```
RRRRRRRRRRRR     UUU         UUU  NNN         NNN     OOOOOOOOO     FFFFFFFFFFFFFFF  FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN         NNN     OOOOOOOOO     FFFFFFFFFFFFFFF  FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN         NNN     OOOOOOOOO     FFFFFFFFFFFFFFF  FFFFFFFFFFFFFFF
RRR         RRR  UUU         UUU  NNN         NNN  OOO         OOO  FFF              FFF
RRR         RRR  UUU         UUU  NNN         NNN  OOO         OOO  FFF              FFF
RRR         RRR  UUU         UUU  NNN         NNN  OOO         OOO  FFF              FFF
RRR         RRR  UUU         UUU  NNNNNN      NNN  OOO         OOO  FFF              FFF
RRR         RRR  UUU         UUU  NNNNNN      NNN  OOO         OOO  FFF              FFF
RRR         RRR  UUU         UUU  NNNNNN      NNN  OOO         OOO  FFF              FFF
RRRRRRRRRRRR     UUU         UUU  NNN   NNN   NNN  OOO         OOO  FFFFFFFFFFFF     FFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN   NNN   NNN  OOO         OOO  FFFFFFFFFFFF     FFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN   NNN   NNN  OOO         OOO  FFFFFFFFFFFF     FFFFFFFFFFFF
RRR   RRR        UUU         UUU  NNN      NNNNNN  OOO         OOO  FFF              FFF
RRR   RRR        UUU         UUU  NNN      NNNNNN  OOO         OOO  FFF              FFF
RRR   RRR        UUU         UUU  NNN      NNNNNN  OOO         OOO  FFF              FFF
RRR      RRR     UUU         UUU  NNN         NNN  OOO         OOO  FFF              FFF
RRR      RRR     UUU         UUU  NNN         NNN  OOO         OOO  FFF              FFF
RRR      RRR     UUU         UUU  NNN         NNN  OOO         OOO  FFF              FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN         NNN     OOOOOOOOO     FFF              FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN         NNN     OOOOOOOOO     FFF              FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN         NNN     OOOOOOOOO     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
```

FILEID**OUTLIN

H 12

```
  OOOOOO    UU      UU  TTTTTTTTTT  LL            IIIIII    NN      NN
  OOOOOO    UU      UU  TTTTTTTTTT  LL            IIIIII    NN      NN
OO      OO  UU      UU      TT      LL              II      NN      NN
OO      OO  UU      UU      TT      LL              II      NN      NN
OO      OO  UU      UU      TT      LL              II      NNNN    NN
OO      OO  UU      UU      TT      LL              II      NNNN    NN
OO      OO  UU      UU      TT      LL              II      NN  NN  NN
OO      OO  UU      UU      TT      LL              II      NN  NN  NN
OO      OO  UU      UU      TT      LL              II      NN    NNNN
OO      OO  UU      UU      TT      LL              II      NN    NNNN
OO      OO  UU      UU      TT      LL              II      NN      NN      ....
OO      OO  UU      UU      TT      LL              II      NN      NN      ....
  OOOOOO    UUUUUUUUUU      TT      LLLLLLLLLL    IIIIII    NN      NN      ....
  OOOOOO    UUUUUUUUUU      TT      LLLLLLLLLL    IIIIII    NN      NN      ....

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 'directs output or move of TSF'
;     2       0002 0 MODULE OUTLIN ( 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 !       Either saves information described by TSF or causes it to
;    38       0038 1 !       be output immediately.
;    39       0039 1 !
;    40       0040 1 ! ENVIRONMENT:  Transportable
;    41       0041 1 !
;    42       0042 1 ! AUTHOR:       R.W.Friday      CREATION DATE:  May, 1978
;    43       0043 1
;    45       0044  1 %SBTTL 'Revision History'
;    46       0045  1 ! MODIFIED BY:
;    47       0046  1 !
;    48       0047  1 !          019      KAD00019        Keith Dawson    9-May-1983
;    49       0048  1 !                   Remove support for .DX, .PX.
;    50       0049  1 !
;    51       0050  1 !          018      RER00018        Ron Randall     20-Mar-1983
;    52       0051  1 !                   For DSRPLUS:  Added code for topnotes.
;    53       0052  1 !
;    54       0053  1 !          017      KAD00017        Keith Dawson    20-Mar-1983
;    55       0054  1 !                   Removed LN01 conditionals and all references to .BIX
;    56       0055  1 !                   and .BTC files.
;    57       0056  1 !
;    58       0057  1 !          016      KAD00016        Keith Dawson    07-Mar-1983
;    59       0058  1 !                   Global edit of all modules. Updated module names, idents,
;    60       0059  1 !                   copyright dates. Changed require files to BLISS library.
;    61       0060  1 !--
;    62       0061  1
;      64          0062   1 %SBTTL 'Module Level Declarations'
;      65          0063   1 !
;      66          0064   1 ! TABLE OF CONTENTS:
;      67          0065   1 !
;      68          0066   1 FORWARD ROUTINE
;      69          0067   1     OUTLIN       : NOVALUE,
;      70          0068   1     OUTCRG       : NOVALUE,
;      71          0069   1     OUTJ         : NOVALUE,
;      72          0070   1     OUTNJ        : NOVALUE,
;      73          0071   1     OUTPAS       : NOVALUE;
;      74          0072   1 !
;      75          0073   1 ! INCLUDE FILES:
;      76          0074   1 !
;      77          0075   1 LIBRARY 'NXPORT:XPORT';                        ! XPORT Library
;      78          0076   1 REQUIRE 'REQ:RNODEF';                          ! RUNOFF variant definitions
;      79          0207   1
;      80        U 0208   1 %IF DSRPLUS %THEN
;      81        U 0209   1 LIBRARY 'REQ:DPLLIB';                          ! DSRPLUS BLISS Library
;      82          0210   1 %ELSE
;      83          0211   1 LIBRARY 'REQ:DSRLIB';                          ! DSR BLISS Library
;      84          0212   1 %FI
;      85          0213   1
;      86          0214   1 !
;      87          0215   1 ! EXTERNAL REFERENCES:
;      88          0216   1 !
