;  186     0313  4              THEN
;  187     0314  4                  RETURN false
;  188     0315  4              ELSE
;  189     0316  4                  fill_count = .ttable [.i] - .current_ext_hl;
;  190     0317  4
;  191     0318  4      !+
;  192     0319  4      ! At this point we are certain that this tab can be expanded. All spacing
;  193     0320  4      ! characters have been counted.  Now force the current word (if there is one)
;  194     0321  4      ! to be ended.  That resets lots of counters, and in particular, forces the
;  195     0322  4      ! final pending character into MRA.
;  196     0323  4      !-
;  197     0324  4              endwrd (false, false, false);       !Anticipate no spaces or justification.
;  198     0325  4
;  199     0326  4              !Now remove all justification marks from TSF.
;  200     0327  4              !This is done by scanning the entire MRA, where the line is
;  201     0328  4              !being built up.
;  202     0329  4              s_ptr = .fs_start (mra);
;  203     0330  4              d_ptr = .s_ptr;
;  204     0331  4
;  205     0332  4              INCR i FROM 1 TO .tsf_int_hl + .tsf_int_vl DO
;  206     0333  5                  BEGIN
;  207     0334  5                  LOCAL
;  208     0335  5                      x,
;  209     0336  5                      y;
;  210     0337  5
;  211     0338  5                  x = CH$RCHAR_A (s_ptr);
;  212     0339  5
;  213     0340  5                  IF .x EQL rintes
;  214     0341  5                  THEN
```

```
;  215     0342  6                    BEGIN
;  216     0343  6                    !See if this is a justification mark.
;  217     0344  6                    y = CH$RCHAR_A (s_ptr);
;  218     0345  6
;  219     0346  6                    IF .y EQL %C'J'
;  220     0347  6                    THEN
;  221     0348  7                        BEGIN                       !Throw away the justification mark
;  222     0349  7                        s_ptr = CH$PLUS (.s_ptr, 1);
;  223     0350  7                        i = .i + 2;
;  224     0351  7                        END
;  225     0352  6                    ELSE
;  226     0353  7                        BEGIN                       !Save this escape sequence
;  227     0354  7                        CH$WCHAR_A (rintes, d_ptr);
;  228     0355  7                        CH$WCHAR_A (.y, d_ptr);
;  229     0356  7                        CH$WCHAR_A (CH$RCHAR_A (s_ptr), d_ptr);
;  230     0357  7                        i = .i + 2;
;  231     0358  7                        END
;  232     0359  6                    END
;  233     0360  5                ELSE
;  234     0361  5                    CH$WCHAR_A (.x, d_ptr);     !Move a "normal" character
;  235     0362  4                END;                            !End of compression.
;  236     0363  4
;  237     0364  4            !Update TSF and SCA
;  238     0365  4            tsf_jus_cnt = 0;
;  239     0366  4            sca_wrd_pntr = .d_ptr;
;  240     0367  4            !Update MRA information
;  241     0368  4            fs_next (mra) = .d_ptr;
;  242     0369  4            fs_length (mra) = ch$diff (.d_ptr, .fs_start (mra));
;  243     0370  4            tsf_int_hl = .fs_length (mra) - .tsf_int_vl;
;  244     0371  4 !
;  245     0372  4 !+
;  246     0373  4 ! Now put the required number of spaces onto the line. The spaces are
;  247     0374  4 ! inserted in such a way that they become part of the word that will later be
;  248     0375  4 ! built.  The assumption being made is that the next word will fit on the
;  249     0376  4 ! line being put together.  If it does not, then the word with the expanded
;  250     0377  4 ! tab prefixed to it will appear on the following line.  However, this will
;  251     0378  4 ! probably not happen, since RUNOFF users will use tabs to set up columns,
;  252     0379  4 ! and will have counted carefully. Another reason for going through the
;  253     0380  4 ! standard character processing is that the tab may be the first thing to be
;  254     0381  4 ! output.  In that case, it is important that certain information that
;  255     0382  4 ! appears before the first character be added in proper sequence. When
;  256     0383  4 ! outputting spaces, turn off underlining, bolding, etc.  This avoids the
;  257     0384  4 ! problem of having input such as ^&A     B\& generate A_____B.
;  258     0385  4 !-
;  259     0386  4            hold_cnbits = .sca_wrd_cnLits;
;  260     0387  4            sca_wrd_acnbits = false;
;  261     0388  4
;  262     0389  4            INCR i FROM 1 TO .fill_count DO
;  263     0390  5                BEGIN
;  264     0391  5                sca_wrd_cnbits = false;
;  265     0392  5                ENDCHR (%C' ');
;  266     0393  4                END;
;  267     0394  4            sca_wrd_cnbits = .hold_cnbits;
;  268     0395  4            RETURN TRUE;                    !Successful tab expansion.
;  269     0396  3            END;                            !End of processing of an applicable tab
;  270     0397  3
;  271     0398  2        END;
```

