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

```
_$2

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

RLI
RNO
RNO
RTY
SAV
STR
STR
STR
STR
STR
STR

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

FILEID**OFT

```
  000000    FFFFFFFFFF  TTTTTTTTTT
  000000    FFFFFFFFFF  TTTTTTTTTT
00      00  FF              TT
00      00  FF              TT
00      00  FF              TT
00      00  FF              TT
00      00  FFFFFFFF        TT
00      00  FFFFFFFF        TT
00      00  FF              TT
00      00  FF              TT
00      00  FF              TT          ....
00      00  FF              TT          ....
  000000    FF              TT          ....
  000000    FF              TT          ....

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
```

OUT
V04

```
;    1          0001  0 %TITLE 'Make changes to flag table'
;    2          0002  0 MODULE oft (    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 ! FACILITY:      DSR (Digital Standard RUNOFF) / DSRPLUS
;   33          0033  1 !
;   34          0034  1 ! ABSTRACT: Handles all .FLAGS and .NO FLAGS commands
;   35          0035  1 !
;   36          0036  1 !
;   37          0037  1 ! ENVIRONMENT: Transportable
;   38          0038  1 !
;   39          0039  1 ! AUTHOR: R.W.Friday     CREATION DATE: May, 1978
;   40          0040  1 !
```

```
;    42          0041  1 %SBTTL 'Revision History'
;    43          0042  1 !
;    44          0043  1 ! MODIFIED BY:
;    45          0044  1 !
;    46          0045  1 !       012     REM00012        Ray Marshall    17-November-1983
;    47          0046  1 !               Modified the external definition of ATABLE to use the new
;    48          0047  1 !                 macro ATABLE_DEFINITION defined in ATCODE.REQ.
;    49          0048  1 !
;    50          0049  1 !       011     KFA00011        Ken Alden       24-Jun-1983
;    51          0050  1 !               Fixed bug that kept existing flags from being 'NATE'd.
;    52          0051  1 !
;    53          0052  1 !       010     KFA00010        Ken Alden       21-Mar-1983
;    54          0053  1 !               No more Passthrough flag.
;    55          0054  1 !
;    56          0055  1 !       009     KFA00009        Ken Alden       15-Mar-1983
;    57          0056  1 !               For DSRPLUS: added functionality for passthrough flag.
;    58          0057  1 !
;    59          0058  1 !       008     KAD00008        Keith Dawson    07-Mar-1983
;    60          0059  1 !               Global edit of all modules. Updated module names, idents,
;    61          0060  1 !               copyright dates. Changed require files to BLISS library.
;    62          0061  1 !
;    63          0062  1 !--
```

```
;    65          0063  1 %SBTTL 'Module Level Declarations'
;    66          0064  1
;    67          0065  1 !
;    68          0066  1 ! TABLE OF CONTENTS:
;    69          0067  1 !
;    70          0068  1 !
;    71          0069  1 ! INCLUDE FILES:
;    72          0070  1 !
;    73          0071  1
;    74          0072  1 LIBRARY 'NXPORT:XPORT';                 ! XPORT Library
;    75          0073  1 REQUIRE 'REQ:RNODEF';                   ! RUNOFF variant definitions
;    76          0204  1
;    77        U 0205  1 %IF DSRPLUS %THEN
;    78        U 0206  1 LIBRARY 'REQ:DPLLIB';                   ! DSRPLUS BLISS Library
;    79          0207  1 %ELSE
;    80          0208  1 LIBRARY 'REQ:DSRLIB';                   ! DSR BLISS Library
;    81          0209  1 %FI
;    82          0210  1
;    83          0211  1 !
;    84          0212  1 ! MACROS:
;    85          0213  1 !
;    86          0214  1 !
;    87          0215  1 ! EQUATED SYMBOLS:
;    88          0216  1 !
;    89          0217  1 !
;    90          0218  1 ! OWN STORAGE:
;    91          0219  1 !
;    92          0220  1 !
;    93          0221  1 ! EXTERNAL REFERENCES:
;    94          0222  1 !
;    95          0223  1
;    96          0224  1 EXTERNAL LITERAL                                ! Error messages
;    97          0225  1     RNFFAU;
;    98          0226  1
;    99          0227  1 EXTERNAL
;   100          0228  1     atable : atable_definition, ! Action table.  Used to identify what type of
;   101          0229  1                                 ! action is to be taken on encountering any
;   102          0230  1                                 ! given character.
;   103          0231  1     FLGT : FLAG_TABLE,
;   104          0232  1     GCA : GCA_DEFINITION,
;   105          0233  1     SCA : SCA_DEFINITION;
;   106          0234  1
;   107          0235  1 EXTERNAL
;   108          0236  1     FLTSO : VECTOR;
;   109          0237  1
;   110          0238  1 BIND
;   111          0239  1     SEARCH_ORDER = FLTSO : VECTOR;
;   112          0240  1
;   113          0241  1 EXTERNAL ROUTINE
;   114          0242  1     ERMS,
;   115          0243  1     FNDFLG,
;   116          0244  1     GETONE,
;   117          0245  1     NATE;
```