;      89          0217   1 EXTERNAL
;      90          0218   1     BRNOOB       : $XPO_IOB (),
;      91          0219   1     FOOREC       : FOOREC_DEFINITION,
;      92          0220   1     FNCT         : FNCT_DEFINITION,
;      93          0221   1     FRA          : FIXED_STRING,
;      94          0222   1     GCA          : GCA_DEFINITION,
;      95          0223   1     MRA          : REF FIXED_STRING,
;      96          0224   1     OUTOPT       : OUTOPT_DEFINE,
;      97          0225   1     SCA          : SCA_DEFINITION,
;      98          0226   1     TSF          : TSF_DEFINITION;
;      99          0227   1
;     100        U 0228   1 %IF DSRPLUS %THEN
;     101        U 0229   1 EXTERNAL
;     102        U 0230   1     TOPNOT       : TN_DEFINITION,
;     103        U 0231   1     TNREC        : TNREC_DEFINITION;
;     104        U 0232   1
;     105        U 0233   1 EXTERNAL ROUTINE
;     106        U 0234   1     TNFIL;
;     107          0235   1 %FI
;     108          0236   1
;     109          0237   1 EXTERNAL LITERAL
;     110          0238   1     RNFCJL,
;     111          0239   1     RNFLOC;
;     112          0240   1
;     113          0241   1 EXTERNAL ROUTINE
;     114          0242   1     endwrd,     erm,     foofil,        lout,
;     115          0243   1     putndy,     puttxt, scl,           unpus;
;     116          0244   1 !!    XOUT;
;     117          0245   1
;  119          0246  1 %sbttl 'OUTLIN -- output full MRA.'
;  120          0247  1 GLOBAL ROUTINE outlin (justify) : NOVALUE =
;  121          0248  1
;  122          0249  1 !++
;  123          0250  1 ! FUNCTIONAL DESCRIPTION:
;  124          0251  1 !
;  125          0252  1 !       OUTLIN is called when MRA is full, and the text should be output. It
;  126          0253  1 !       routes normal text lines directly to LOUT, for output. Other records
;  127          0254  1 !       are passed to other processors.  For example, footnote records get
;  128          0255  1 !       saved in the footnote file, topnote records get saved in the topnote
;  129          0256  1 !       file, and indexing records get sent to the indexing routines.
;  130          0257  1 !
;  131          0258  1 ! FORMAL PARAMETERS:
;  132          0259  1 !
;  133          0260  1 !       justify - Indicates whether or not the line should be justified.
;  134          0261  1 !
;  135          0262  1 ! IMPLICIT INPUTS:      None
;  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  2     BEGIN
;  146          0273  2
;  147          0274  2     IF   .TSF_INDEX                      ! Is this TSF something for the index?
;  148          0275  2         OR .TSF_BTC                     ! Is this TSF something for the table of contents?
;  149          0276  2     THEN
;  150          0277  3         BEGIN                           ! This record describes an index or table of contents entry
;  151          0278  3         LOCAL
;  152          0279  3             IADDR,                      ! Address of text (as opposed to CH$PTR)
;  153          0280  3             INT_HL,                     ! Internal length of text.
;  154          0281  3             IPTR;                       ! CH$PTR to the text.
;  155          0282  3
;  156          0283  3         INT_HL = .TSF_INT_HL;           ! Save internal length of text.
;  157          0284  3         IPTR = .FS_START(MRA);          ! Save pointer to text.
;  158          0285  3         IADDR = (.FS_START(MRA))<0,%BPADDR,0> + %BLISS(BLISS36);
;  159          0286  3
;  160          0287  3         TSF_LINES = 0;                  ! This does not generate any text.
;  161          0288  3
;  162        U 0289  3 %IF DSRPLUS %THEN
;  163        U 0290  3         !
;  164        U 0291  3         ! Direct indexing and table of contents information from the topnote
;  165        U 0292  3         ! to the topnote file.  Such information from the main body of the
;  166        U 0293  3         ! document goes directly to the processing routine.
;  167        U 0294  3         !
;  168        U 0295  3         IF .TN_COLLECTING               ! If collecting topnotes,
;  169        U 0296  3         THEN
;  170        U 0297  3             TNFIL (TN_WRITE)            ! save information in topnote file.
;  171        U 0298  3         ELSE
;  172        U 0299  3             BEGIN
;  173          0300  3 %FI
;  174          0301  3
;  175          0302  3         !
;  176          0303  3             ! Direct indexing and table of contents information from the footnote
;  177          0304  3             ! to the footnote file. Such information from the main body of the
;  178          0305  3             ! document goes directly to the processing routine.
;  179          0306  3             !
;  180          0307  3             IF .FNCT_COLLECTING                 ! If collecting footnotes,
;  181          0308  3             THEN
;  182          0309  3                 FOOFIL (FOO_WRIT)               ! save information in footnote file.
;  183          0310  3             ELSE
;  184          0311  4                 BEGIN
;  185          0312  4
;  186          0313  4                 ! If user wants to debug the index or table of contents,
;  187          0314  4                 ! regurgitate the information.
;  188          0315  5                 IF (NOT .GCA_SKIP_OUT)
;  189          0316  4                 THEN
;  190          0317  5                     BEGIN
;  191          0318  5
;  192          0319  5                     ! User wants to see information on this document page.
;  193          0320  5                     IF .GCA_DEBUG_INDEX
;  194          0321  5                         AND .TSF_INDEX
;  195          0322  5                     THEN
;  196          0323  5                         ! Echo an index entry.
;  197          0324  5                         UNPUS (0)
;  198          0325  5                     ELSE
;  199          0326  6                         BEGIN
;  200          0327  6
;  201          0328  6                         IF .GCA_DEBUG_TOC
;  202          0329  6                             AND .TSF_BTC
;  203          0330  6                         THEN
;  204          0331  6                             ! Echo a table of contents entry.
;  205          0332  6                             UNPUS (1);
;  206          0333  6
;  207          0334  5                         END;
;  208          0335  5
;  209          0336  4                     END;
;  210          0337  4
;  211          0338  4                 ! Send index entries to the intermediate file, only if the user
;  212          0339  4                 ! said /INTERMEDIATE.
;  213          0340  4 !!              IF NOT .GCA_BIX
;  214          0341  4 !!              THEN
;  215          0342  4 !!                  IF .TSF_INDEX
;  216          0343  4 !!                  THEN
;  217          0344  4 !!                      XOUT (.INT_HL, .IPTR, .TSF_FIRST_XTN, .TSF_H_BARS);
;  218          0345  4
;  219          0346  5                 IF (.gca_bix AND .tsf_index)
;  220          0347  4                 THEN
;  221          0348  4                     putndy (.int_hl, .iaddr, .tsf_first_xtn, .tsf_h_bars);
;  222          0349  4
;  223          0350  4                 ! Send table of contents records to the intermediate file.