```
;  272          0399  2
;  273          0400  2      RETURN FALSE;                   !Falling through means no suitable tab setting was found.
;  274          0401  1      END;                                        !End of XTAB


                                                                .TITLE  XTAB
                                                                .IDENT  \V04-000\

                                                                .EXTRN  RINTES, MRA, SCA
                                                                .EXTRN  TSF, TTABLE, ENDCHR
                                                                .EXTRN  ENDWRD, FCIMRA, REMNEG

                                                                .PSECT  $CODE$,NOWRT,2

                                                    0FFC 00000  .ENTRY  XTAB, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 ; 0228
                                                 5E         04  C2 00002  SUBL2  #4, SP                              ;
                                          50 00000000G      EF  D0 00005  MOVL   TSF, R0                             ; 0272
                                                   04       A0  D5 0000C  TSTL   4(R0)                               ;
                                                            23  12 0000F  BNEQ   2$                                  ;
                                          1C 00000000G      EF  E9 00011  BLBC   SCA+148, 2$                         ; 0273
                                             00000000G      EF  D5 00018  TSTL   SCA+220                             ; 0279
                                                            06  15 0001E  BLEQ   1$                                  ;
                                             00000000G      EF  D4 00020  CLRL   SCA+220                             ; 0281
                                52 00000000G  FF 00000000G  EF  C1 00026  1$:    ADDL3  SCA+220, @SCA+116, CURRENT_EXT_HL ; 0283
                                                            04  11 00032  BRB    3$                                  ; 0272
                                             52         04  A0  D0 00034  2$:    MOVL   4(R0), CURRENT_EXT_HL        ; 0286
                                                            09  18 00038  3$:    BGEQ   4$                           ; 0288
                                   00000000G  EF             00  FB 0003A  CALLS  #0, REMNEG                       ; 0291
                                                            52  D4 00041  CLRL   CURRENT_EXT_HL                    ; 0292
                                50          52 00000000G  EF  C1 00043  4$:    ADDL3  SCA+256, CURRENT_EXT_HL, R0  ; 0295
                                52          50 00000000G  EF  C1 0004B  ADDL3  SCA+332, R0, CURRENT_EXT_HL         ;
                                            20 00000000G  EF  D1 00053  CMPL   SCA+280, #32                         ; 0297
                                                            02  19 0005A  BLSS   5$                                  ;
                                                            52  D6 0005C  INCL   CURRENT_EXT_HL                    ; 0299
                                            5A 00000000G    EF  D0 0005E  5$:    MOVL   TTABLE+4, R10                ; 0301
                                                            54  D4 00065  CLRL   I                                  ; 0303
                                                         00BF   31 00067  6$:    BRW    16$                          ;
                                            50 00000000GEF  44  D0 0006A  7$:    MOVL   TTABLE+4[I], R0              ;
                                            52              50  D1 00072  CMPL   R0, CURRENT_EXT_HL                 ;
                                                            F0  15 00075  BLEQ   6$                                 ;
                                   00000000G  FF            50  D1 00077  CMPL   R0, @SCA+120                      ; 0312
                                                            03  19 0007E  BLSS   8$                                  ;
                                                         00AC   31 00080  BRW    17$                                 ;
                                6E          50              52  C3 00083  8$:    SUBL3  CURRENT_EXT_HL, R0, FILL_COUNT ; 0316
                                                            7E  7C 00087  CLRQ   -(SP)                              ; 0324
                                                            7E  D4 00089  CLRL   -(SP)                              ;
                                   00000000G  EF            03  FB 0008B  CALLS  #3, ENDWRD                         ;
                                            50 00000000G    EF  D0 00092  MOVL   MRA, R0                            ; 0329
                                            57              60  D0 00099  MOVL   (R0), S_PTR                       ;
                                            53              57  D0 0009C  MOVL   S_PTR, D_PTR                      ; 0330
                                            51 00000000G    EF  D0 0009F  MOVL   TSF, R1                            ; 0332
                                59          61        18    A1  C1 000A6  ADDL3  24(R1), (R1), R9                   ;
                                                            56  D4 000AB  CLRL   I                                  ;
                                                            2E  11 000AD  BRB    13$                                ;
                                            55              87  9A 000AF  9$:    MOVZBL (S_PTR)+, X                  ; 0338
                                   00000000C  8F            55  D1 000B2  CMPL   X, #RINTES                        ; 0340
                                                            1F  12 000B9  BNEQ   12$                                ;
```