```
;  119          0246  1 GLOBAL ROUTINE oft (handler_code, which_flag) : NOVALUE =
;  120          0247  1
;  121          0248  1 !++
;  122          0249  1 ! FUNCTIONAL DESCRIPTION:
;  123          0250  1 !
;  124          0251  1 !       See the ABSTRACT, above.
;  125          0252  1 !
;  126          0253  1 ! FORMAL PARAMETERS:
;  127          0254  1 !
;  128          0255  1 !       HANDLER_CODE indicates which command is to be processed.
;  129          0256  1 !       WHICH_FLAG indicates to which flag the command applies.
;  130          0257  1 !
;  131          0258  1 ! IMPLICIT INPUTS:
;  132          0259  1 !
;  133          0260  1 !       This routine relies on the fact that the HANDLER_CODEs
;  134          0261  1 !       for .FLAG commands are all less than the HANDLER_CODEs
;  135          0262  1 !       for .NO FLAG commands.
;  136          0263  1 !
;  137          0264  1 ! IMPLICIT OUTPUTS:     None
;  138          0265  1 !
;  139          0266  1 ! ROUTINE VALUE:
;  140          0267  1 ! COMPLETION CODES:     None
;  141          0268  1 !
;  142          0269  1 ! SIDE EFFECTS: None
;  143          0270  1 !
;  144          0271  1 !--
;  145          0272  1
;  146          0273  2     BEGIN
;  147          0274  2     !Flag enabling and disabling can occur at two different, independent
;  148          0275  2     !levels.  These levels correspond to the
;  149          0276  2     !   1) .FLAGS ALL and .NO FLAGS ALL commands, and the
;  150          0277  2     !   2) .FLAG <flag name> and .NO FLAG <flag name> commands.
;  151          0278  2     !Associated with each flag is a TRUE/FALSE setting that indicates
;  152          0279  2     !whether or not it should be recognized.  And associated with each
;  153          0280  2     !flag is also a character that defines which input character is to
;  154          0281  2     !be recognized as representing that flag. Assume that you have some
;  155          0282  2     !character 'X' that represents the UNDERLINE flag. 'X' will be recognized
;  156          0283  2     !as a flag only if.
;  157          0284  2     !   1) The user as said .FLAGS ALL (which is the default)
;  158          0285  2     !   **AND**
;  159          0286  2     !   2) The user has issued a .FLAG UNDERLINE command.
;  160          0287  2     !Note that the user may issue these commands at any time, and one
;  161          0288  2     !command does not imply the other. For example, if he has said .NO FLAGS
;  162          0289  2     !ALL, then 'X' will not be recognized as a flag, even if he says .FLAG
;  163          0290  2     !UNDERLINE; if he says .FLAG UNDERLINE when .NO FLAGS ALL is in effect,
;  164          0291  2     !RUNOFF simply remembers the fact that the UNDERLINE flag is enabled, and
;  165          0292  2     !makes it possible to recognize that flag after the user says .FLAGS ALL.
;  166          0293  2     !
;  167          0294  2     !The implementation of this is as follows:
;  168          0295  2     !Normally, each position in ATABLE tells what a particular character is.
;  169          0296  2     !If a character represents a flag the corresponding ATABLE entry notes
;  170          0297  2     !that fact.  Otherwise, the ATABLE entry says something else.
;  171          0298  2     !When the user says .NO FLAG UNDERLINE (above example) the ATABLE
;  172          0299  2     !entry corresponding to the character 'X' is changed to indicate
;  173          0300  2     !that 'X' is a normal character.  When he says .FLAG UNDERLINE,
;  174          0301  2     !the ATABLE entry is reset to indicate that 'X' represents a flag.
;  175          0302  2     !If the user says .NO FLAGS ALL, then again, the ATABLE entry (and
```