;  224          0351  4                 IF .tsf_btc
;  225          0352  4                 THEN
;  226          0353  4                     puttxt (.int_hl, .iptr, .tsf_major, .tsf_minor);
;  227          0354  4
;  228          0355  3                 END;
;  229          0356  3
;  230        U 0357  3 %IF DSRPLUS %THEN
;  231        U 0358  3                 END;
;  232          0359  3 %FI
;  233         0360  3
;  234         0361  3          RETURN;
;  235         0362  2          END;
;  236         0363  2
;  237         0364  2  !+
;  238         0365  2  ! This is a record full of "normal" text to be output.
;  239         0366  2  !-
;  240         0367  2
;  241       U 0368  2  %IF DSRPLUS %THEN
;  242       U 0369  2      IF  NOT .TN_EXPANDING                    ! Expanding topnotes?
;  243       U 0370  2      THEN
;  244       U 0371  2          BEGIN
;  245         0372  2  %FI
;  246         0373  2
;  247         0374  2      IF  NOT .FNCT_EXPANDING                  ! Expanding footnotes?
;  248         0375  2      THEN
;  249         0376  2      ! Set up justification for all lines except those that are
;  250         0377  2      ! being fetched from the footnote file. For these records,
;  251         0378  2      ! the information was computed before they were written to
;  252         0379  2      ! the footnote temporary file.
;  253         0380  3          BEGIN
;  254         0381  3          TSF_JUSTIFY  = .JUSTIFY;
;  255         0382  3          TSF_JUST_ALG = .TSF_JUST_ALG + 1;        ! Justification algorithm feedback for next time.
;  256         0383  3          TSF_PADDING  = .SCA_RM - .TSF_EXT_HL;    ! Space count for padding.
;  257         0384  2          END;
;  258         0385  2
;  259       U 0386  2  %IF DSRPLUS %THEN
;  260       U 0387  2          END;
;  261       U 0388  2
;  262       U 0389  2      IF  NOT .TN_COLLECTING
;  263       U 0390  2      THEN
;  264       U 0391  2          BEGIN
;  265         0392  2  %FI
;  266         0393  2
;  267         0394  2      IF  .TSF_PADDING LSS 0 AND NOT .FNCT_COLLECTING
;  268         0395  2      THEN
;  269         0396  3          BEGIN
;  270         0397  3          ! This can happen only if a word is encountered that does
;  271         0398  3          ! not fit onto the line between the margins. It can also
;  272         0399  3          ! happen as a side effect of breaking a word that does not
;  273         0400  3          ! fit in TSF (see ENDCHR, where this is done).
;  274         0401  3          ! The effect of the error handling is that the line is
;  275         0402  3          ! output without justification, but with nothing removed.
;  276         0403  3          ! The resulting line may have text exceeding the right margin.
;  277         0404  3          ERM (RNFCJL, 0, 0);
;  278         0405  3
;  279       U 0406  3  %IF DSRPLUS %THEN
;  280       U 0407  3          IF  NOT .TN_EXPANDING                ! Expanding topnotes?
;  281       U 0408  3          THEN
;  282       U 0409  3              BEGIN
;  283         0410  3  %FI
;  284         0411  3
;  285         0412  3          IF  NOT .FNCT_EXPANDING              ! Expanding footnotes?
;  286         0413  3          THEN
;  287         0414  3              ! Output line and page only if this is not happening
;  288         0415  3              ! inside a footnote. If it's happening inside a footnote
;  289         0416  3              ! then the line and page numbers will be the wrong ones.
;   290           0417  3                 ERM (RNFLOC, 0, 0);
;   291           0418  3
;   292           0419  3             TSF_PADDING = 0;
;   293           0420  3
;   294         U 0421  3 %IF DSRPLUS %THEN
;   295         U 0422  3                 END;
;   296           0423  3 %FI
;   297           0424  3
;   298           0425  2             END;
;   299           0426  2
;   300         U 0427  2 %IF DSRPLUS %THEN
;   301         U 0428  2             END;
;   302         U 0429  2
;   303         U 0430  2         IF  NOT .TN_COLLECTING              ! Collecting topnotes?
;   304         U 0431  2         THEN
;   305         U 0432  2             BEGIN
;   306           0433  2 %FI
;   307           0434  2
;   308           0435  2         IF  NOT .FNCT_COLLECTING            ! Collecting footnotes?
;   309           0436  2         THEN
;   310           0437  2             ! Output some text directly to the document.
;   311           0438  3             BEGIN
;   312           0439  3             FS_INIT (FRA);
;   313           0440  3
;   314         U 0441  3 %IF DSRPLUS %THEN
;   315         U 0442  3             GCA_LINE_PEND = 0;      ! This line is going to be output, pend no more.
;   316           0443  3 %FI
;   317           0444  3
;   318           0445  3             LOUT ();
;   319           0446  3             END
;   320           0447  2         ELSE
;   321           0448  2             ! Save this record in the footnote file for use later.
;   322           0449  3             BEGIN
;   323           0450  3
;   324           0451  3             ! Compute the number of lines represented by this record.
;   325           0452  3             ! TSF_LINES currently contains a count of the number of
;   326           0453  3             ! lines that will be generated when the code generated by
;   327           0454  3             ! the Gxxxx routines (see module GCODE) is "executed".
;   328           0455  3             IF .TSF_UND AND .OUTOPT_UND_SEP
;   329           0456  3             THEN
;   330           0457  3             ! Add an extra line if underlining is done by putting something
;   331           0458  3             ! on an extra line (e.g. dashes under the text).
;   332           0459  3                 TSF_LINES = .TSF_LINES + 1;
;   333           0460  3
;   334           0461  3             IF .TSF_EXT_HL NEQ 0
;   335           0462  3             THEN
;   336           0463  3                 TSF_LINES = .TSF_LINES + 1;
;   337           0464  3
;   338           0465  3             ! Now actually save the record.
