```
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
```

```
EEEEEEEEEE  NN      NN  DDDDDDDD    WW      WW  RRRRRRRR    DDDDDDDD
EEEEEEEEEE  NN      NN  DDDDDDDD    WW      WW  RRRRRRRR    DDDDDDDD
EE          NN      NN  DD      DD  WW      WW  RR      RR  DD      DD
EE          NN      NN  DD      DD  WW      WW  RR      RR  DD      DD
EE          NNNN    NN  DD      DD  WW      WW  RR      RR  DD      DD
EE          NNNN    NN  DD      DD  WW      WW  RR      RR  DD      DD
EEEEEEEE    NN  NN  NN  DD      DD  WW      WW  RRRRRRRR    DD      DD
EEEEEEEE    NN  NN  NN  DD      DD  WW      WW  RRRRRRRR    DD      DD
EE          NN    NNNN  DD      DD  WW  WW  WW  RR  RR      DD      DD
EE          NN    NNNN  DD      DD  WW  WW  WW  RR  RR      DD      DD
EE          NN      NN  DD      DD  WWWW  WWWW  RR    RR    DD      DD      ....
EE          NN      NN  DD      DD  WWWW  WWWW  RR    RR    DD      DD      ....
EEEEEEEEEE  NN      NN  DDDDDDDD    WW      WW  RR      RR  DDDDDDDD        ....
EEEEEEEEEE  NN      NN  DDDDDDDD    WW      WW  RR      RR  DDDDDDDD        ....

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
```

```
0 %TITLE 'Perform end-word processing'
0 MODULE endwrd ( IDENT = 'V04-000'
P 0 %BLISS32[, ADDRESSING_MODE (EXTERNAL    = LONG_RELATIVE,
0                               NONEXTERNAL = LONG_RELATIVE)]
0                ) =
1 BEGIN
1 !
1 !*****************************************************************************
1 !*                                                                           *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
1 !*  ALL RIGHTS RESERVED.                                                     *
1 !*                                                                           *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
1 !*  TRANSFERRED.                                                             *
1 !*                                                                           *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
1 !*  CORPORATION.                                                             *
1 !*                                                                           *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
1 !*                                                                           *
1 !*                                                                           *
1 !*****************************************************************************
1 !
1 !++
1 ! FACILITY:     DSR (Digital Standard RUNOFF) / DSRPLUS
1 !
1 ! ABSTRACT:     Handles end of word processing.
1 !
1 ! ENVIRONMENT:  Transportable
1 !
1 ! AUTHOR:       R.W.Friday       CREATION DATE: May, 1978
1 !
1
```

```
;  42    0041  1 %SBTTL 'Revision History'
;  43    0042  1 !
;  44    0043  1 ! MODIFIED BY:
;  45    0044  1 !
;  46    0045  1 !       009     REM00009        Ray Marshall    7-November-1983
;  47    0046  1 !               Added logic to reset the multinational character set case
;  48    0047  1 !                 rules, too.
;  49    0048  1 !
;  50    0049  1 !       008     KFA00008        Ken Alden       13-Sep-1983
;  51    0050  1 !               Added functionality for pass-through escape sequences.
;  52    0051  1 !
;  53    0052  1 !       007     RER00007        Ron Randall     07-Mar-1983
;  54    0053  1 !               Global edit of all modules. Updated module names, idents,
;  55    0054  1 !               copyright dates. Changed require files to BLISS library.
;  56    0055  1 !
;  57    0056  1 !--
;  58    0057  1
```

```
;  60          0058  1 %SBTTL 'Module Level Declarations'
;  61          0059  1
;  62          0060  1 !
;  63          0061  1 ! TABLE OF CONTENTS:
;  64          0062  1 !
;  65          0063  1 !
;  66          0064  1 ! INCLUDE FILES:
;  67          0065  1 !
;  68          0066  1 LIBRARY 'NXPORT:XPORT';                    ! XPORT Library
;  69          0067  1 REQUIRE 'REQ:RNODEF';                      ! RUNOFF variant definitions
;  70          0198  1
;  71        U 0199  1 %IF DSRPLUS %THEN
;  72        U 0200  1 LIBRARY 'REQ:DPLLIB';                     ! DSRPLUS BLISS Library
;  73          0201  1 %ELSE
;  74          0202  1 LIBRARY 'REQ:DSRLIB';                      ! DSR BLISS Library
;  75          0203  1 %FI
;  76          0204  1
;  77          0205  1 !
;  78          0206  1 ! EXTERNAL REFERENCES:
;  79          0207  1 !
;  80          0208  1 EXTERNAL LITERAL
;  81          0209  1     RINTES : UNSIGNED (8);
;  82          0210  1
;  83          0211  1 EXTERNAL
;  84          0212  1     GCA : GCA_DEFINITION,