```
;  176          0303  2      !those for all other applicable flags) is set to inhibit flag recognition.
;  177          0304  2      !However, there is another difference between .NO FLAG UNDERLINE
;  178          0305  2      !and .NO FLAGS ALL.  The FLAG TABLE indicates the actual setting
;  179          0306  2      !of each flag.  The .FLAGS ALL and .NO FLAGS ALL commands do not change
;  180          0307  2      !those settings; they change ATABLE only.
;  181          0308  2      !The .FLAG UNDERLINE command may not only change ATABLE (if the user
;  182          0309  2      !said .FLAGS ALL) but the FLAG TABLE as well.
;  183          0310  2
;  184          0311  2      SELECTONE .HANDLER_CODE OF
;  185          0312  2          SET
;  186          0313  2
;  187          0314  2          [h_no_flags_all] :
;  188          0315  3              BEGIN
;  189          0316  3
;  190          0317  3              INCR i FROM 0 TO .search_order [-1] - 1 DO
;  191          0318  4                  BEGIN
;  192          0319  4                  !Disable recognition of only those flags that
;  193          0320  4                  !are marked as enabled.
;  194          0321  4
;  195          0322  4                  IF .flgt [.search_order [.i], flag_enabled]
;  196          0323  4                  THEN
;  197          0324  4                  !Set ATABLE entry to 'normal' setting.
;  198          0325  4                      nate (.flgt [.search_order [.i], flag_character]);
;  199          0326  3                  END;
;  200          0327  3
;  201          0328  3              !Remember that flags are not to be recognized.
;  202          0329  3              sca_flags = false;
;  203          0330  3              !Remember the last .FLAGS ALL/.NO FLAGS ALL command.
;  204          0331  3              gca_flag_cmd = h_no_flags_all;
;  205          0332  2              END;
;  206          0333  2
;  207          0334  2          [H_FLAGS_ALL] :
;  208          0335  3              BEGIN
;  209          0336  3
;  210          0337  3              INCR i FROM 0 TO .search_order [-1] - 1 DO
;  211          0338  4                  BEGIN
;  212          0339  4                  !Enable recognition of only those flags that are
;  213          0340  4                  !marked as enabled.
;  214          0341  4                  IF .flgt [.search_order [.i], flag_enabled]
;  215          0342  4                  THEN
;  216          0343  4                  !Mark corresponding ATABLE entry as representing a flag.
;  217          0344  4                      atable [.flgt [.search_order [.i], flag_character]] = a_flag;
;  218          0345  3                  END;
;  219          0346  3
;  220          0347  3              !Remember that flags are to be recognized.
;  221          0348  3              sca_flags = true;
;  222          0349  3              !Remember last .FLAGS ALL/.NO FLAGS ALL command.
;  223          0350  3              gca_flag_cmd = h_flags_all;
;  224          0351  2              END;
;  225          0352  2
;  226          0353  2          [OTHERWISE] :
;  227          0354  3              BEGIN
;  228          0355  3              IF .HANDLER_CODE LSS H_NO_FLAGS_ALL
;  229        U 0356  3 %IF DSRPLUS %THEN
;  230        U 0357  3                  !Special case for DSRPLUS: FLAG NOFERMUTE command's handler
;  231        U 0358  3                  !code is not in the same range as the others.
;  232        U 0359  3                  OR
```