;   339           0466  3             FOOFIL (FOO_WRIT);
;   340           0467  2             END;
;   341           0468  2
;   342         U 0469  2 %IF DSRPLUS %THEN
;   343         U 0470  2             END
;   344         U 0471  2         ELSE
;   345         U 0472  2             BEGIN
;   346         U 0473  2
;  347      U 0474  2           IF .TSF_UND AND .OUTOPT_UND_SEP
;  348      U 0475  2           THEN
;  349      U 0476  2               TSF_LINES = .TSF_LINES + 1;
;  350      U 0477  2
;  351      U 0478  2           IF .TSF_EXT_HL NEQ 0
;  352      U 0479  2           THEN
;  353      U 0480  2               TSF_LINES = .TSF_LINES + 1;
;  354      U 0481  2
;  355      U 0482  2           TNFIL (TN_WRITE);
;  356      U 0483  2           END;
;  357      U 0484  2
;  358      U 0485  2       IF  NOT .TN_EXPANDING                 ! Expanding topnotes?
;  359      U 0486  2       THEN
;  360      U 0487  2           BEGIN
;  361        0488  2 %FI
;  362        0489  2
;  363        0490  2       IF  NOT .FNCT_EXPANDING               ! Expanding footnotes?
;  364        0491  2       THEN
;  365        0492  2           ! Throw away leftovers from line just output.
;  366        0493  2           ! Note that it wouldn't hurt anything to always do this. It's just that
;  367        0494  2           ! it's a waste of time if footnotes are being output.
;  368        0495  3           BEGIN
;  369        0496  3           TSF_INT_HL = 0;
;  370        0497  3           TSF_EXT_HL = 0;
;  371        0498  3           TSF_INT_VL = 0;
;  372        0499  3           TSF_NBITS = 0;
;  373        0500  3           TSF_BARS = FALSE;                           ! Turn off change bars.
;  374        0501  3           TSF_H_BARS = FALSE;
;  375        0502  3           TSF_JUS_CNT = 0;
;  376        0503  3           TSF_FIRST_XTN = 0;
;  377        0504  3           TSF_LAST_XTN = 0;
;  378        0505  3           SCA_WRD_LST_JUS = 0;
;  379        0506  3           SCA_WRD_LST_UND = FALSE;
;  380        0507  3           SCA_WRD_LST_SP = 0;
;  381        0508  3           TSF_TEXT = FALSE;                           ! There's nothing there now.
;  382        0509  3           TSF_ADJUST = 0;
;  383        0510  3           TSF_NEXT_REG = 0;
;  384        0511  3           TSF_LINES = 0;
;  385        0512  3           TSF_FOOTW = 0;                              ! No footnotes attached to this line.
;  386        0513  3           ! Clean up the MRA by moving the last word in the buffer to the left.
;  387        0514  3           SCL ();
;  388        0515  2           END;
;  389        0516  2
;  390      U 0517  2 %IF DSRPLUS %THEN
;  391      U 0518  2           END;
;  392        0519  2 %FI
;  393        0520  2
;  394        0521  1       END;                                        ! End of OUTLIN


                                                                        .TITLE  OUTLIN directs output or move of TSF
                                                                        .IDENT  \V04-000\

                                                                        .EXTRN  BRNOOB, FOOREC, FNCT
                                                                        .EXTRN  FRA, GCA, MRA, OUTOPT
                                                                        .EXTRN  SCA, TSF, RNFCJL
                                                                        .EXTRN  RNFLOC, ENDWRD, ERM
```

```
                                                             .EXTRN  FOOFIL, LOUT, PUTNDY
                                                             .EXTRN  PUTTXT, SCL, UNPUS

                                                             .PSECT  $CODE$,NOWRT,2

                                          0FFC 00000         .ENTRY  OUTLIN, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,- ; 0247
                                                                     R11
                           5B 00000000G EF  9E 00002         MOVAB   ERM, R11                                  ;
                           5A 00000000G EF  9E 00009         MOVAB   FOOFIL, R10                               ;
                           59 00000000G EF  9E 00010         MOVAB   GCA+116, R9                               ;
                           58 00000000G EF  9E 00017         MOVAB   FRA, R8                                   ;
                           57 00000000G EF  9E 0001E         MOVAB   FNCT+20, R7                               ;
                           56 00000000G EF  9E 00025         MOVAB   TSF, R6                                   ;
                           52           66  D0 0002C         MOVL    TSF, R2                                   ; 0274
                           04        14 A2  E8 0002F         BLBS    20(R2), 1$                                ;
                           75        2C A2  E9 00033         BLBC    44(R2), 8$                                ; 0275
                           53           62  D0 00037 1$:     MOVL    (R2), INT_HL                              ; 0283
                           54 00000000G FF  D0 0003A         MOVL    @MRA, IPTR                                ; 0284
                           55 00000000G FF  D0 00041         MOVL    @MRA, IADDR                               ; 0285
                                     34 A2  D4 00048         CLRL    52(R2)                                    ; 0287
                           06           67  E9 0004B         BLBC    FNCT+20, 2$                               ; 0307
                                        04  DD 0004E         PUSHL   #4                                        ; 0309
                           6A           01  FB 00050         CALLS   #1, FOOFIL                                ;
                                            04 00053         RET                                               ;
                           1C        FC A9  E8 00054 2$:     BLBS    GCA+112, 5$                               ; 0315
                           08           69  E9 00058         BLBC    GCA+116, 3$                               ; 0320
                           04        14 A2  E9 0005B         BLBC    20(R2), 3$                                ; 0321
                                        7E  D4 0005F         CLRL    -(SP)                                     ; 0324
                                        0A  11 00061         BRB     4$                                        ;
             0D            69           01  E1 00063 3$:     BBC     #1, GCA+116, 5$                           ; 0328
                           09        2C A2  E9 00067         BLBC    44(R2), 5$                                ; 0329
                                        01  DD 0006B         PUSHL   #1                                        ; 0332
                 00000000G EF           01  FB 0006D 4$:     CALLS   #1, UNPUS                                 ;
             1D         08 A9           02  E1 00074 5$:     BBC     #2, GCA+124, 6$                           ; 0346
                           50           66  D0 00079         MOVL    TSF, R0                                   ;
                           16        14 A0  E9 0007C         BLBC    20(R0), 6$                                ;
                           50           66  D0 00080         MOVL    TSF, R0                                   ; 0348
7E     0080  C0            01           00  EF 00083         EXTZV   #0, #1, 128(R0), -(SP)                    ;
                                     38 A0  DD 0008A         PUSHL   56(R0)                                    ;
                                        28  BB 0008D         PUSHR   #^M<R3,R5>                                ;
                 00000000G EF           04  FB 0008F         CALLS   #4, PUTNDY                                ;
