```
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
```

```
PPPPPPPP    UU      UU    SSSSSSSS
PPPPPPPP    UU      UU    SSSSSSSS
PP      PP  UU      UU  SS
PP      PP  UU      UU  SS
PP      PP  UU      UU  SS
PP      PP  UU      UU  SS
PPPPPPPP    UU      UU    SSSSSS
PPPPPPPP    UU      UU    SSSSSS
PP          UU      UU          SS
PP          UU      UU          SS
PP          UU      UU          SS      ....
PP          UU      UU          SS      ....
PP          UUUUUUUUUU  SSSSSSSS        ....
PP          UUUUUUUUUU  SSSSSSSS        ....


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 'PUS -- Process Uneaten String'
;  2          0002  0 MODULE pus (    IDENT = 'V04-000'
;  3        P 0003  0                 %BLISS32 [, ADDRESSING_MODE (EXTERNAL    = LONG_RELATIVE,
;  4          0004  0                                              NONEXTERNAL = LONG_RELATIVE)]
;  5          0005  0                 ) =
;  6          0006  1 BEGIN
;  7          0007  1 !
;  8          0008  1 !*****************************************************************************
;  9          0009  1 !*                                                                           *
; 10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
; 11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
; 12          0012  1 !*  ALL RIGHTS RESERVED.                                                     *
; 13          0013  1 !*                                                                           *
; 14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
; 18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
; 19          0019  1 !*  TRANSFERRED.                                                             *
; 20          0020  1 !*                                                                           *
; 21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
; 22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
; 23          0023  1 !*  CORPORATION.                                                             *
; 24          0024  1 !*                                                                           *
; 25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
; 26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
; 27          0027  1 !*                                                                           *
; 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 !       Decide if record is a command, or contains text,
; 38          0038  1 !       and dispatch accordingly.
; 39          0039  1 !
; 40          0040  1 ! ENVIRONMENT:  Transportable
; 41          0041  1 !
; 42          0042  1 ! AUTHOR:       R.W.Friday       CREATION DATE: April, 1978
; 43          0043  1 !
; 44          0044  1
```

```
;    46          0045  1 %SBTTL 'Revision History'
;    47          0046  1 !
;    48          0047  1 ! MODIFIED BY:
;    49          0048  1 !
;    50          0049  1 !       008     KFA00008        Ken Alden       19-Sep-1983
;    51          0050  1 !               Added support for end-of-line escape sequences.
;    52          0051  1 !
;    53          0052  1 !       007     KFA00007        Ken Alden       10-March-1983
;    54          0053  1 !               Fixed bug with sca_ker.  Conditionals needed cleaning up.
;    55          0054  1 !
;    56          0055  1 !       006     RER00006        Ron Randall     07-March-1983
;    57          0056  1 !               Global edit of all modules. Updated module names, idents,
;    58          0057  1 !               copyright dates. Changed require files to BLISS library.
;    59          0058  1 !               Changed handler_code to kwitem_handler_code.
;    60          0059  1 !--
;    61          0060  1
```

```
;    63          0061  1 %SBTTL 'Module Level Declarations'
;    64          0062  1
;    65          0063  1 !
;    66          0064  1 ! TABLE OF CONTENTS:
;    67          0065  1 !
;    68          0066  1 ! INCLUDE FILES:
;    69          0067  1 !
;    70          0068  1 LIBRARY 'NXPORT:XPORT';                 ! XPORT Library
;    71          0069  1 REQUIRE 'REQ:RNODEF';                   ! RUNOFF variant definitions
;    72          0200  1
;    73        U 0201  1 %IF DSRPLUS %THEN
;    74        U 0202  1 LIBRARY 'REQ:DPLLIB';                 ! DSRPLUS BLISS Library
;    75          0203  1 %ELSE
;    76          0204  1 LIBRARY 'REQ:DSRLIB';                 ! DSR BLISS Library
;    77          0205  1 %FI
;    78          0206  1
;    79          0207  1 !
;    80          0208  1 ! EQUATED SYMBOLS:
;    81          0209  1 !
;    82          0210  1 LITERAL
;    83          0211  1     FINISH_COMMAND = ARECCC_COUNT + 1,
;    84          0212  1     PROCESS_TEXT   = ARECCC_COUNT + 2;
;    85          0213  1
;    86          0214  1 !
;    87          0215  1 ! EXTERNAL REFERFNCES:
;    88          0216  1 !
;    89          0217  1 EXTERNAL LITERAL
;    90          0218  1     RINTES : UNSIGNED (8);
;    91          0219  1
;    92          0220  1 EXTERNAL
;    93          0221  1     GCA : GCA_DEFINITION,
;    94          0222  1     IRA : FIXED_STRING,
;    95          0223  1     KHAR,
;    96          0224  1     SCA : SCA_DEFINITION,
;    97          0225  1     TSF : TSF_DEFINITION;
;    98          0226  1
;    99          0227  1 EXTERNAL LITERAL
;   100          0228  1     rnfdns,     ! W - .NO SPACE illegal in this context