;  85          0213  1     IRAC : IRAC_DEFINITION,
;  86          0214  1     MRA : REF FIXED_STRING,
;  87          0215  1     SCA : SCA_DEFINITION,
;  88          0216  1     TSF : TSF_DEFINITION;
;  89          0217  1
;  90          0218  1 EXTERNAL ROUTINE
;  91          0219  1     ENDCHR,
;  92          0220  1     OUTLIN,
;  93          0221  1     OUTXPH;
;  94          0222  1
```

```
;  96          0223  1 GLOBAL ROUTINE ENDWRD (SPACE, JUSTIFY, HYPHENATE) : NOVALUE =
;  97          0224  1
;  98          0225  1 !++
;  99          0226  1 ! FUNCTIONAL DESCRIPTION:
; 100          0227  1 !
; 101          0228  1 !       ENDWRD is called when the end of a word is detected.
; 102          0229  1 !       If the word would fit on the current line it puts it
; 103          0230  1 !       there, otherwise it outputs the current line before
; 104          0231  1 !       putting the current word on a new line.
; 105          0232  1 !
; 106          0233  1 ! FORMAL PARAMETERS:
; 107          0234  1 !
; 108          0235  1 !       SPACE indicates whether a space should be output following
; 109          0236  1 !       the word.
; 110          0237  1 !       JUSTIFY indicates whether a justification mark should be output
; 111          0238  1 !       following the word.
; 112          0239  1 !
; 113          0240  1 ! IMPLICIT INPUTS:      None
; 114          0241  1 !
; 115          0242  1 ! IMPLICIT OUTPUTS:     None
; 116          0243  1 !
; 117          0244  1 ! ROUTINE VALUE:
; 118          0245  1 ! COMPLETION CODES:     None
; 119          0246  1 !
; 120          0247  1 ! SIDE EFFECTS:         None
; 121          0248  1 !--
; 122          0249  1
; 123          0250  2     BEGIN
; 124          0251  2     LOCAL
; 125          0252  2         ADJUST;
; 126          0253  2
; 127          0254  2     !Check for an active <INDEX flag> immediately, otherwise the
; 128          0255  2     !index entry could get skipped.
; 129          0256  2
; 130          0257  2     IF  .SCA_X_FLAG
; 131          0258  2     THEN
; 132          0259  2         OUTXPH ();                          !The <INDEX flag> is active, output the indexed word.
; 133          0260  2
; 134          0261  3     IF  (.SCA_WRD_CPEND EQL RINTES)
; 135          0262  3         AND (.SCA_WRD_F_XTN EQL 0)          !Maybe just an indexing pointer
; 136          0263  3         AND (.SCA_WRD_FOOTW EQL 0)          !Maybe just a footnote
; 137          0264  2         AND
; 138          0265  3         (( NOT .SCA_FILL) AND (.SCA_WRD_LST_SP EQL 0))
; 139        U 0266  3 %IF DSRPLUS %THEN
; 140        U 0267  3         AND (NOT .sca_wrd_pass)
; 141          0268  3 %FI
; 142          0269  2     THEN
; 143          0270  2         RETURN                              !Nothing to put onto the line.
; 144          0271  2     ELSE
; 145          0272  2         ENDCHR (RINTES);                    !Force out last character of word.
; 146          0273  2
; 147        U 0274  2 %IF DSRPLUS %THEN
; 148        U 0275  2     sca_wrd_pass = false;
; 149          0276  2 %FI
; 150          0277  2     sca_frc_case = false;                   !Turn off case forcing.
; 151          0278  2
; 152          0279  2     !Compute how long line would be if hyphenation is to be
```

```
0280  2     !considered.  In the call on ENDWRD for the previous word,
0281  2     !hyphenation information for it was left behind (if there was
0282  2     !any requested.  What happens now is to see what would happen
0283  2     !if we tried to force this word on at the end of this line.
0284  3     ADJUST = (IF  .SCA_WRD_LST_HYP
0285  3               THEN
0286  3                 -1
0287  3               ELSE
0288  2                 0)     +
0289  3              (IF .HYPHENATE
0290  3               THEN
0291  3                 1
0292  3               ELSE
0293  2                 0);
0294  2
0295  2     !The OR test detects the case when MRA is about to
0296  2     !overflow, but the word would still fit.  If this test is
0297  2     !not done, ENDCHR could go into a loop.
0298  2
0299  3     IF  ((.TSF_EXT_HL + .SCA_WRD_LST_SP + .SCA_WRD_EXT_L + .ADJUST) GTR .SCA_RM)
0300  3         OR ((.FS_MAXSIZE (MRA) - .FS_LENGTH (MRA)) LSS 10)
0301  2     THEN
0302  3         BEGIN                                   !There is no room on this line
0303  3         OUTLIN (.SCA_JUSTIFY);
0304  3         END
0305  2     ELSE
0306  3         BEGIN
0307  3         !First, put trailing spaces and justification marks from the
0308  3         !previous word in the TSF into the TSF.  The information was
0309  3         !already there (i.e., the characters, or intermediate code), but