                           50           66  D0 00096 6$:     MOVL    TSF, R0                                   ; 0351
                           01        2C A0  E8 00099         BLBS    44(R0), 7$                                ;
                                            04 0009D         RET                                               ;
                           7E        38 A0  7D 0009E 7$:     MOVQ    56(R0), -(SP)                             ; 0353
                                        18  BB 000A2         PUSHR   #^M<R3,R4>                                ;
                 00000000G EF           04  FB 000A4         CALLS   #4, PUTTXT                                ;
                                            04 000AB         RET                                               ; 0277
                           15        04 A7  E8 000AC 8$:     BLBS    FNCT+24, 9$                               ; 0374
                           50           66  D0 000B0         MOVL    TSF, R0                                   ; 0380
                        24 A0        04 AC  D0 000B3         MOVL    JUSTIFY, 36(R0)                           ; 0381
                                     64 A0  D6 000B8         INCL    100(R0)                                   ; 0382
         40  A0  00000000G FF        04 A0  C3 000BB         SUBL3   4(R0), @SCA+120, 64(R0)                   ; 0383
                           50           66  D0 000C5 9$:     MOVL    TSF, R0                                   ; 0394
                                     40 A0  D5 000C8         TSTL    64(R0)                                    ;
                                        23  18 000CB         BGEQ    11$                                       ;
                           37           67  E8 000CD         BLBS    FNCT+20, 12$                              ;
```

```
                                      7E  7C 000D0            CLRQ    -(SP)                                  ; 0404
                         00000000G    8F  DD 000D2            PUSHL   #RNFCJL                                ;
                    6B                03  FB 000D8            CALLS   #3, ERM                                ;
                    0B          04    A7  E8 000DB            BLBS    FNCT+24, 10$                           ; 0412
                                      7E  7C 000DF            CLRQ    -(SP)                                  ; 0417
                         00000000G    8F  DD 000E1            PUSHL   #RNFLOC                                ;
                    6B                03  FB 000E7            CALLS   #3, ERM                                ;
                    50                66  D0 000EA  10$:      MOVL    TSF, R0                                ;
                                40    A0  D4 000ED            CLRL    64(R0)                                 ; 0419
                    14                67  E8 000F0  11$:      BLBS    FNCT+20, 12$                           ; 0435
                                0C    A8  D4 000F3            CLRL    FRA+12                                 ; 0439
                    68          10    A8  9E 000F6            MOVAB   FRA+16, FRA                            ;
               04   A8                68  D0 000FA            MOVL    FRA, FRA+4                             ;
        00000000G   EF                00  FB 000FE            CALLS   #0, LOUT                               ; 0445
                                      1F  11 00105            BRB     15$                                    ; 0435
                    50                66  D0 00107  12$:      MOVL    TSF, R0                                ; 0455
   0A          08   A0                01  E1 0010A            BBC     #1, 8(R0), 13$                         ;
                    03   00000000G    EF  E9 0010F            BLBC    OUTOPT+8, 13$                          ;
                                34    A0  D6 00116            INCL    52(R0)                                 ; 0459
                                04    A0  D5 00119  13$:      TSTL    4(R0)                                  ; 0461
                                      03  13 0011C            BEQL    14$                                    ;
                                34    A0  D6 0011E            INCL    52(R0)                                 ; 0463
                                      04  DD 00121  14$:      PUSHL   #4                                     ; 0466
                    6A                01  FB 00123            CALLS   #1, FOOFIL                             ;
                    3A          04    A7  E8 00126  15$:      BLBS    FNCT+24, 16$                           ; 0490
                    50                66  D0 0012A            MOVL    TSF, R0                                ; 0495
                                      60  7C 0012D            CLRQ    (R0)                                   ; 0496
                                18    A0  D4 0012F            CLRL    24(R0)                                 ; 0498
               7C   A0                01  8A 00132            BICB2   #1, 124(R0)                            ; 0500
             0080   C0                01  8A 00136            BICB2   #1, 128(R0)                            ; 0501
                                20    A0  D4 0013B            CLRL    32(R0)                                 ; 0502
                                38    A0  7C 0013E            CLRQ    56(R0)                                 ; 0503
                         00000000G    EF  7C 00141            CLRQ    SCA+336                                ; 0505
                         00000000G    EF  D4 00147            CLRL    SCA+332                                ; 0507
                                60    A0  D4 0014D            CLRL    96(R0)                                 ; 0508
                                28    A0  D4 00150            CLRL    40(R0)                                 ; 0509
                              0088    C0  D4 00153            CLRL    136(R0)                                ; 0510
                                34    A0  D4 00157            CLRL    52(R0)                                 ; 0511
                                08    A0  7C 0015A            CLRQ    8(R0)                                  ; 0499
        00000000G   EF                00  FB 0015D            CALLS   #0, SCL                                ; 0514
                                          04 00164  16$:      RET                                            ; 0521

; Routine Size:  357 bytes,    Routine Base:  $CODE$ + 0000


;  395          0522  1
```

```
1 %sbttl 'OUTNJ -- output text unjustified'
1 GLOBAL ROUTINE outnj : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Forces out the current text, without allowing it to be justified.
1 !
1 ! FORMAL PARAMETERS:    None
1 !
1 ! IMPLICIT INPUTS:      None
1 !
1 ! IMPLICIT OUTPUTS:     None
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:     None
1 !
1 ! SIDE EFFECTS:         None
1 !
1 !--
1
2       BEGIN
2
3       IF ( NOT .SCA_FC)
3           OR ( NOT .SCA_FILL)
2       THEN
2           ENDWRD (FALSE, FALSE, FALSE)
2       ELSE
3           BEGIN
3
4           IF (.SCA_WRD_FOOTW NEQ 0)                   ! Footnotes attached to this word?
3           THEN
3               ! Take care of pending footnote.
3               BEGIN
4               TSF_FOOTW = .TSF_FOOTW + .SCA_WRD_FOOTW;    ! Update count of footnotes associated with this lin
4               SCA_WRD_FOOTW = 0                           ! Make sure footnotes don't get counted twice
3               END;
3
3           ! Is there an index entry associated with this word?
3           IF .SCA_WRD_F_XTN NEQ 0
3           THEN
3               ! Take care of pending index entry.