;   101          0229  1     rnfilc;     ! W - Illegal command: "%S"
;   102          0230  1
;   103          0231  1 EXTERNAL ROUTINE
;   104          0232  1     AREC,       DOCM,   ENDCMT, ENDWRD, ERMA,
;   105          0233  1     ERML,       GCSKIP, GUSKIP, LIT,    OUTCRG,
;   106          0234  1     OUTJ,       OUTNJ,  PARAG,  RCS,    SCANT;
;   107          0235  1
```

```
;  109     0236  1 %sbttl 'PUS -- process uneaten string'
;  110     0237  1 GLOBAL ROUTINE PUS (end_word) : NOVALUE =
;  111     0238  1
;  112     0239  1 !++
;  113     0240  1 ! FUNCTIONAL DESCRIPTION:
;  114     0241  1 !
;  115     0242  1 !   Decide if record is a command, or contains text, and dispatch accordingly.
;  116     0243  1 !
;  117     0244  1 ! FORMAL PARAMETERS:
;  118     0245  1 !
;  119     0246  1 !       END_WORD    TRUE if PUS is to terminate words when found.  i.e. if this
;  120     0247  1 !                   parameter is true, ENDWRD will be called.
;  121     0248  1 !                   FALSE to prevent PUS from calling ENDWRD where it normally
;  122     0249  1 !                   would have.
;  123     0250  1 !
;  124     0251  1 ! IMPLICIT INPUTS:     None
;  125     0252  1 !
;  126     0253  1 ! IMPLICIT OUTPUTS:    None
;  127     0254  1 !
;  128     0255  1 ! ROUTINE VALUE:
;  129     0256  1 ! COMPLETION CODES:    None
;  130     0257  1 !
;  131     0258  1 ! SIDE EFFECTS:        Many and varied! Be sure you understand the
;  132     0259  1 !                      implications before making any changes to this routine.
;  133     0260  1 !
;  134     0261  1 !--
;  135     0262  1
;  136     0263  2     BEGIN
;  137     0264  2
;  138     0265  2     LOCAL
;  139     0266  2         action,
;  140     0267  2         new_record,                         !PUS needs to know if new record or continuing on.
;  141     0268  2         save_count,
;  142     0269  2         save_khar,
;  143     0270  2         save_next;
;  144     0271  2
;  145     0272  2     save_count = .fs_length (ira);      ! Needed if this turns out to be a
;  146     0273  2     save_khar = .khar;                  !  literal line.
;  147     0274  2     save_next = .fs_next (ira);
;  148     0275  2
;  149     0276  2     action = arec ();           ! Find out what to do with this uneaten string.
;  150     0277  2
;  151     0278  2     new_record = true;          ! This is a brand new record.
;  152     0279  2
;  153     0280  2     WHILE 1 DO                  ! This loop allows the analysis results
;  154     0281  3         BEGIN                   !     to be altered.
;  155     0282  3
;  156     0283  3         IF  .action NEQ finish_command
;  157     0284  3         THEN
;  158     0285  4             BEGIN
;  159     0286  4             !If in the FALSE branch of an .IF command, ignore everything
;  160     0287  4             !that cannot possibly be a "normal" RUNOFF command.
;  161     0288  4
;  162     0289  4             IF  .gca_skipping
;  163     0290  4             THEN
;  164     0291  4
;  165     0292  4                 IF  .action NEQ cf_letter
```

```
;  166     0293  4                    THEN
;  167     0294  4                        RETURN;
;  168     0295  4
;  169     0296  4                IF  .gca_literal
;  170     0297  6                    AND NOT ((.action EQL end_of_string)
;  171     0298  5                    AND NOT .new_record )
;  172     0299  4                THEN
;  173     0300  5                    BEGIN                       ! Special processing for literals
;  174     0301  5                    LOCAL
;  175     0302  5                        rcs_result;
;  176     0303  5
;  177     0304  5                    LABEL
;  178     0305  5                        literal_block;
;  179     0306  5
;  180     0307  5 literal_block :
;  181     0308  6                    BEGIN
;  182     0309  6                    !The following test is satisfied when a counted literal has run
;  183     0310  6                    ! run out, or it's not a counted literal.  In the case where a
;  184     0311  6                    ! counted literal is supposed to run out, but the line does not
;  185     0312  6                    ! contain a .END LITERAL command, the appropriate error action
;  186     0313  6                    ! is taken by LIT.
;  187     0314  6
;  188     0315  6                    IF  .gca_lit_count EQL 0
;  189     0316  6                    THEN
;  190     0317  7                        BEGIN
;  191     0318  7
;  192     0319  7                        IF  .action EQL cf_letter
;  193     0320  7                        THEN
;  194     0321  8                            BEGIN
;  195     0322  8                            rcs_result = rcs ();
;  196     0323  8
;  197     0324  8                            IF  .rcs_result NEQ 0
;  198     0325  8                            THEN
;  199     0326  9                                BEGIN
;  200     0327  9                                MAP
;  201     0328  9                                    rcs_result : REF VECTOR;
;  202     0329  9
;  203     0330  9                                IF  .kwitem_handler_code (rcs_result) EQL h_end_literal
;  204     0331  9                                THEN
;  205     0332 10                                    BEGIN           !Terminate the literal.
;  206     0333 10                                    docm (h_end_literal, 0);
;  207     0334 10                                    action = finish_command;