0310  3         !it was not yet counted as part of TSF.
0311  3         !Also, since there is room for this word, turn off any
0312  3         !hyphenation that may be queued on the line already.
0313  3         IF .SCA_WRD_LST_HYP
0314  3         THEN
0315  4             BEGIN
0316  4             TSF_EXT_HL = .TSF_EXT_HL - 1;        !Cancel the '-'.
0317  4             CH$WCHAR (%C'N', .SCA_WRD_HYP_PTR);
0318  3             END;
0319  3
0320  3         TSF_INT_HL = .TSF_INT_HL + .SCA_WRD_LST_SP       !count of trailing spaces
0321  3         + .SCA_WRD_LST_UND*3                     !Count underlined spaces code.
0322  3         + 3*.SCA_WRD_LST_JUS;                    !size of justification mark, if any.
0323  3         TSF_EXT_HL = .TSF_EXT_HL + .SCA_WRD_LST_SP;      !count trailing spaces here also
0324  3         TSF_JUS_CNT = .TSF_JUS_CNT + .SCA_WRD_LST_JUS;  !Count number of words.
0325  2         END;
0326  2
0327  2     ! There are some rather obscure implications as a consequence of what
0328  2     ! is happening here.  In part, this is because of possible error handling.
0329  2     !   Notice that at this point ENDWRD just assumes that the current word
0330  2     !   will fit on the line, and goes ahead and updates the various
0331  2     !   pointers; yet, if you look around, you will find no checks that
0332  2     !   verify that the word will, indeed, fit onto the line between
0333  2     !   the margins.  That checking is done by OUTLIN.  The idea is that
0334  2     !   if a word is too big, then the line can be forced out anyway,
0335  2     !   even if the right margin is violated.  Another point has to do
0336  2     !   with the logic in ENDCHR, when the MRA is about to overflow.
```

```
;  210          0337  2      !   In such a case, it is important that the characters still go
;  211          0338  2      !   through, because of the recursive call ENDCHR makes on itself.
;  212          0339  2      !
;  213          0340  2      !Next, make this current word part of TSF
;  214          0341  2      TSF_NBITS = (.TSF_NBITS OR .SCA_WRD_NBITS); ! Special function bits (e.g, bolding)
;  215          0342  2      TSF_UND = .TSF_UND OR (.SCA_WRD_LST_UND GTR 0);
;  216          0343  2      TSF_H_BARS = (.TSF_H_BARS OR .SCA_WRD_BARS);        !Change bar status.
;  217          0344  2      TSF_BAR_CHAR = .SCA_WRD_BAR_CHR;            !Use latest change bar character.
;  218          0345  2      TSF_SEQN_FLAG = .SCA_WRD_SEQN_F;            !Input sequence number information.
;  219          0346  2      TSF_ISEQ = .GCA_CMD_ISQ;
;  220          0347  2      TSF_ISEQN = .SCA_WRD_ISEQN;
;  221          0348  2      TSF_IPAGEN = .SCA_WRD_IPAGEN;
;  222          0349  2
;  223          0350  2      !If footnote are attached to this word, attach them
;  224          0351  2      !to this line.
;  225          0352  2      TSF_FOOTW = .TSF_FOOTW + .SCA_WRD_FOOTW;            !Add footnote count total for this word to that for
;  226          0353  2      SCA_WRD_FOOTW = 0;                                  !Now forget it so it won't get counted twice.
;  227          0354  2
;  228          0355  2      !Consider hyphenation.
;  229          0356  2      IF  .HYPHENATE
;  230          0357  2      THEN
;  231          0358  3          BEGIN
;  232          0359  3          TSF_INT_HL = .TSF_INT_HL + 3;           !NOTE: The '-' has not yet
;  233          0360  3          TSF_EXT_HL = .TSF_EXT_HL + 1;           !been put into the MRA!!
;  234          0361  2          END;
;  235          0362  2
;  236          0363  2      TSF_DRAFT_FLAG = .SCA_WRD_DRAFT_F;          !Indicate if a DRAFT document.
;  237          0364  2      TSF_INT_HL = (.TSF_INT_HL + .SCA_WRD_INT_L);
;  238          0365  2      TSF_EXT_HL = (.TSF_EXT_HL + .SCA_WRD_EXT_L);
;  239          0366  2
;  240          0367  3      IF  (.TSF_FIRST_XTN EQL 0)          !If an index entry is associated with this word
;  241          0368  2      THEN                                !attach it to this line.
;  242          0369  2          TSF_FIRST_XTN = .SCA_WRD_F_XTN;
;  243          0370  2
;  244          0371  3      IF  (.SCA_WRD_L_XTN NEQ 0)
;  245          0372  2      THEN
;  246          0373  2          TSF_LAST_XTN = .SCA_WRD_L_XTN;
;  247          0374  2
;  248          0375  2      !Wipe out leftovers from previous word
;  249          0376  2      SCA_WRD_LST_UND = 0;
;  250          0377  2      SCA_WRD_LST_SP = 0;
;  251          0378  2      SCA_WRD_LST_JUS = 0;
;  252          0379  2      SCA_WRD_LST_HYP = FALSE;