4               BEGIN
4
4               IF .TSF_FIRST_XTN EQL 0
4               THEN
4                   TSF_FIRST_XTN = .SCA_WRD_F_XTN;
4
4               TSF_LAST_XTN = .SCA_WRD_L_XTN;
3               END;
2           END;
2
2       SCA_WRD_F_XTN = 0;
2       SCA_WRD_L_XTN = 0;
2       OUTLIN (FALSE);                                 ! Don't justify line.
2       TSF_JUST_ALG = 0;                               ! Reset justification algorithm.
2       SCA_CONT = FALSE;                               ! Don't allow concatenation (.NO SPACE)
```

```
;  454          0580  1       END;                                            ! End of OUTNJ



                        000C 00000        .ENTRY  OUTNJ, Save R2,R3         ; 0524
          53 00000000G EF 9E 00002        MOVAB   TSF, R3                   ;
          52 00000000G EF 9E 00009        MOVAB   SCA+292, R2               ;
          05      FF70 C2 E9 00010        BLBC    SCA+148, 1$               ; 0546
          0D      FF44 D2 E8 00015        BLBS    @SCA+104, 2$              ; 0547
                       7E 7C 0001A 1$:    CLRQ    -(SP)                     ; 0549
                       7E D4 0001C        CLRL    -(SP)                     ;
00000000G EF           03 FB 0001E        CALLS   #3, ENDWRD                ;
                       25 11 00025        BRB     5$                        ;
          51           62 D0 00027 2$:    MOVL    SCA+292, R1               ; 0553
                       09 13 0002A        BEQL    3$                        ;
          50           63 D0 0002C        MOVL    TSF, R0                   ; 0556
       0C A0           51 C0 0002F        ADDL2   R1, 12(R0)                ; 0557
                       62 D4 00033        CLRL    SCA+292                   ; 0558
          51        04 A2 D0 00035 3$:    MOVL    SCA+296, R1               ; 0562
                       11 13 00039        BEQL    5$                        ;
          50           63 D0 0003B        MOVL    TSF, R0                   ; 0567
                    38 A0 D5 0003E        TSTL    56(R0)                    ;
                       04 12 00041        BNEQ    4$                        ;
       38 A0           51 D0 00043        MOVL    R1, 56(R0)                ; 0569
       3C A0        08 A2 D0 00047 4$:    MOVL    SCA+300, 60(R0)           ; 0571
                    04 A2 7C 0004C 5$:    CLRQ    SCA+296                   ; 0575
                       7E D4 0004F        CLRL    -(SP)                     ; 0577
     FE45 CF           01 FB 00051        CALLS   #1, OUTLIN                ;
          50           63 D0 00056        MOVL    TSF, R0                   ;
                    64 A0 D4 00059        CLRL    100(R0)                   ; 0578
                    80 A2 D4 0005C        CLRL    SCA+164                   ; 0579
                          04 0005F        RET                               ; 0580

; Routine Size:  96 bytes,    Routine Base:  $CODE$ + 0165


;  455          0581  1
```

```
1 %sbttl 'OUTJ -- output text justified'
1 GLOBAL ROUTINE outj : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Forces out the current text, causing it to be justified.
1 !
1 ! FORMAL PARAMETERS:     None
1 !
1 ! IMPLICIT INPUTS:       None
1 !
1 ! IMPLICIT OUTPUTS:      None
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:      None
1 !
1 ! SIDE EFFECTS:          None
1 !
1 !--
1
2     BEGIN
2
3     IF  ( NOT .SCA_FC)
3         OR ( NOT .SCA_FILL)
2     THEN
2         ENDWRD (FALSE, FALSE, FALSE)
2     ELSE
3         BEGIN
3
4         IF (.SCA_WRD_FOOTW NEQ 0)             ! Footnotes attached to this word?
3         THEN
3             ! Take care of pending footnote.
4             BEGIN
4             TSF_FOOTW = .TSF_FOOTW + .SCA_WRD_FOOTW;    ! Update count of footnotes associated with this lin
4             SCA_WRD_FOOTW = 0                        ! Make sure footnotes are not counted twice
3             END;
3
3         IF .SCA_WRD_F_XTN NEQ 0
3         THEN
3             ! Take care of pending index entry.
4             BEGIN
4
4             IF .TSF_FIRST_XTN EQL 0
4             THEN
4                 TSF_FIRST_XTN = .SCA_WRD_F_XTN;
4
4             TSF_LAST_XTN = .SCA_WRD_L_XTN;
3             END;
2         END;
2
2     SCA_WRD_F_XTN = 0;
2     SCA_WRD_L_XTN = 0;
2     OUTLIN (TRUE);                                   ! Justify line.
2     TSF_JUST_ALG = 0;                                ! Reset justification algorithm.
2     SCA_CONT = FALSE;                                ! Don't allow concatenation (.NO SPACE)
1     END;                                             ! End of OUTJ
```

```
                                  000C 00000           .ENTRY  OUTJ, Save R2,R3                         ; 0583
                53 00000000G   EF   9E 00002           MOVAB   TSF, R3                                  ;
                52 00000000G   EF   9E 00009           MOVAB   SCA+292, R2                              ;
                05       FF70  C2   E9 00010           BLBC    SCA+148, 1$                              ; 0605
                0D       FF44  D2   E8 00015           BLBS    @SCA+104, 2$                             ; 0606
                               7E   7C 0001A  1$:      CLRQ    -(SP)                                    ; 0608
                               7E   D4 0001C           CLRL    -(SP)                                    ;
        00000000G   EF         03   FB 0001E           CALLS   #3, ENDWRD                               ;
                               25   11 00025           BRB     5$                                       ;
                    51         62   D0 00027  2$:      MOVL    SCA+292, R1                              ; 0612
                               09   13 0002A           BEQL    3$                                       ;
                    50         63   D0 0002C           MOVL    TSF, R0                                  ; 0615
              0C    A0         51   C0 0002F           ADDL2   R1, 12(R0)                               ; 0616
                               62   D4 00033           CLRL    SCA+292                                  ; 0617
                    51     04  A2   D0 00035  3$:      MOVL    SCA+296, R1                              ; 0620
                               11   13 00039           BEQL    5$                                       ;
                    50         63   D0 0003B           MOVL    TSF, R0                                  ; 0625
                           38  A0   D5 0003E           TSTL    56(R0)                                   ;
                               04   12 00041           BNEQ    4$                                       ;
              38    A0         51   D0 00043           MOVL    R1, 56(R0)                               ; 0627
              3C    A0     08  A2   D0 00047  4$:      MOVL    SCA+300, 60(R0)                          ; 0629
                           04  A2   7C 0004C  5$:      CLRQ    SCA+296                                  ; 0633
                               01   DD 0004F           PUSHL   #1                                       ; 0635
            FDE5    CF         01   FB 00051           CALLS   #1, OUTLIN                               ;
                    50         63   D0 00056           MOVL    TSF, R0                                  ;
                           64  A0   D4 00059           CLRL    100(R0)                                  ; 0636
                           80  A2   D4 0005C           CLRL    SCA+164                                  ; 0637
                                    04 0005F           RET                                              ; 0638

; Routine Size:  96 bytes,    Routine Base:  $CODE$ + 01C5


;  514          0639  1
```

```
  1 %sbttl 'OUTCRG -- remove carrage control from TSF'
  1 GLOBAL ROUTINE outcrg : NOVALUE =
  1
  1 !++
  1 ! FUNCTIONAL DESCRIPTION:
  1 !