```
                                58              87  9A 000BB          MOVZBL  (S_PTR)+, Y                             ; 0344
                          0000004A  8F          58  D1 000BE          CMPL    Y, #74                                  ; 0346
                                                04  12 000C5          BNEQ    10$                                     ;
                                                57  D6 000C7          INCL    S_PTR                                   ; 0349
                                                0A  11 000C9          BRB     1T$                                     ; 0346
                                83        00G   8F  90 000CB 10$:     MOVB    #RINTES, (D_PTR)+                       ; 0354
                                83              58  90 000CF          MOVB    Y, (D_PTR)+                             ; 0355
                                83              87  90 000D2          MOVB    (S_PTR)+, (D_PTR)+                      ; 0356
                                56              02  C0 000D5 11$:     ADDL2   #2, I                                   ; 0350
                                                03  11 000D8          BRB     13$                                     ; 0361
                                83              55  90 000DA 12$:     MOVB    X, (D_PTR)+                             ;
                             CE                 59  F3 000DD 13$:     AOBLEQ  R9, I, 9$                               ; 0332
                                          20    A1  D4 000E1          CLRL    32(R1)                                  ; 0365
                          00000000G  EF         53  D0 000E4          MOVL    D_PTR, SCA+248                          ; 0366
                                04  A0          53  D0 000EB          MOVL    D_PTR, 4(R0)                            ; 0368
                          0C    A0              60  C3 000EF          SUBL3   (R0), D_PTR, 12(R0)                     ; 0369
                          61    0C  A0    18    A1  C3 000F4          SUBL3   24(R1), 12(R0), (R1)                    ; 0370
                                5B  00000000G   EF  D0 000FA          MOVL    SCA+196, HOLD_CNBITS                    ; 0386
                                    00000000G   EF  D4 00101          CLRL    SCA+200                                 ; 0387
                                                55  D4 00107          CLRL    I                                       ; 0389
                                                0F  11 00109          BRB     15$                                     ;
                                    00000000G   EF  D4 0010B 14$:     CLRL    SCA+196                                 ; 0391
                                                20  DD 00111          PUSHL   #32                                     ; 0392
                          00000000G  EF         01  FB 00113          CALLS   #1, ENDCHR                              ;
                             ED                 55  F3 0011A 15$:     AOBLEQ  FILL_COUNT, I, 14$                      ; 0389
                          00000000G  EF         5B  D0 0011E          MOVL    HOLD_CNBITS, SCA+196                    ; 0394
                                50              01  D0 00125          MOVL    #1, R0                                  ; 039:
                                                    04 00128          RET                                             ;
          FF3B                  54              01  5A  F1 00129 16$:     ACBL    R10, #1, I, 7$                  ; 0301
                                                50  D4 0012F 17$:     CLRL    R0                                      ; 0401
                                                    04 00131          RET                                             ;

; Routine Size:  306 bytes,    Routine Base:  $CODE$ + 0000

;  275          0402  1
;  276          0403  1 END                                        !End of module
;  277          0404  0 ELUDOM



;                                  PSECT SUMMARY
;
;       Name                     Bytes                          Attributes
;
;  $CODE$                          306  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                                Library Statistics
;
;                                          -------- Symbols --------      Pages       Processing
;       File                               Total    Loaded   Percent      Mapped      Time
```

```
;
;  _$255$DUA28:[SYSLIB]XPORT.L32;1              590        0        0      252          00:00.1
;  _$255$DUA28:[RUNOFF.SRC]DSRLIB.L32;1        1248       29        2       86          00:00.3
```

```
;                                       COMMAND QUALIFIERS

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

; Size:         306 code + 0 data bytes
; Run Time:        00:07.6
; Elapsed Time:    00:20.7
; Lines/CPU Min:    3189
; Lexemes/CPU-Min: 13689
; Memory Used:  97 pages
; Compilation Complete
```

0350 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

XCASE
LIS

SDADEF
SDL

RMSDEF
MAR

UNPUS
LIS

SDATEST
MAP

SDA

SDA
MAP

DCLDEF
MAR

SCSDEF
MAR

TPR
LIS

XTAB
LIS

IPROBE
LIS

SYSDEF
MAR

IMGDEF
MAR

NETDEF
MAR

VAXOPS
REQ

MACROS
MAR

CLUSTER
LIS