```
;  233        U 0360  3                .handler_code eql h_flags_nopermu
;  234        U 0361  3
;  235          0362  3 %FI
;  236          0363  3             THEN
;  237          0364  4                 BEGIN                          !It's a .FLAGS type command
;  238          0365  4
;  239          0366  4                 LOCAL
;  240          0367  4                     hold_khar,
;  241          0368  4                     alternate_flag;        !This will = WHICH_FLAG unless we are
;  242          0369  4                                            !processing SUBINDEX, in which case it
;  243          0370  4                                            !will then point to the INDEX flag.
;  244          0371  4                 IF .which_flag EQL sbx_flag
;  245          0372  4                     THEN alternate_flag = IND_FLAG
;  246          0373  4                     ELSE alternate_flag = .which_flag;
;  247          0374  4                 !Get the new flag character, if any.
;  248          0375  4                 hold_khar = getone (.which_flag, .alternate_flag);
;  249          0376  4
;  250          0377  4                 IF .hold_khar eql 0
;  251          0378  4                 THEN
;  252          0379  4                 !No new flag character.  Use present one.
;  253          0380  4                     hold_khar = .flgt [.which_flag, flag_character];
;  254          0381  4
;  255          0382  4                 !Check whether GETONE found a flag conflict. If so, complain
;  256          0383  4                 !and disallow the indexing command.
;  257          0384  4                 IF .hold_khar LSS 0
;  258          0385  4                 THEN
;  259          0386  5                     BEGIN
;  260          0387  5                     LOCAL
;  261          0388  5                         temp;
;  262          0389  5                     temp = -.hold_khar;
;  263          0390  5                     erms (rnffau, temp, 1);     !'Flag already in use'
;  264          0391  5                     RETURN;
;  265          0392  4                     END;
;  266          0393  4
;  267          0394  4                 IF NOT
;  268          0395  5                     (.which_flag EQL com_flag
;  269          0396  5                     OR .which_flag EQL con_flag
;  270          0397  5                     OR .which_flag EQL efo_flag
;  271          0398  5                     OR .which_flag EQL sbx_flag)
;  272          0399  4                     AND .sca_flags
;  273          0400  4                 THEN
;  274          0401  5                     BEGIN
;  275          0402  5                     !Set corresponding ATABLE entry to indicate that the
;  276          0403  5                     !old flag_character is now just a normal character.
;  277          0404  5                     IF .flgt [.which_flag, flag_enabled] EQL true  !Flag was enabled.
;  278          0405  5                     THEN
;  279          0406  5                         nate (.flgt [.which_flag, flag_character]);
;  280          0407  5                     !
;  281          0408  5                     !Make ATABLE entry show the new character is a flag.
;  282          0409  5                     atable [.hold_khar] = a_flag;
;  283          0410  4                     END;
;  284          0411  4
;  285          0412  4                 flgt [.which_flag, flag_character] = .hold_khar;  !Save new flag character.
;  286          0413  4                 flgt [.which_flag, flag_enabled] = true;          !Flag is enabled.
;  287          0414  4                 END
;  288          0415  3             ELSE
;  289          0416  4                 BEGIN                          !It's a .NO FLAGS type command
```

```
;  290     0417  4                flgt [.which_flag, flag_enabled] = false;       !Flag is disabled.
;  291     0418  4
;  292     0419  4                IF NOT
;  293     0420  5                    (.which_flag EQL com_flag
;  294     0421  5                    OR .which_flag EQL con_flag
;  295     0422  5                    OR .which_flag EQL efo_flag
;  296     0423  5                    OR .which_flag EQL sbx_flag)
;  297     0424  4                THEN
;  298     0425  4                    IF                                  !Normal cases
;  299     0426  5                        (fndflg (.flgt[.which_flag,flag_character]) eql flag_count+1)
;  300     0427  4                        AND .sca_flags
;  301     0428  4                    THEN
;  302     0429  4                    !Set corresponding ATABLE entry to indicate it's a 'normal' character.
;  303     0430  4                        nate (.flgt [.which_flag, flag_character])
;  304     0431  4                    ELSE
;  305     0432  4                    !Because this character represents another
;  306     0433  4                    !flag as well, leave ATABLE alone so that
;  307     0434  4                    !the other flag will be recognized.
;  308     0435  4                        (0);
;  309     0436  3                END;
;  310     0437  3
;  311     0438  2            END;
;  312     0439  2        TES;
;  313     0440  1    END;                                                    !End of OFT


                                                .TITLE  OFT Make changes to flag table
                                                .IDENT  \V04-000\

                                                .EXTRN  RNFFAU, ATABLE, FLGT
                                                .EXTRN  GCA, SCA, FLTSO
                                                .EXTRN  ERMS, FNDFLG, GETONE
                                                .EXTRN  NATE