;  253          0380  2      SCA_WRD_HYP_PTR = 0;
;  254          0381  2
;  255          0382  2      !Next, put the trailing spaces and justification marks
;  256          0383  2      !in, but don't actually make them part of the TSF.
;  257          0384  2      IF  .SCA_WRD_INT_L NEQ 0
;  258          0385  2      THEN
;  259          0386  3          BEGIN                                   !Processing associated with a non-null word.
;  260          0387  3
;  261          0388  3          IF .SCA_WRD_AC_UND OR .SPACE
;  262          0389  3          THEN
;  263          0390  4              BEGIN
;  264          0391  4
;  265          0392  4              IF  .SCA_WRD_AC_UND
;  266          0393  4              THEN
```

```
0394  5                 BEGIN                                  !It's an underlined space!
0395  5                 SCA_WRD_LST_UND = 1;
0396  5                 FS_QCHAR (MRA, RINTES);                !Generate the intermediate code.
0397  5                 FS_WCHAR (MRA, %C'U');
0398  5                 FS_WCHAR (MRA, %C' ');
0399  5                 SCA_WRD_AC_UND = FALSE;
0400  4                 END;
0401  4
0402  4             FS_WCHAR (MRA, %C' ');                     !Add a single trailing space,
0403  4             SCA_WRD_LST_SP = 1;                        !and count it.
0404  3             END;
0405  3
0406  3         IF .SCA_WRD_LC_PNCT
0407  3             AND .SPACE
0408  3         THEN
0409  4             BEGIN                                      !Add an extra space for punctuation.
0410  4             FS_WCHAR (MRA, %C' ');
0411  4             SCA_WRD_LST_SP = .SCA_WRD_LST_SP + 1;
0412  4             SCA_WRD_LC_PNCT = FALSE;
0413  3             END;
0414  3
0415  3         !Next, add a justification point, if appropriate.
0416  3         IF .JUSTIFY
0417  3         THEN
0418  4             BEGIN
0419  4             SCA_WRD_LST_JUS = 1;                       !Indicate one justification mark
0420  4             FS_QCHAR (MRA, RINTES);
0421  4             FS_WCHAR (MRA, %C'J');
0422  4             FS_WCHAR (MRA, %C' ');
0423  3             END;
0424  3
0425  3         !Put in code for hyphenation, if requested.
0426  3         IF .HYPHENATE
0427  3         THEN
0428  4             BEGIN
0429  4             SCA_WRD_LST_HYP = TRUE;
0430  4             FS_QCHAR (MRA, RINTES);
0431  4             SCA_WRD_HYP_PTR = .FS_NEXT (MRA);          !Remember where '-' is.
0432  4             FS_QCHAR (MRA, %C'I');                     !INSERT this character.
0433  4             FS_WCHAR (MRA, %C'-');                     !Here's the '-'.
0434  3             END;
0435  3
0436  2         END;                                           !End of processing of a non-null word
0437  2
0438  2     !Next, reset for the next word
0439  2     SCA_WRD_NBITS = FALSE;                             !Clear bolding, underlining, and overstriking
0440  2     SCA_WRD_INT_L = 0;
0441  2     SCA_WRD_EXT_L = 0;
0442  2     SCA_WRD_F_XTN = 0;
0443  2     SCA_WRD_L_XTN = 0;
0444  2     SCA_WRD_BARS = FALSE;
0445  2
0446  2     !Reset case rules.
0447  2     SCA_WORD_SET = FALSE;                              !Turn off <CAPITALIZE flag>
0448  2     SCA_FC_UT = .SCA_FCBE_UT;
0449  2     SCA_OC_UT = .SCA_OCBE_UT;
0450  2     SCA_FC_LT = .SCA_FCBE_LT;
```

```
;  324          0451  2     SCA_OC_LT = .SCA_OCBE_LT;
;  325          0452  2     SCA_WRD_FC_UT = .SCA_FCBE_UT;
;  326          0453  2     SCA_WRD_OC_UT = .SCA_OCBE_UT;
;  327          0454  2     SCA_WRD_FC_LT = .SCA_FCBE_LT;
;  328          0455  2     SCA_WRD_OC_LT = .SCA_OCBE_LT;
;  329          0456  2     SCA_MNFC_UT = .SCA_MNFCBE_UT;
;  330          0457  2     SCA_MNOC_UT = .SCA_MNOCBE_UT;
;  331          0458  2     SCA_MNFC_LT = .SCA_MNFCBE_LT;
;  332          0459  2     SCA_MNOC_LT = .SCA_MNOCBE_LT;
;  333          0460  2     SCA_MNWRD_FC_UT = .SCA_MNFCBE_UT;
;  334          0461  2     SCA_MNWRD_OC_UT = .SCA_MNOCBE_UT;
;  335          0462  2     SCA_MNWRD_FC_LT = .SCA_MNFCBE_LT;
;  336          0463  2     SCA_MNWRD_OC_LT = .SCA_MNOCBE_LT;
;  337          0464  2
;  338          0465  2     !Remember where next word will start.  (Needed by SCL.)
;  339          0466  2     SCA_WRD_PNTR = .FS_NEXT (MRA);
;  340          0467  2     SCA_FC = TRUE;                          !Next character is first character in word
;  341          0468  2     SCA_FC_CASE = TRUE;                     !Use first character case rules on next word.
;  342          0469  1     END;                                    !End of ENDWRD


                                                        .TITLE  ENDWRD Perform end-word processing