;  208     0335 10                                    LEAVE literal_block;
;  209     0336  9                                    END;
;  210     0337  9
;  211     0338  8                                END;
;  212     0339  8
;  213     0340  7                            END;
;  214     0341  7
;  215     0342  6                    END;
;  216     0343  6
;  217     0344  6                !Just another literal line
;  218     0345  6                fs_length (ira) = .save_count;  ! Restore possibly modified
;  219     0346  6                fs_next (ira) = .save_next;     !  count, pointer,
;  220     0347  6                khar = .save_khar;              !  and stacked character.
;  221     0348  6
;  222     0349  6                lit (0);                        ! Let LIT count this line.
```

```
;  223     0350  6
;  224     0351  6                 IF  .action NEQ end_of_string
;  225     0352  6                 THEN
;  226     0353  6                     action = process_text;
;  227     0354  6
;  228     0355  5                 END;            ! ***** end of LITERAL_BLOCK *****
;  229     0356  5
;  230     0357  4                 END;
;  231     0358  3             END;
;  232     0359  3
;  233     0360  3         CASE .action FROM 1 TO process_text OF
;  234     0361  3             SET
;  235     0362  3
;  236     0363  3             [end_of_string] :
;  237     0364  4                 BEGIN
;  238     0365  4
;  239     0366  4                 IF  .sca_fill AND .gca_autopara AND .new_record
;  240     0367  4                 THEN
;  241     0368  4                     parag (-1)
;  242     0369  4                 ELSE
;  243     0370  5                     BEGIN
;  244     0371  5                     IF  .sca_fill
;  245     0372  5                     THEN                        ! Fill mode
;  246     0373  6                         BEGIN
;  247     0374  7                         IF  ((NOT .sca_fc) AND .end_word)
;  248     0375  6                         THEN
;  249     0376  6                             endwrd (.sca_fill, .sca_justify, false);
;  250     0377  6                         END
;  251     0378  6
;  252     0379  5                     ELSE                        ! Nofill mode
;  253     0380  5                         IF  .sca_ker            ! If "Keeping Empty Records",
;  254     0381  5                         THEN                    !  force a blank line
;  255     0382  6                             BEGIN
;  256     0383  6
;  257     0384  6                             IF  .tsf_int_hl GTR 0 !Force out the current line
;  258     0385  6                             THEN
;  259     0386  6                                 outnj ();       !   first if it is not empty.
;  260     0387  6
;  261     0388  6                             guskip (1);         ! Generate 'blank-line' code.
;  262     0389  6                             outcrg ();          ! Force out blank line NOW.
;  263     0390  5                             END;
;  264     0391  4                         END;
;  265     0392  4                 RETURN;
;  266     0393  3                 END;
;  267     0394  3
;  268     0395  3             [CF_CF] :                       ! Two <CONTROL flag>s
;  269     0396  3                 action = not_space;
;  270     0397  3
;  271     0398  3             [CF_COMMENT] :
;  272     0399  4                 BEGIN                       ! A comment starting '.!' or '.;'.
;  273     0400  4                 kcns ();                    ! Position into comment
;  274     0401  4                 endcmt ();
;  275     0402  4                 action = finish_command;
;  276     0403  3                 END;
;  277     0404  3
;  278     0405  3             [CF_BREAK, CF_LETTER] :
;  279     0406  4                 BEGIN                       ! Command recognition and processing
```

```
;  280          0407  4                  LOCAL
;  281          0408  4                      rcs_result;
;  282          0409  4
;  283          0410  4                  rcs_result = rcs ();    ! Try to recognize command.
;  284          0411  4
;  285          0412  4                  IF  .rcs_result EQL 0
;  286          0413  4                  THEN                    ! If the command was not recognized,
;  287          0414  4                      action = cf_other   !  change the action indicator
;  288          0415  4                  ELSE
;  289          0416  5                      BEGIN
;  290          0417  5                      MAP
;  291          0418  5                          rcs_result : REF VECTOR;
;  292          0419  5
;  293          0420  5                      ! Process command
;  294          0421  5                      docm (.kwitem_handler_code (rcs_result), .kwitem_actions (rcs_result));
;  295          0422  5
;  296          0423  5                      action = finish_command;    ! Finish processing command.
;  297          0424  4                      END;
;  298          0425  4
;  299          0426  3                  END;
;  300          0427  3
;  301          0428  3              [CF_OTHER] :
;  302          0429  4                  BEGIN
;  303          0430  4                  !Something that can't be recognized.
;  304          0431  4                  erma (rnfilc, true);        ! Skip to end of "command" and issue an error message.
;  305          0432  4                  action = finish_command;  ! Standard end-of-command processing.