                                                .PSECT  $CODE$,NOWRT,2

                                    07FC 00000      .ENTRY  OFT, Save R2,R3,R4,R5,R6,R7,R8,R9,R10        ; 0246
                5A 00000000G   EF   9E 00002      MOVAB   ATABLE, R10                                  ;
                59 00000000G   EF   9E 00009      MOVAB   NATE, R9                                     ;
                58 00000000G   EF   9E 00010      MOVAB   SEARCH_ORDER-4, R8                           ;
                57 00000000G   EF   9E 00017      MOVAB   SCA+144, R7                                  ;
                56 00000000G   EF   9E 0001E      MOVAB   FLGT+72, R6                                  ;
                5E                  04  C2 00025      SUBL2   #4, SP                                   ;
                50         04       AC  D0 00028      MOVL    HANDLER_CODE, R0                         ; 0311
    0000007C    8F                  50  D1 0002C      CMPL    R0, #124                                 ; 0314
                                    28  12 00033      BNEQ    3$                                       ;
                53                  68  D0 00035      MOVL    SEARCH_ORDER-4, R3                       ; 0317
                52                  01  CE 00038      MNEGL   #1, I                                    ;
                                    10  11 0003B      BRB     2$                                       ;
                50         04   A842  D0 0003D  1$:    MOVL    SEARCH_ORDER[I], R0                      ; 0322
                06         B8   A640  E9 00042      BLBC    FLGT[R0], 2$                             ;
                                6640  DD 00047      PUSHL   FLGT+72[R0]                              ; 0325
                69                  01  FB 0004A      CALLS   #1, NATE                                 ;
    EC          52                  53  F2 0004D  2$:    AOBLSS  R3, I, 1$                        ; 0317
                           00       B7  D4 00051      CLRL    @SCA+144                                 ; 0329
    00000000G   FF         7C       8F  9A 00054      MOVZBL  #124, @GCA+132                           ; 0331
                                    04  0005C      RET                                              ; 0311
```

```
                00000046   8F              50  D1 0005D 3$:    CMPL    R0, #70                                 ; 0334
                                           28  12 00064       BNEQ    6$
                           51              01  CE 00066       MNEGL   #1, I                                   ; 0337
                                           12  11 00069       BRB     5$
                           50        04 A841  D0 0006B 4$:    MOVL    SEARCH_ORDER[I], R0                     ; 0341
                           08        B8 A640  E9 00070       BLBC    FLGT[R0], 5$
                           50           6640  D0 00075       MOVL    FLGT+72[R0], R0                         ; 0344
                         6A40              03  90 00079       MOVB    #3, ATABLE[R0]
                EA         51              68  F2 0007D 5$:    AOBLSS  SEARCH_ORDER-4, I, 4$                   ; 0337
                     00    B7              01  D0 00081       MOVL    #1, @SCA+144                            ; 0348
           00000000G   FF        46    8F  9A 00085       MOVZBL  #70, @GCA+132                           ; 0350
                                           04  0008D       RET                                             ; 0311
                           53        08    AC  D0 0008E 6$:    MOVL    WHICH_FLAG, R3                          ; 0371
                52         53              02  78 00092       ASHL    #2, R3, R2                              ; 0412
                0000007C   8F              50  D1 00096       CMPL    R0, #124                                ; 0355
                                           72  18 0009D       BGEQ    13$
                                           55  D4 0009F       CLRL    R5                                      ; 0371
                           0D              53  D1 000A1       CMPL    R3, #13
                                           07  12 000A4       BNEQ    7$
                                           55  D6 000A6       INCL    R5
                           50              0A  D0 000A8       MOVL    #10, ALTERNATE_FLAG                     ; 0372
                                           03  11 000AB       BRB     8$
                           50              53  D0 000AD 7$:    MOVL    R3, ALTERNATE_FLAG                      ; 0373
                                           50  DD 000B0 8$:    PUSHL   ALTERNATE_FLAG                          ; 0375
                                           53  DD 000B2       PUSHL   R3
           00000000G       EF              02  FB 000B4       CALLS   #2, GETONE
                           54              50  D0 000BB       MOVL    R0, HOLD_KHAR
                                           04  12 000BE       BNEQ    9$                                      ; 0377
                           54           6643  D0 000C0       MOVL    FLGT+72[R3], HOLD_KHAR                  ; 0380
                                           16  18 000C4 9$:    BGEQ    10$                                     ; 0384
                           6E              54  CE 000C6       MNEGL   HOLD_KHAR, TEMP                         ; 0389
                                           01  DD 000C9       PUSHL   #1                                      ; 0390
                                     04    AE  9F 000CB       PUSHAB  TEMP
                              00000000G    8F  DD 000CE       PUSHL   #RNFFAU
      00000000G                EF          03  FB 000D4       CALLS   #3, ERMS
                                           04  000DB       RET                                             ; 0386
                           0E              53  D1 000DC 10$:    CMPL    R3, #14                                 ; 0395
                                           22  13 000DF       BEQL    12$
                           02              53  D1 000E1       CMPL    R3, #2                                  ; 0396
                                           1D  13 000E4       BEQL    12$
                           01              53  D1 000E6       CMPL    R3, #1                                  ; 0397
                                           18  13 000E9       BEQL    12$
                           15              55  E8 000EB       BLBS    R5, 12$                                 ; 0398
                           11        00    B7  E9 000EE       BLBC    @SCA+144, 12$                           ; 0399
                           01        B8 A643  D1 000F2       CMPL    FLGT[R3], #1                            ; 0404
                                           06  12 000F7       BNEQ    11$
                                        6643  DD 000F9       PUSHL   FLGT+72[R3]                             ; 0406
                           69              01  FB 000FC       CALLS   #1, NATE
                         6A44              03  90 000FF 11$:    MOVB    #3, ATABLE[HOLD_KHAR]                   ; 0409
                                        6642  9F 00103 12$:    PUSHAB  FLGT+72[R2]                             ; 0412
                           9E              54  D0 00106       MOVL    HOLD_KHAR, @(SP)+
                                     B8 A642  9F 00109       PUSHAB  FLGT[R2]                                ; 0413
                           9E              01  D0 0010D       MOVL    #1, @(SP)+
                                           04  00110       RET                                             ; 0355
                                     B8 A642  9F 00111 13$:    PUSHAB  FLGT[R2]                                ; 0417
                                           6E  D4 00115       CLRL    @(SP)+
                           0E              53  D1 00117       CMPL    R3, #14                                 ; 0420
```

```
                                    2C  13 0011A         BEQL    14$                                             ;
                                02  53  D1 0011C         CMPL    R3, #2                                          ; 0421
                                    27  13 0011F         BEQL    14$                                             ;
                                01  53  D1 00121         CMPL    R3, #1                                          ; 0422
                                    22  13 00124         BEQL    14$                                             ;
                                0D  53  D1 00126         CMPL    R3, #13                                         ; 0423
                                    1D  13 00129         BEQL    14$                                             ;
                                      6642  9F 0012B     PUSHAB  FLGT+72[R2]                                     ; 0426
                                        9E  DD 0012E     PUSHL   @(SP)+                                          ;
                         00000000G  EF      01  FB 00130 CALLS   #1, FNDFLG                                      ;
                                13  50  D1 00137         CMPL    R0, #19                                         ;
                                    0C  12 0013A         BNEQ    14$                                             ;
                                08  00  B7  E9 0013C     BLBC    @SCA+144, 14$                                   ; 0427
                                      6642  9F 00140     PUSHAB  FLCT+72[R2]                                     ; 0430
                                        9E  DD 00143     PUSHL   @(SP)+                                          ;
                                69      01  FB 00145     CALLS   #1, NATE                                        ;
                                            04 00148 14$:.  RET                                                  ; 0440

; Routine Size:  329 bytes,    Routine Base:  $CODE$ + 0000

;  314          0441  1
;  315          0442  1 END                                         !End of module
;  316          0443  0 ELUDOM




;                                   PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  $CODE$                         329  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         22         1          86        00:00.3



;                                     COMMAND QUALIFIERS

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

```
; Size:          329 code + 0 data bytes
; Run Time:        00:07.5
; Elapsed Time:    00:19.7
; Lines/CPU Min:    3567
; Lexemes/CPU-Min: 13538
; Memory Used:  92 pages
; Compilation Complete
```

NEWPAG
LIS

NODOPX
LIS

OFT
LIS

OUTTXT
LIS

NDXVRS
LIS

NOTE
LIS

OUTLIN
LIS

NM
LIS

PACK
LIS

OUTXHR
LIS

NDXXTN
LIS

OUTCHA
LIS

OUTHDR
LIS