                                                        .IDENT  \V04-000\

                                                        .EXTRN  RINTES, GCA, IRAC
                                                        .EXTRN  MRA, SCA, TSF, ENDCHR
                                                        .EXTRN  OUTLIN, OUTXPH

                                                        .PSECT  $CODE$,NOWRT,2

                                00FC 00000              .ENTRY  ENDWRD, Save R2,R3,R4,R5,R6,R7          ; 0223
                57 00000000G  EF   9E 00002             MOVAB   TSF, R7                                 ;
                56       00G  8F   9A 00009             MOVZBL  #RINTES, R6                             ;
                55 00000000G  EF   9E 0000D             MOVAB   MRA, R5                                 ;
                54 00000000G  EF   9E 00014             MOVAB   SCA+332, R4                             ;
                07     FF50   C4   E9 0001B             BLBC    SCA+156, 1$                             ; 0257
00000000G       EF            00   FB 00020             CALLS   #0, OUTXPH                              ; 0259
                56       CC   A4   D1 00027 1$:         CMPL    SCA+280, R6                             ; 0261
                              14   12 0002B             BNEQ    2$                                      ;
                         DC   A4   D5 0002D             TSTL    SCA+296                                 ; 0262
                              0F   12 00030             BNEQ    2$                                      ;
                         D8   A4   D5 00032             TSTL    SCA+292                                 ; 0263
                              0A   12 00035             BNEQ    2$                                      ;
                05     FF1C   D4   E8 00037             BLBS    @SCA+104, 2$                            ; 0265
                              64   D5 0003C             TSTL    SCA+332                                 ;
                              01   12 0003E             BNEQ    2$                                      ;
                                   04 00040             RET                                             ;
                7E            56   9A 00041 2$:         MOVZBL  R6, -(SP)                               ; 0272
00000000G       EF            01   FB 00044             CALLS   #1, ENDCHR                              ;
                     FF54     C4   D4 0004B             CLRL    SCA+160                                 ; 0277
                05       F4   A4   E9 0004F             BLBC    SCA+320, 3$                             ; 0284
                51            01   CE 00053             MNEGL   #1, R1                                  ; 0286
                              02   11 00056             BRB     4$                                      ;
                              51   D4 00058 3$:         CLRL    R1                                      ; 0284
                05       0C   AC   E9 0005A 4$:         BLBC    HYPHENATE, 5$                           ; 0289
                50            01   D0 0005E             MOVL    #1, R0                                  ;
                              02   11 00061             BRB     6$                                      ;
```

```
                                    50  D4 00063 5$:     CLRL    R0                                  ;
                        51          50  C0 00065 6$:     ADDL2   R0, ADJUST                          ;
                        52          67  D0 00068         MOVL    TSF, R2                             ; 0299
                        53          64  D0 0006B         MOVL    SCA+332, R3                         ;
          50            53     04   A2  C1 0006E         ADDL3   4(R2), R3, R0                       ;
                        50     B4   A4  C0 00073         ADDL2   SCA+256, R0                         ;
                        50          51  C0 00077         ADDL2   ADJUST, R0                          ;
                FF2C    D4          50  D1 0007A         CMPL    R0, @SCA+120                        ;
                                    0E  14 0007F         BGTR    7$                                  ;
                        50          65  D0 00081         MOVL    MRA, R0                             ; 0300
          51       0C   A0          0A  C1 00084         ADDL3   #10, 12(R0), R1                     ;
                        51     08   A0  D1 00089         CMPL    8(R0), R1                           ;
                                    0D  18 0008D         BGEQ    8$                                  ;
                            FF18    D4  DD 0008F 7$:     PUSHL   @SCA+100                            ; 0303
     00000000G          EF          01  FB 00093         CALLS   #1, OUTLIN                          ;
                                    2A  11 0009A         BRB     10$                                 ; 0299
                        08     F4   A4  E9 0009C 8$:     BLBC    SCA+320, 9$                         ; 0313
                               04   A2  D7 000A0         DECL    4(R2)                               ; 0316
                F8      B4     4E   8F  90 000A3         MOVB    #78, @SCA+324                       ; 0317
          50            62          53  C1 000A8 9$:     ADDL3   R3, (R2), R0                        ; 0320
          51       08   A4          03  C5 000AC         MULL3   #3, SCA+340, R1                     ; 0321
                        51          50  C0 000B1         ADDL2   R0, R1                              ;
          50       04   A4          03  C5 000B4         MULL3   #3, SCA+336, R0                     ; 0322
          62            51          50  C1 000B9         ADDL3   R0, R1, (R2)                        ;
                   04   A2          53  C0 000BD         ADDL2   R3, 4(R2)                           ; 0323
                   20   A2     04   A4  C0 000C1         ADDL2   SCA+336, 32(R2)                     ; 0324
                        50          67  D0 000C6 10$:     MOVL    TSF, R0                             ; 0325
                   08   A0     FF74 C4  C8 000C9         BISL2   SCA+192, 8(R0)                      ; 0341
                                    51  D4 000CF         CLRL    R1                                  ; 0342
                               08   A4  D5 000D1         TSTL    SCA+340                             ;
                                    02  15 000D4         BLEQ    11$                                 ;
                                    51  D6 000D6         INCL    R1                                  ;
     52       08   A0          01  01  EF 000D8 11$:     EXTZV   #1, #1, 8(R0), R2                   ;
                        52          51  88 000DE         BISB2   R1, R2                              ;
08   A0            01  01          52  F0 000E1         INSV    R2, #1, #1, 8(R0)                   ;
     51      0080  C0  01          00  EF 000E7         EXTZV   #0, #1, 128(R0), R1                 ; 0343
     52        C4  A4  01          00  EF 000EE         EXTZV   #0, #1, SCA+272, R2                 ;
                        51          52  88 000F4         BISB2   R2, R1                              ;
0080 C0            01  00          51  F0 000F7         INSV    R1, #0, #1, 128(R0)                 ;
                   1C  A0     C8   A4  D0 000FE         MOVL    SCA+276, 28(R0)                     ; 0344
                   10  A0     D0   A4  D0 00103         MOVL    SCA+284, 16(R0)                     ; 0345
44   A0 00000000G  EF  01          02  EF 00108         EXTZV   #2, #1, GCA+76, 68(R0)              ; 0346
                   48  A0     B8   A4  D0 00112         MOVL    SCA+260, 72(R0)                     ; 0347
                   4C  A0     D4   A4  D0 00117         MOVL    SCA+288, 76(R0)                     ; 0348
                   0C  A0     D8   A4  C0 0011C         ADDL2   SCA+292, 12(R0)                     ; 0352
                               D8  A4  D4 00121         CLRL    SCA+292                             ; 0353
                        06     0C   AC  E9 00124         BLBC    HYPHENATE, 12$                      ; 0356
                        60          03  C0 00128         ADDL2   #3, (R0)                            ; 0359
                               04   A0  D6 0012B         INCL    4(R0)                               ; 0360
                   30  A0     C0   A4  D0 0012E 12$:     MOVL    SCA+268, 48(R0)                     ; 0363
                        60     B0   A4  C0 00133         ADDL2   SCA+252, (R0)                       ; 0364
                   04  A0     B4   A4  C0 00137         ADDL2   SCA+256, 4(R0)                      ; 0365
                               38  A0  D5 0013C         TSTL    56(R0)                              ; 0367
                                    05  12 0013F         BNEQ    13$                                 ;
                   38  A0     DC   A4  D0 00141         MOVL    SCA+296, 56(R0)                     ; 0369
                        51     E0   A4  D0 00146 13$:     MOVL    SCA+300, R1                         ; 0371
                                    04  13 0014A         BEQL    14$                                 ;
```

```
                3C  A0          51  D0 0014C          MOVL    R1, 60(R0)                  ; 0373
                                64  D4 00150 14$:    CLRL    SCA+332                     ; 0377
                            04  A4  7C 00152         CLRQ    SCA+336                     ; 0378
                            F4  A4  7C 00155         CLRQ    SCA+320                     ; 0379
                            B0  A4  D5 00158         TSTL    SCA+252                     ; 0384
                                03  12 0015B         BNEQ    15$                         ;
                              00C0  31 0015D         BRW     21$                         ;
            0A     FF7C     C4  01  E0 00160 15$:    BBS     #1, SCA+200, 16$            ; 0388
                            42  04  AC E9 00166      BLBC    SPACE, 18$                  ;
            2C     FF7C     C4  01  E1 0016A         BBC     #1, SCA+200, 17$            ; 0392
                     08     A4  01  D0 00170 16$:    MOVL    #1, SCA+340                 ; 0395
                            50  65  D0 00174         MOVL    MRA, R0                     ; 0396
                            51  04  A0 9E 00177      MOVAB   4(R0), R1                   ;
                     00     B1  56  90 0017B         MOVB    R6, @0(R1)                  ;
                                61  D6 0017F         INCL    (R1)                        ;
                            0C  A0  D6 00181         INCL    12(R0)                      ;
                     00     B1  55  8F 90 00184      MOVB    #85, @0(R1)                 ; 0397
                                61  D6 00189         INCL    (R1)                        ;
                            0C  A0  D6 0018B         INCL    12(R0)                      ;
                     00     B1  20  90 0018E         MOVB    #32, @0(R1)                 ; 0398
                                61  D6 00192         INCL    (R1)                        ;
                            0C  A0  D6 00194         INCL    12(R0)                      ;
                   FF7C     C4  02  8A 00197         BICB2   #2, SCA+200                 ; 0399
                            50  65  D0 0019C 17$:    MOVL    MRA, R0                     ; 0402
                     04     B0  20  90 0019F         MOVB    #32, @4(R0)                 ;
                            04  A0  D6 001A3         INCL    4(R0)                       ;
                            0C  A0  D6 001A6         INCL    12(R0)                      ;
                            64  01  D0 001A9         MOVL    #1, SCA+332                 ; 0403
                            16  FC  A4 E9 001AC 18$: BLBC    SCA+328, 19$                ; 0406
                            12  04  AC E9 001B0      BLBC    SPACE, 19$                  ; 0407
                            50  65  D0 001B4         MOVL    MRA, R0                     ; 0410
                     04     B0  20  90 001B7         MOVB    #32, @4(R0)                 ;
                            04  A0  D6 001BB         INCL    4(R0)                       ;
                            0C  A0  D6 001BE         INCL    12(R0)                      ;
                            64  D6 001C1             INCL    SCA+332                     ; 0411
                            FC  A4  D4 001C3         CLRL    SCA+328                     ; 0412
                            27  08  AC E9 001C6 19$: BLBC    JUSTIFY, 20$                ; 0416
                     04     A4  01  D0 001CA         MOVL    #1, SCA+336                 ; 0419
                            50  65  D0 001CE         MOVL    MRA, R0                     ; 0420
                            51  04  A0 9E 001D1      MOVAB   4(R0), R1                   ;
                     00     B1  56  90 001D5         MOVB    R6, @0(R1)                  ;
                                61  D6 001D9         INCL    (R1)                        ;
                            0C  A0  D6 001DB         INCL    12(R0)                      ;
                     00     B1  4A  8F 90 001DE      MOVB    #74, @0(R1)                 ; 0421
                                61  D6 001E3         INCL    (R1)                        ;
                            0C  A0  D6 001E5         INCL    12(R0)                      ;
                     00     B1  20  90 001E8         MOVB    #32, @0(R1)                 ; 0422
                                61  D6 001EC         INCL    (R1)                        ;
                            0C  A0  D6 001EE         INCL    12(R0)                      ;