  1 !       Clears out carriage control sequences from the TSF.
  1 !
  1 ! FORMAL PARAMETERS:     None
  1 !
  1 ! IMPLICIT INPUTS:       None
  1 !
  1 ! IMPLICIT OUTPUTS:      None
  1 !
  1 ! ROUTINE VALUE:
  1 ! COMPLETION CODES:      None
  1 !
  1 ! SIDE EFFECTS:          None
  1 !--
  1
  2     BEGIN
  2     ! This is a record full of "normal" text to be output.
  2     FS_INIT (FRA);
  2     TSF_JUSTIFY = FALSE;
  2     TSF_PADDING = 0;
  2
U 2 %IF DSRPLUS %THEN
U 2
U 2     ! If collecting topnotes, go through OUTNJ to make sure blank lines
U 2     ! get put in the right place.
U 2     IF  .TN_COLLECTING                          ! Collecting topnotes?
U 2     THEN
U 2         OUTNJ ()
U 2     ELSE
U 2         BEGIN
  2 %FI
  2
  2     ! If collecting footnotes, go through OUTNJ to make sure blank lines
  2     ! get put in the right place.
  2     IF  .FNCT_COLLECTING                        ! Collecting footnotes?
  2     THEN
  2         OUTNJ ()
  2     ELSE
  2         LOUT ();
  2
U 2 %IF DSRPLUS %THEN
U 2         END;
  2 %FI
  2
  2     ! Throw away leftovers from line just output.
  2     TSF_INT_HL = 0;
  2     TSF_EXT_HL = 0;
  2     TSF_INT_VL = 0;
  2     TSF_NBITS = 0;
  2     TSF_BARS = FALSE;                           ! Turn off change bars.
  2     TSF_H_BARS = FALSE;
  2     TSF_JUS_CNT = 0;
```

```
;  573          0697  2       TSF_FIRST_XTN = 0;
;  574          0698  2       TSF_LAST_XTN = 0;
;  575          0699  2       TSF_TEXT = FALSE;                                      ! There's nothing there now.
;  576          0700  2       TSF_ADJUST = 0;
;  577          0701  2       TSF_FOOTW = 0;
;  578          0702  2       FS_INIT (MRA);
;  579          0703  2       SCA_WRD_PNTR = .FS_START (MRA);
;  580          0704  2       TSF_NEXT_REG = 0;
;  581          0705  1       END;                                                   ! End of OUTCRG



                                               000C 00000         .ENTRY  OUTCRG, Save R2,R3                              ; 0641
                       53 00000000G   EF   9E 00002         MOVAB   TSF, R3                                                 ;
                       52 00000000G   EF   9E 00009         MOVAB   FRA, R2                                                 ;
                                0C    A2   D4 00010         CLRL    FRA+12                                                  ; 0662
                       62       10    A2   9E 00013         MOVAB   FRA+16, FRA                                             ;
            04    A2            62         D0 00017         MOVL    FRA, FRA+4                                              ;
                  50            63         D0 0001B         MOVL    TSF, R0                                                 ;
                                24    A0   D4 0001E         CLRL    36(R0)                                                  ; 0663
                                40    A0   D4 00021         CLRL    64(R0)                                                  ; 0664
                       07 00000000G   EF   E9 00024         BLBC    FNCT+20, 1$                                             ; 0679
          FF10    CF                  00   FB 0002B         CALLS   #0, OUTNJ                                               ; 0681
                                      07   11 00030         BRB     2$                                                      ;
     00000000G    EF                  00   FB 00032 1$:     CALLS   #0, LOUT                                                ; 0683
                  50                  63   D0 00039 2$:     MOVL    TSF, R0                                                 ; 0686
                                      60   7C 0003C         CLRQ    (R0)                                                    ; 0690
                                18    A0   D4 0003E         CLRL    24(R0)                                                  ; 0692
            7C    A0                  01   8A 00041         BICB2   #1, 124(R0)                                             ; 0694
          0080    C0                  01   8A 00045         BICB2   #1, 128(R0)                                             ; 0695
                                20    A0   D4 0004A         CLRL    32(R0)                                                  ; 0696
                                38    A0   7C 0004D         CLRQ    56(R0)                                                  ; 0697
                                60    A0   D4 00050         CLRL    96(R0)                                                  ; 0699
                                28    A0   D4 00053         CLRL    40(R0)                                                  ; 0700
                                08    A0   7C 00056         CLRQ    8(R0)                                                   ; 0693
                       51 00000000G   EF   D0 00059         MOVL    MRA, R1                                                 ; 0702
                                0C    A1   D4 00060         CLRL    12(R1)                                                  ;
                       61       10    A1   9E 00063         MOVAB   16(R1), (R1)                                            ;
            04    A1            61         D0 00067         MOVL    (R1), 4(R1)                                             ;
     00000000G    EF            61         D0 0006B         MOVL    (R1), SCA+248                                           ; 0703
                              0088    C0   D4 00072         CLRL    136(R0)                                                 ; 0704
                                           04 00076         RET                                                             ; 0705

; Routine Size:  119 bytes,    Routine Base:  $CODE$ + 0225


;  582          0706  1
```

```
  1 %sbttl 'OUTPAS -- directs pass-though record'
  1 GLOBAL ROUTINE outpas (fullwords, address, xtn_pointer, type) : NOVALUE =
  1
  1 !++
  1 ! FUNCTIONAL DESCRIPTION:
  1 !
  1 !           Directs a passthrough record to the appropriate location.
  1 !