;  306          0433  3                  END;
;  307          0434  3
;  308          0435  3              [END_FOOTNOTE] :
;  309          0436  4                  BEGIN
;  310          0437  4                  !<ENDFOOTNOTE flag>
;  311          0438  4                  kcns ();                    !Skip the <ENDFOOTNOTE flag>
;  312          0439  4                  docm (h_end_footnote, 0);   !Terminate a footnote.
;  313          0440  4                  action = finish_command;    !Finish processing coommand.
;  314          0441  3                  END;
;  315          0442  3
;  316          0443  3              [FIRST_SPACE] :
;  317          0444  4                  BEGIN
;  318          0445  4
;  319          0446  4                  IF  .sca_fill AND .gca_autopara
;  320          0447  4                  THEN
;  321          0448  4                      parag (-1);             !Found an auto'd paragraph.
;  322          0449  4
;  323          0450  4                  action = process_text;      !Do normal text processing.
;  324          0451  3                  END;
;  325          0452  3
;  326          0453  3              [FIRST_SEMI] :                  !A ';' following a command.
;  327          0454  3                  action = process_text;      !Do normal text processing.
;  328          0455  3
;  329          0456  3              [NOT_SPACE] :
;  330          0457  4                  BEGIN
;  331          0458  4
;  332          0459  4                  IF  .sca_fill AND .gca_autotabl
;  333          0460  4                  THEN
;  334          0461  4                      parag (-1);             !Found an auto'd paragraph.
;  335          0462  4
;  336          0463  4                  action = process_text;      !Do normal text processing.
```

```
;  337     0464  3                    END;
;  338     0465  3
;  339     0466  3                [PROCESS_TEXT] :
;  340     0467  4                    BEGIN
;  341     0468  4
;  342     0469  5                    IF  NOT (.gca_concat AND .sca_cont)
;  343     0470  4                    THEN
;  344     0471  4                    !Normal case, when end of line means end
;  345     0472  4                    !of word.  The check on SCA_WRD_CPEND avoids
;  346     0473  4                    !extra spaces if the input line ended with
;  347     0474  4                    !a space or tab.
;  348     0475  4
;  349     0476  4                        ! If there was a word pending, end it.
;  350     0477  4                        ! In DSRPLUS the pass-through facility will
;  351     0478  4                        ! push through a sequence and not count it against
;  352     0479  4                        ! the margins.  However when this type of sequence is
;  353     0480  4                        ! dropped at the end of a line, then sca_wrd_cpend will
;  354     0481  4                        ! equal rintes.  ENDWRD still needs to be called however.
;  355     0482  4                        ! For this reason, if a pass-through was part of the last
;  356     0483  4                        ! word, then we'll call endwrd if sca_wrd_cpend eql rintes.
;  357     0484  4
;  358     0485  4                        IF  .end_word AND
;  359     0486  6                            ((.sca_wrd_cpend NEQ rintes)
;  360   U 0487  6 %IF DSRPLUS %THEN
;  361   U 0488  6                                            OR
;  362   U 0489  6                            ((.sca_wrd_cpend EQL rintes) AND (.sca_wrd_pass)))
;  363     0490  5 %ELSE)
;  364     0491  5 %FI
;  365     0492  4                        THEN
;  366     0493  4                            endwrd (.sca_fill, .sca_justify, false);
;  367     0494  4
;  368     0495  5                    IF  .gca_concat AND (NOT .sca_cont)
;  369     0496  4                    THEN                    !NO SPACE canceled by some other command
;  370     0497  4                        erml (RNFDNS);      !Issue error message.
;  371     0498  4
;  372     0499  4                    gca_concat = false; !Concatination no longer pending.
;  373     0500  4                    scant ();           !Go scan text.
;  374     0501  4
;  375     0502  4                    !In .NO FILL mode, the end of an input line
;  376     0503  4                    !corresponds to the end of an output line.
;  377     0504  4
;  378     0505  4                    IF  NOT .sca_fill
;  379     0506  4                    THEN
;  380     0507  4
;  381     0508  4                        IF  .sca_justify
;  382     0509  4                        THEN                ! No fill,justify
;  383     0510  4                            outj ()
;  384     0511  4                        ELSE                ! No fill, no justify
;  385     0512  4                            outnj ();
;  386     0513  4
;  387     0514  4                    RETURN;
;  388     0515  3                    END;
;  389     0516  3
;  390     0517  3                [FINISH_COMMAND] :
;  391     0518  4                    BEGIN
;  392     0519  4                    new_record = false;     ! Continue same record.
;  393     0520  4                    action = arec ();       ! Find out what to do.
```

```
;  394          0521  4
;  395          0522  4                    IF  .action EQL end_of_string
;  396          0523  4                    THEN
;  397          0524  4                        RETURN;
;  398          0525  4
;  399          0526  4                    IF  .action EQL first_semi
;  400          0527  4                    THEN
;  401          0528  5                        BEGIN
;  402          0529  5                        kcns ();            ! Skip the ';',
;  403          0530  5                        action = arec ();   ! find out what to do with remainder,
;  404          0531  5                        new_record = true;  ! & handle it like a brand new record.
;  405          0532  4                        END;
;  406          0533  4
;  407          0534  4            ! If the command just processed was a .LITERAL command, don't let the
;  408          0535  4            ! old pointers, etc. get set back at the top of the loop.
;  409          0536  4
;  410          0537  4                    IF  .gca_literal
;  411          0538  4                    THEN
;  412          0539  5                        BEGIN
;  413          0540  5                        save_count = .fs_length (ira);
;  414          0541  5                        save_khar = .khar;
;  415          0542  5                        save_next = .fs_next (ira);
;  416          0543  4                        END;
;  417          0544  4
;  418          0545  3                    END;
;  419          0546  3
;  420          0547  3                TES;
;  421          0548  3
;  422          0549  2            END;
;  423          0550  2
;  424          0551  1        END;                                    !End of PUS


                                                .TITLE  PUS PUS -- Process Uneaten String
                                                .IDENT  \V04-000\

                                                .EXTRN  RINTES, GCA, IRA
                                                .EXTRN  KHAR, SCA, TSF, RNFDNS
                                                .EXTRN  RNFILC, AREC, DOCM
                                                .EXTRN  ENDCMT, ENDWRD, ERMA
                                                .EXTRN  ERML, GCSKIP, GUSKIP
                                                .EXTRN  LIT, OUTCRG, OUTJ
                                                .EXTRN  OUTNJ, PARAG, RCS
                                                .EXTRN  SCANT

                                                .PSECT  $CODE$,NOWRT,2

                                    0FFC 00000  .ENTRY  PUS, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11    ; 0237
                5B 00000000G   EF    9E 00002  MOVAB   KHAR, R11                                     ;
                5A 00000000G   EF    9E 00009  MOVAB   GCA+48, R10                                   ;
                59 00000000G   EF    9E 00010  MOVAB   SCA+104, R9                                   ;
                58 00000000G   EF    9E 00017  MOVAB   IRA+12, R8                                    ;
                57             68    D0 0001E  MOVL    IRA+12, SAVE_COUNT                            ; 0272
                56             6B    D0 00021  MOVL    KHAR, SAVE_KHAR                               ; 0273
                55       F8    A8    D0 00024  MOVL    IRA+4, SAVE_NEXT                              ; 0274
00000000G       EF             00    FB 00028  CALLS   #0, AREC                                      ; 0276
                52             50    D0 0002F  MOVL    R0, ACTION                                    ;
```

```
                        54           01  D0 00032          MOVL    #1, NEW_RECORD                          ; 0278
                        0B           52  D1 00035 1$:      CMPL    ACTION, #11                             ; 0283
                                     5B  13 00038          BEQL    5$
                        06      6C   AA  E9 0003A          BLBC    GCA+156, 2$                             ; 0289
                        05           52  D1 0003E          CMPL    ACTION, #5                              ; 0292
                                     01  13 00041          BEQL    2$
                                         04 00043          RET
                        4D      20   AA  E9 00044 2$:      BLBC    GCA+80, 5$                              ; 0296
                        01           52  D1 00048          CMPL    ACTION, #1                              ; 0297
                                     03  12 0004B          BNEQ    3$
                        45           54  E9 0004D          BLBC    NEW_RECORD, 5$                          ; 0298
                                 08  AA  D5 00050 3$:      TSTL    GCA+56                                  ; 0315
                                     25  12 00053          BNEQ    4$
                        05           52  D1 00055          CMPL    ACTION, #5                              ; 0319
                                     20  12 00058          BNEQ    4$
                00000000G    EF          00  FB 0005A          CALLS   #0, RCS                             ; 0322
                                         50  D5 00061          TSTL    RCS_RESULT                          ; 0324
                                         15  13 00063          BEQL    4$
                        3C       04  A0  D1 00065          CMPL    4(RCS_RESULT), #60                      ; 0330
                                     0F  12 00069          BNEQ    4$
                        7E           3C  7D 0006B          MOVQ    #60, -(SP)                              ; 0333
                00000000G    EF          02  FB 0006E          CALLS   #2, DOCM
                        52           0B  D0 00075          MOVL    #11, ACTION                             ; 0334
                                     1B  11 00078          BRB     5$                                      ; 0335
                        68           57  D0 0007A 4$:      MOVL    SAVE_COUNT, IRA+12                      ; 0345
                     F8 A8           55  D0 0007D          MOVL    SAVE_NEXT, IRA+4                        ; 0346
                        6B           56  D0 00081          MOVL    SAVE_KHAR, KHAR                         ; 0347
                                         7E  D4 00084          CLRL    -(SP)                               ; 0349
                00000000G    EF          01  FB 00086          CALLS   #1, LIT
                        01           52  D1 0008D          CMPL    ACTION, #1                              ; 0351
                                         03  13 00090          BEQL    5$
                        52           0C  D0 00092          MOVL    #12, ACTION                             ; 0353
                     0B              01  52  CF 00095 5$:      CASEL   ACTION, #1, #11                     ; 0360
0077                 0072            0096        0018          00099 6$:      .WORD   7$-6$,-
00FE                 00E2            00BD        0096          000A1                  18$-6$,-
0104                 015E            00EC        00AC          000A9                  14$-6$,-
                                                                                    15$-6$,-
                                                                                    18$-6$,-
                                                                                    21$-6$,-
                                                                                    27$-6$,-
                                                                                    30$-6$,-
                                                                                    20$-6$,-
                                                                                    28$-6$,-
                                                                                    37$-6$,-
                                                                                    32$-6$
                        53       00  B9  D0 000B1 7$:      MOVL    @SCA+104, R3                            ; 0366
                        2E           53  E9 000B5          BLBC    R3, 11$
                        0E       D4  BA  E9 000B8          BLBC    @GCA+4, 8$
                        0B           54  E9 000BC          BLBC    NEW_RECORD, 8$
                        7E           01  CE 000BF          MNEGL   #1, -(SP)                               ; 0368
                00000000G    EF          01  FB 000C2          CALLS   #1, PARAG
                                         04  000C9          RET
                        19           53  E9 000CA 8$:      BLBC    R3, 11$                                 ; 0371
                        01       2C  A9  E9 000CD          BLBC    SCA+148, 9$                             ; 0374
                                         04  000D1          RET
                        01       04  AC  E8 000D2 9$:      BLBS    END_WORD, 10$
                                         04  000D6          RET
```

```
                                                7E  D4 000D7 10$:   CLRL    -(SP)                                   ; 0376
                                             FC B9  DD 000D9      PUSHL   @SCA+100                                ;
                                                53  DD 000DC      PUSHL   R3                                      ;
                                00000000G    EF 03  FB 000DE      CALLS   #3, ENDWRD                              ;
                                                04  000E5      RET                                             ; 0371
                                          01    1C B9  E8 000E6 11$:   BLBS    @SCA+132, 12$                    ; 0380
                                                04  000EA      RET                                             ;
                                     00000000G  FF  D5 000EB 12$:   TSTL    @TSF                             ; 0384
                                                07  15 000F1      BLEQ    13$                                     ;
                                00000000G    EF 00  FB 000F3      CALLS   #0, OUTNJ                               ; 0386
                                                01  DD 000FA 13$:   PUSHL   #1                                      ; 0388
                                00000000G    EF 01  FB 000FC      CALLS   #1, GUSKIP                              ;
                                00000000G    EF 00  FB 00103      CALLS   #0, OUTCRG                              ; 0389
                                                04  0010A      RET                                             ; 0364
                                          52    0A  D0 0010B 14$:   MOVL    #10, ACTION                         ; 0396
                                                69  11 0010E      BRB     26$                                     ;
                                                68  D5 00110 15$:   TSTL    IRA+12                              ; 0400
                                                09  14 00112      BGTR    16$                                     ;
                                          6B    00G 8F  9A 00114      MOVZBL  #RINTES, KHAR                           ;
                                          68        01  CE 00118      MNEGL   #1, IRA+12                              ;
                                                09  11 0011B      BRB     17$                                     ;
                                          6B    F8  B8  9A 0011D 16$:   MOVZBL  @IRA+4, KHAR                    ;
                                                F8  A8  D6 00121      INCL    IRA+4                                   ;
                                                68  D7 00124      DECL    IRA+12                                  ;
                                00000000G    EF 00  FB 00126 17$:   CALLS   #0, ENDCMT                       ; 0401
                                                47  11 0012D      BRB     25$                                     ; 0402
                                00000000G    EF 00  FB 0012F 18$:   CALLS   #0, RCS                          ; 0410
                                                50  D5 00136      TSTL    RCS_RESULT                          ; 0412
                                                05  12 00138      BNEQ    19$                                     ;
                                          52    09  D0 0013A      MOVL    #9, ACTION                          ; 0414
                                                5B  11 0013D      BRB     31$                                     ;
                                          7E    04  A0  7D 0013F 19$:   MOVQ    4(RCS_RESULT), -(SP)            ; 0421
                                                2A  11 00143      BRB     24$                                     ;
                                                01  DD 00145 20$:   PUSHL   #1                                   ; 0431
                                     00000000G  8F  DD 00147      PUSHL   #RNFILC                                 ;
                                00000000G    EF 02  FB 0014D      CALLS   #2, ERMA                                ;
                                                20  11 00154      BRB     25$                                     ; 0432
                                                68  D5 00156 21$:   TSTL    IRA+12                              ; 0438
                                                09  14 00158      BGTR    22$                                     ;
                                          6B    00G 8F  9A 0015A      MOVZBL  #RINTES, KHAR                           ;
                                          68        01  CE 0015E      MNEGL   #1, IRA+12                              ;
                                                09  11 00161      BRB     23$                                     ;
                                          6B    F8  B8  9A 00163 22$:   MOVZBL  @IRA+4, KHAR                    ;
                                                F8  A8  D6 00167      INCL    IRA+4                                   ;
                                                68  D7 0016A      DECL    IRA+12                                  ;
                                          7E        39  7D 0016C 23$:   MOVQ    #57, -(SP)                      ; 0439
                                00000000G    EF 02  FB 0016F 24$:   CALLS   #2, DOCM                         ;
                                          52    0B  D0 00176 25$:   MOVL    #11, ACTION                         ; 0440
                                                1F  11 00179 26$:   BRB     31$                                     ; 0360
                                          18    00  B9  E9 0017B 27$:   BLBC    @SCA+104, 30$                   ; 0446
                                          14    D4  BA  E9 0017F      BLBC    @GCA+4, 30$                     ;
                                                08  11 00183      BRB     29$                                     ; 0448
                                          0E    00  B9  E9 00185 28$:   BLBC    @SCA+104, 30$                   ; 0459
                                          0A    DC  BA  E9 00189      BLBC    @GCA+12, 30$                    ;
                                          7E        01  CE 0018D 29$:   MNEGL   #1, -(SP)                       ; 0461
                                00000000G    EF 01  FB 00190      CALLS   #1, PARAG                               ;
                                          52    0C  D0 00197 30$:   MOVL    #12, ACTION                         ; 0463
```

```
                                        FE98  31 0019A 31$:    BRW     1$                                      ; 0360
                                  04      6A  E9 0019D 32$:    BLBC    GCA+48, 33$                             ; 0469
                                  1E      3C  A9  E8 001A0    BLBS    SCA+164, 34$                            ;
                                  1A      04  AC  E9 001A4 33$:    BLBC    END_WORD, 34$                       ; 0485
                00000000G         8F    00B0  C9  D1 001A8    CMPL    SCA+280, #RINTES                        ; 0486
                                              0F  13 001B1    BEQL    34$                                     ;
                                              7E  D4 001B3    CLRL    -(SP)                                   ; 0493
                                        FC    B9  DD 001B5    PUSHL   @SCA+100                                ;
                                        00    B9  DD 001B8    PUSHL   @SCA+104                                ;
                00000000G               EF        03  FB 001BB    CALLS   #3, ENDWRD                          ;
                                        11        6A  E9 001C2 34$:    BLBC    GCA+48, 35$                     ; 0495
                                        0D      3C  A9  E8 001C5    BLBS    SCA+164, 35$                        ;
                                           00000000G  8F  DD 001C9    PUSHL   #RNFDNS                         ; 0497
                00000000G               EF        01  FB 001CF    CALLS   #1, ERML                            ;
                                                  6A  D4 001D6 35$:    CLRL    GCA+48                          ; 0499
                00000000G               EF        00  FB 001D8    CALLS   #0, SCANT                           ; 0500
                                        5E      00    B9  E8 001DF    BLBS    @SCA+104, 42$                   ; 0505
                                        08      FC    B9  E9 001E3    BLBC    @SCA+100, 36$                   ; 0508
                00000000G               EF        00  FB 001E7    CALLS   #0, OUTJ                            ; 0510
                                                  04  00 001EE    RET                                         ;
                00000000G               EF        00  FB 001EF 36$:    CALLS   #0, OUTNJ                       ; 0512
                                                  04  00 001F6    RET                                         ; 0508
                                                  54  D4 001F7 37$:    CLRL    NEW_RECORD                      ; 0519
                00000000G               EF        00  FB 001F9    CALLS   #0, AREC                            ; 0520
                                        52        50  D0 00200    MOVL    R0, ACTION                          ;
                                        01        52  D1 00203    CMPL    ACTION, #1                          ; 0522
                                                  39  13 00206    BEQL    42$                                 ;
                                        08        52  D1 00208    CMPL    ACTION, #8                          ; 0526
                                                  23  12 0020B    BNEQ    40$                                 ;
                                                  68  D5 0020D    TSTL    IRA+12                              ; 0529
                                                  09  14 0020F    BGTR    38$                                 ;
                                     6B       00G   8F  9A 00211    MOVZBL  #RINTES, KHAR                       ;
                                     68             01  CE 00215    MNEGL   #1, IRA+12                          ;
                                                    09  11 00218    BRB     39$                                 ;
                                     6B       F8    B8  9A 0021A 38$:    MOVZBL  @IRA+4, KHAR                    ;
                                              F8    A8  D6 0021E    INCL    IRA+4                               ;
                                                    68  D7 00221    DECL    IRA+12                              ;
                00000000G                     EF        00  FB 00223 39$:    CALLS   #0, AREC                    ; 0530
                                              52        50  D0 0022A    MOVL    R0, ACTION                      ;
                                              54        01  D0 0022D    MOVL    #1, NEW_RECORD                  ; 0531
                                              0A      20    AA  E9 00230 40$:    BLBC    GCA+80, 41$             ; 0537
                                              57            68  D0 00234    MOVL    IRA+12, SAVE_COUNT          ; 0540
                                              56            6B  D0 00237    MOVL    KHAR, SAVE_KHAR             ; 0541
                                              55      F8    A8  D0 0023A    MOVL    IRA+4, SAVE_NEXT            ; 0542
                                                 FDF4       31  0023E 41$:    BRW     1$                      ; 0280
                                                            04  00241 42$:    RET                             ; 0551

; Routine Size:  578 bytes,    Routine Base:  $CODE$ + 0000

;  425          0552  1
;  426          0553  1 END                                                !End of module
;  427          0554  0 ELUDOM
```

```
;                                   PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  $CODE$                        578  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        43         3          86        00:00.2



;                                   COMMAND QUALIFIERS

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

; Size:         578 code + 0 data bytes
; Run Time:       00:12.0
; Elapsed Time:   00:25.3
; Lines/CPU Min:   2781
; Lexemes/CPU-Min: 14876
; Memory Used:  153 pages
; Compilation Complete
```

PAGE
LIS

PERMUTE
LIS

PAGCMB
LIS

PANPA
LIS

RCS
LIS

POOL
LIS

RINIT
LIS

PERIOD
LIS

PUS
LIS

PUTNDX
LIS

PARSEP
LIS

PUTTOC
LIS

REQUIR
LIS

PAGMRG
LIS

PARAG
LIS

REPEAT
LIS