                            2B  0C  AC E9 001F1 20$: BLBC    HYPHENATE, 21$              ; 0426
                     F4     A4  01  D0 001F5         MOVL    #1, SCA+320                 ; 0429
                            50  65  D0 001F9         MOVL    MRA, R0                     ; 0430
                            51  04  A0 9E 001FC      MOVAB   4(R0), R1                   ;
                     00     B1  56  90 00200         MOVB    R6, @0(R1)                  ;
                                61  D6 00204         INCL    (R1)                        ;
                            0C  A0  D6 00206         INCL    12(R0)                      ;
                     F8     A4  61  D0 00209         MOVL    (R1), SCA+324               ; 0431
```

```
                    00   B1     49   8F  90 0020D          MOVB    #73, @0(R1)                  ; 0432
                                      61  D6 00212          INCL    (R1)                         ;
                                 0C   A0  D6 00214          INCL    12(R0)                       ;
                    00   B1           2D  90 00217          MOVB    #45, @0(R1)                  ; 0433
                                      61  D6 0021B          INCL    (R1)                         ;
                                 0C   A0  D6 0021D          INCL    12(R0)                       ;
                               FF74   C4  D4 00220 21$:     CLRL    SCA+192                      ; 0439
                                 B0   A4  7C 00224          CLRQ    SCA+252                      ; 0440
                                 DC   A4  7C 00227          CLRQ    SCA+296                      ; 0442
                    C4   A4           01  8A 0022A          BICB2   #1, SCA+272                  ; 0444
                               FF14   C4  D4 0022E          CLRL    SCA+96                       ; 0447
                  FEB4   C4    FED4   C4  7D 00232          MOVQ    SCA+32, SCA                  ; 0448
                  FEBC   C4    FEDC   C4  7D 00239          MOVQ    SCA+40, SCA+8                ; 0450
                  FEC4   C4    FED4   C4  D0 00240          MOVL    SCA+32, SCA+16               ; 0452
                  FECC   C4    FED8   C4  D0 00247          MOVL    SCA+36, SCA+24               ; 0453
                  FEC8   C4    FEDC   C4  D0 0024E          MOVL    SCA+40, SCA+20               ; 0454
                  FED0   C4    FEE0   C4  D0 00255          MOVL    SCA+44, SCA+28               ; 0455
                  FEE4   C4    FF04   C4  7D 0025C          MOVQ    SCA+80, SCA+48               ; 0456
                  FEEC   C4    FF0C   C4  7D 00263          MOVQ    SCA+88, SCA+56               ; 0458
                  FEF4   C4    FF04   C4  D0 0026A          MOVL    SCA+80, SCA+64               ; 0460
                  FEFC   C4    FF08   C4  D0 00271          MOVL    SCA+84, SCA+72               ; 0461
                  FEF8   C4    FF0C   C4  D0 00278          MOVL    SCA+88, SCA+68               ; 0462
                  FF00   C4    FF10   C4  D0 0027F          MOVL    SCA+92, SCA+76               ; 0463
                     50                65  D0 00286          MOVL    MRA, R0                      ; 0466
                     AC   A4       04  A0  D0 00289          MOVL    4(R0), SCA+248               ;
                  FF48   C4           01  D0 0028E          MOVL    #1, SCA+148                  ; 0467
                     84   A4          01  D0 00293          MOVL    #1, SCA+208                  ; 0468
                                          04 00297          RET                                  ; 0469

; Routine Size:  664 bytes,    Routine Base:  $CODE$ + 0000

;  343          0470  1
;  344          0471  1 END                                          !End of module
;  345          0472  0 ELUDOM


;                                  PSECT SUMMARY
;
;      Name                      Bytes                          Attributes
;
;  $CODE$                          664  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        00:00.2
;  _$255$DUA28:[RUNOFF.SRC]DSRLIB.L32;1   1248        92         7         86        00:00.3
```

```
;                                       COMMAND QUALIFIERS

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

; Size:          664 code + 0 data bytes
; Run Time:         00:17.5
; Elapsed Time:     00:50.5
; Lines/CPU Min:     1615
; Lexemes/CPU-Min: 23462
; Memory Used:  205 pages
; Compilation Complete
```

FIND
LIS

ENDWRD
LIS

ERROR
LIS

FIGURE
LIS

FLGSEM
LIS

FOOFIL
LIS

GCODE
LIS

FCIMRA
LIS

FNONLY
LIS

FJNFNJ
LIS

FOOBOT
LIS

GBLDCL
LIS

FNDFLG
LIS

FOOOUT
LIS

FORMAT
LIS