  1 ! FORMAL PARAMETERS:
  1 !
  1 !           fullwords   - Number of fullwords in the vector pointed to by address.
  1 !           address     - Address of the block of data to write.
  1 !           type        - Minor record type, as defined in FOOREC.REQ or TNREC.REQ.
  1 !           xtn_pointer - Offset into the passthrough record to apply in
  1 !                         order to find the transaction number.
  1 !
  1 ! IMPLICIT INPUTS:        None
  1 !
  1 ! IMPLICIT OUTPUTS:       None
  1 !
  1 ! ROUTINE VALUE:
  1 ! COMPLETION CODES:       None
  1 !
  1 ! SIDE EFFECTS:           None
  1 !--
  1
  2     BEGIN
  2
U 2 %IF DSRPLUS %THEN
U 2     !
U 2     ! If topnotes are being collected, then write the record to the topnote
U 2     ! file.  Otherwise, direct the record to the appropriate utility.
U 2     !
U 2     IF   .TN_COLLECTING
U 2     THEN
U 2         !
U 2         ! Set up some fields in the record header and
U 2         ! write the record to the work file.
U 2         !
U 2         BEGIN
U 2         TNREC_MAJOR_TYPE  = TNREC_MAJ_PASS;
U 2         TNREC_MINOR_TYPE  = .TYPE;
U 2         TNREC_XTN_PTR     = .XTN_POINTER;
U 2         TNREC_ADDRESS     = .ADDRESS;
U 2         TNREC_RECORD_SIZE = .FULLWORDS;
U 2         TNFIL (TN_PWRITE)
U 2         END
U 2     ELSE
U 2         BEGIN
  2 %FI
  2
  2     ! If footnotes are being collected, then write the record to the footnote
  2     ! work file. Otherwise, direct the record to the appropriate utility.
  2     IF .FNCT_COLLECTING
  2     THEN
  2         ! We are between .FOOTNOTE and .END FOOTNOTE
  2         ! Write the record to the work file.
```

```
;  641          0764  3         BEGIN
;  642          0765  3         ! First set up some fields in the record header.
;  643          0766  3         FOOREC_MAJOR_TYPE  = FOOREC_MAJ_PASS;
;  644          0767  3         FOOREC_MINOR_TYPE  = .TYPE;
;  645          0768  3         FOOREC_XTN_PTR     = .XTN_POINTER;
;  646          0769  3         FOOREC_ADDRESS     = .ADDRESS;
;  647          0770  3         FOOREC_RECORD_SIZE = .FULLWORDS;
;  648          0771  3         FOOFIL (FOO_PWRIT)
;  649          0772  3         END
;  650          0773  2     ELSE
;  651          0774  2         ! Not between .FOOTNOTE and .END FOOTNOTE. Direct the record to
;  652          0775  2         ! the appropriate .B%% file and write the binary record.
;  653          0776  2         $XPO_PUT (IOB = BRNOOB, BINARY_DATA = (.FULLWORDS, .ADDRESS));
;  654          0777  2
;  655        U 0778  2 %IF DSRPLUS %THEN
;  656        U 0779  2         END;
;  657          0780  2 %FI
;  658          0781  2
;  659          0782  1     END;                                        ! End of OUTPAS


                                                                          .EXTRN  XPO$PUT, XPO$FAILURE

                                                       000C 00000         .ENTRY  OUTPAS, Save R2,R3                              ; 0708
                                       53 00000000G  EF  9E 00002         MOVAB   IOB$+68, R3                                     ;
                                       52 00000000G  EF  9E 00009         MOVAB   FOOREC, R2                                      ;
                                       5E            08  C2 00010         SUBL2   #8, SP                                          ;
                                       21 00000000G  EF  E9 00013         BLBC    FNCT+20, 1$                                     ; 0760
                                       62            02  D0 0001A         MOVL    #2, FOOREC                                      ; 0766
                                   04  A2       10   AC  D0 0001D         MOVL    TYPE, FOOREC+4                                  ; 0767
                                   0C  A2       0C   AC  D0 00022         MOVL    XTN_POINTER, FOOREC+12                          ; 0768
                                   10  A2       08   AC  D0 00027         MOVL    ADDRESS, FOOREC+16                              ; 0769
                                   08  A2       04   AC  D0 0002C         MOVL    FULLWORDS, FOOREC+8                             ; 0770
                                                     0A  DD 00031         PUSHL   #10                                             ; 0771
                            00000000G  EF            01  FB 00033         CALLS   #1, FOOFIL                                      ;
                                                         04 0003A         RET                                                     ;
                         6E        04  AC            04  A5 0003B 1$:     MULW3   #4, FULLWORDS, $IOB$OUTPUT                      ; 0776
                                   02  AE            02  90 00040         MOVB    #2, $IOB$OUTPUT+2                               ;
                                   03  AE            01  90 00044         MOVB    #1, $IOB$OUTPUT+3                               ;
                                   04  AE       08   AC  D0 00048         MOVL    ADDRESS, $IOB$OUTPUT+4                          ;
                                       63            6E  9E 0004D         MOVAB   $IOB$OUTPUT, IOB$+68                            ;
                                   E8  A3            07  90 00050         MOVB    #7, IOB$+44                                     ;
                                          00000000G  EF  9F 00054         PUSHAB  XPO$FAILURE                                     ;
                                                     7E  D4 0005A         CLRL    -(SP)                                           ;
                                                BC   A3  9F 0005C         PUSHAB  IOB$                                            ;
                            00000000G  EF            03  FB 0005F         CALLS   #3, XPO$PUT                                     ;
                                                         04 00066         RET                                                     ; 0782

; Routine Size:  103 bytes,    Routine Base:  $CODE$ + 029C


;  660          0783  1
;  661          0784  1 END                                             ! End of module
;  662          0785  0 ELUDOM
;                                    PSECT SUMMARY

;       Name                    Bytes                          Attributes ;

;    $CODE$                          771  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        99        16            252          00:00.1
;    _$255$DUA28:[RUNOFF.SRC]DSRLIB.L32;1     1248        73         5             86          00:00.3




;                                    COMMAND QUALIFIERS

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

; Size:          771 code + 0 data bytes
; Run Time:         00:18.6
; Elapsed Time:     00:39.1
; Lines/CPU Min:      2529
; Lexemes/CPU-Min: 23474
; Memory Used:  156 pages
; Compilation Complete
```

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY