```
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
```

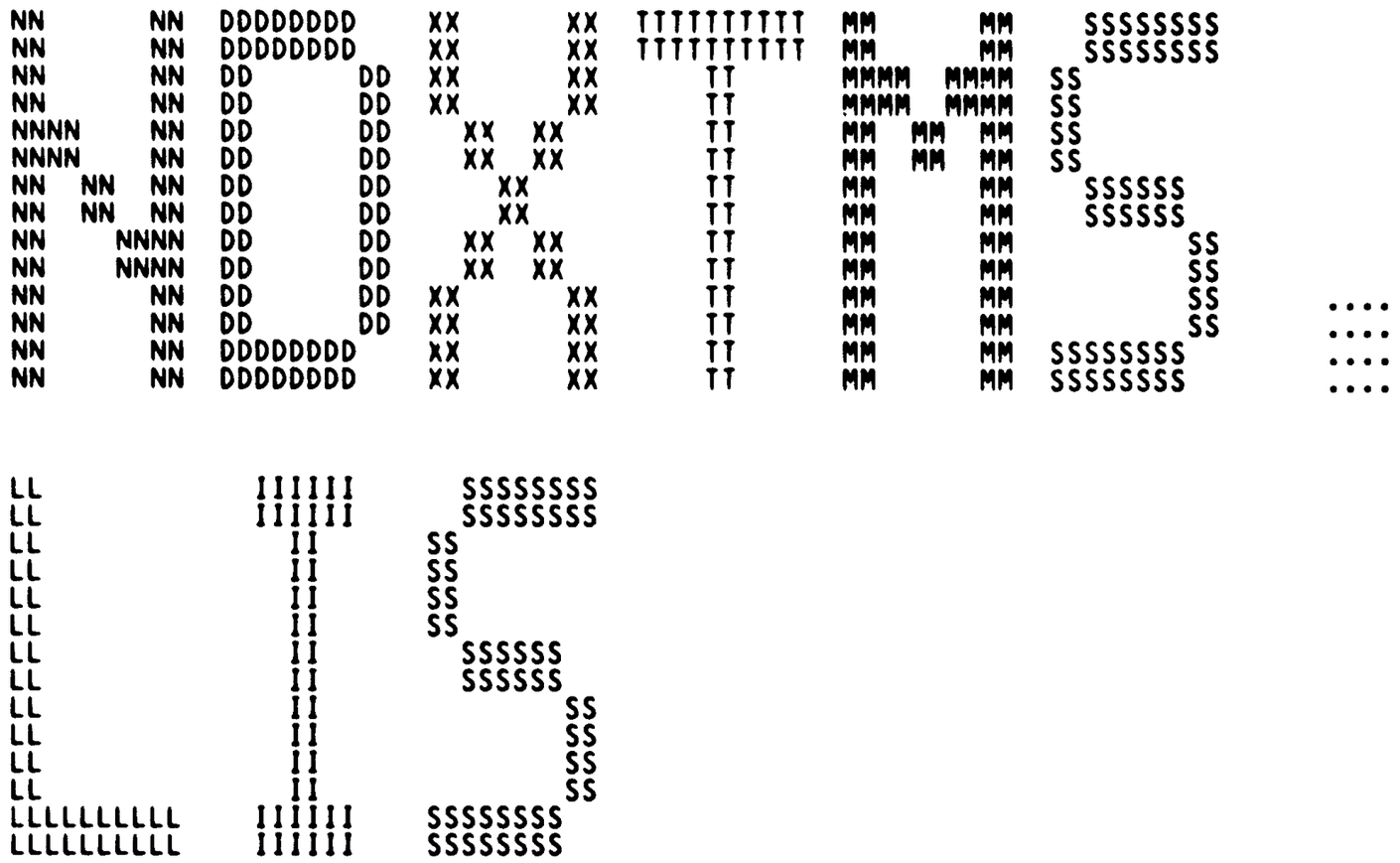
```
_$2

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**NDXTMS



```
;    1          0001  0 %TITLE 'NDXTMS -- RUNOFF to TMS-11 conversion'
;    2          0002  0 MODULE NDXTMS (IDENT = 'V04-000'
;    3          0003  0                %BLISS32 [, ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE, NONEXTERNAL = LONG_RELATIVE)]
;    4          0004  0                ) =
;    5          0005  1 BEGIN
;    6          0006  1
;    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:
;   34          0034  1 !   DSR (Digital Standard RUNOFF) /DSRPLUS DSRINDEX/INDEX Utility
;   35          0035  1 !
;   36          0036  1 ! ABSTRACT:
;   37          0037  1 !   This module contains code to convert RUNOFF input text lines to
;   38          0038  1 !   TMS-11 text lines.
;   39          0039  1 !
;   40          0040  1 ! ENVIRONMENT:   Transportable
;   41          0041  1 !
;   42          0042  1 ! AUTHOR:        JPK
;   43          0043  1 !
;   44          0044  1 ! CREATION DATE: February 1982
;   45          0045  1 !
;   46          0046  1 ! MODIFIED BY:
;   47          0047  1 !
;   48          0048  1 !       003     JPK00015        04-Feb-1983
;   49          0049  1 !               Cleaned up module names, modified revision history to
;   50          0050  1 !               conform with established standards. Updated copyright dates.
;   51          0051  1 !
;   52          0052  1 !       002     JPK00012        24-Jan-1983
;   53          0053  1 !               Modified NDXVMSMSG.MSG to define error messages for both
;   54          0054  1 !               DSRINDEX and INDEX.
;   55          0055  1 !               Added require of NDXVMSREQ.R32 to NDXOUT, NDXFMT, NDXDAT,
;   56          0056  1 !               INDEX, NDXMSG, NDXXTN, NDXTMS, NDXVMS and NDXPAG for BLISS32.
;   57          0057  1 !               Since this file defines the error message literals,
```

```
;   58          0058  1 !               the EXTERNAL REFERENCEs for the error message literals
;   59          0059  1 !               have been removed.
;   60          0060  1 !
;   61          0061  1 !--
```

```
;    63          0062  1 !
;    64          0063  1 ! TABLE OF CONTENTS:
;    65          0064  1 !
;    66          0065  1
;    67          0066  1 FORWARD ROUTINE
;    68          0067  1     RNOTMS : NOVALUE,                   ! Generate TMS-11 output
;    69          0068  1     TMSPUT : NOVALUE;                   ! Put a line to output file
;    70          0069  1
;    71          0070  1 !
;    72          0071  1 ! INCLUDE FILES:
;    73          0072  1 !
;    74          0073  1
;    75          0074  1 LIBRARY 'NXPORT:XPORT';
;    76          0075  1
;    77        L 0076  1 %IF %BLISS (BLISS32)
;    78          0077  1 %THEN
;    79          0078  1
;    80          0079  1 REQUIRE 'REQ:NDXVMSREQ';
;    81          0360  1
;    82          0361  1 %FI
;    83          0362  1
;    84          0363  1 !
;    85          0364  1 ! MACROS:
;    86          0365  1 !
;    87          0366  1 !
;    88          0367  1 ! EQUATED SYMBOLS:
;    89          0368  1 !
;    90          0369  1
;    91          0370  1 GLOBAL LITERAL
;    92          0371  1     TMSSTD = 20,                        ! Relative size of average character for TMS
;    93          0372  1     MSPACE = 32;                        ! TMS 'em' space size
;    94          0373  1
;    95          0374  1 LITERAL
;    96          0375  1     TRUE = 1,
;    97          0376  1     FALSE = 0,
;    98          0377  1     BUFFER_SIZE = 2000;
;    99          0378  1
;   100          0379  1 !
;   101          0380  1 ! OWN STORAGE:
;   102          0381  1 !
;   103          0382  1
;   104          0383  1 GLOBAL                                  ! Top of file string
;   105          0384  1     TMSTOF : $STR_DESCRIPTOR (CLASS = DYNAMIC, STRING = (0, 0));
;   106          0385  1
;   107          0386  1 GLOBAL
;   108          0387  1     TMSSIZ : INITIAL (40);              ! TMS file size in blocks
;   109          0388  1
;   110          0389  1 GLOBAL
;   111          0390  1     CHRSIZ : REF VECTOR [256];
;   112          0391  1
;   113          0392  1 GLOBAL
;   114          0393  1     CHRSZA : VECTOR [256] INITIAL (             ! TMS11 character size vector for "A" format
;   115          0394  1         REP 32 OF (0),                          ! NUL - US
;   116          0395  1         13, 14, 20, 18, 18, 30, 28, 10, 14, 14, ! space ! " # $ % & ' ( )
;   117          0396  1         18, 24, 11, 18, 11, 12,                 ! * + , - . /
;   118          0397  1         REP 10 OF (18),                         ! 0 1 2 3 4 5 6 7 8 9
;   119          0398  1         12, 12, 24, 24, 24, 18, 30,             ! : ; < = > ? @
```

```
;  120     0399  1          26, 26, 26, 28, 26, 23, 28, 30, 13, 19,  ! A B C D E F G H I J
;  121     0400  1          30, 24, 36, 30, 28, 24, 28, 26, 23, 26,  ! K L M N O P Q R S T
;  122     0401  1          28, 26, 36, 30, 26, 24,                  ! U V W X Y Z
;  123     0402  1          10, 18, 10, 19, 26, 19,                  ! [ \ ] ^ _ `
;  124     0403  1          19, 21, 17, 21, 17, 12, 18, 21, 11, 11,  ! a b c d e f g h i j
;  125     0404  1          20, 11, 32, 21, 18, 21, 20, 14, 15, 14,  ! k l m n o p q r s t
;  126     0405  1          21, 19, 26, 19, 19, 17,                  ! u v w x y z
;  127     0406  1          11,  9, 11, 19,  0,                      ! { | } ~ DEL
;  128     0407  1          REP 128 OF (0));                         ! Rest of control characters
;  129     0408  1
;  130     0409  1 GLOBAL
;  131     0410  1      CHRSZE : VECTOR [256] INITIAL (              ! TMS11 character size vector for "E" format
;  132     0411  1          REP 32 OF (0),                           ! NUL - US
;  133     0412  1          13, 12, 12, 28, 20, 32, 26, 12, 10, 10,  ! space ! " # $ % & ' ( )
;  134     0413  1          16, 32, 12, 20, 12, 10,                  ! * + , - . /
;  135     0414  1          REP 10 OF (20),                          ! 0 1 2 3 4 5 6 7 8 9
;  136     0415  1          12, 12, 30, 36, 30, 14, 36,              ! : ; < = > ? @
;  137     0416  1          24, 22, 24, 28, 18, 18, 28, 28, 10, 10,  ! A B C D E F G H I J
;  138     0417  1          22, 18, 32, 28, 30, 20, 30, 22, 18, 20,  ! K L M N O P Q R S T
;  139     0418  1          28, 24, 36, 22, 22, 22,                  ! U V W X Y Z
;  140     0419  1          12, 12, 12, 12, 20, 12,                  ! [ \ ] ^ _ `
;  141     0420  1          18, 20, 18, 20, 18, 10, 18, 20, 10, 10,  ! a b c d e f g h i j
;  142     0421  1          18, 10, 30, 20, 20, 20, 20, 12, 14, 10,  ! k l m n o p q r s t
;  143     0422  1          20, 18, 28, 18, 18, 18,                  ! u v w x y z
;  144     0423  1          12, 8, 12, 12, 0,                        ! { | } ~ DEL
;  145     0424  1          REP 128 OF (0));                         ! Rest of control characters
;  146     0425  1
;  147     0426  1 OWN                                               ! Holds old file name when creating a new file
;  148     0427  1      OLD_FILE : $STR_DESCRIPTOR (CLASS = DYNAMIC, STRING = (0, 0));
;  149     0428  1
;  150     0429  1 OWN
;  151     0430  1      TMSCHR  : INITIAL (0),                       ! Number of characters written to output file
;  152     0431  1      FILE_NO : INITIAL (0);                       ! Continuation file number
;  153     0432  1
;  154     0433  1 !
;  155     0434  1 ! EXTERNAL REFERENCES:
;  156     0435  1 !
;  157     0436  1 EXTERNAL LITERAL
;  158     0437  1      TAB : UNSIGNED (8);
;  159     0438  1
;  160   L 0439  1 %IF %BLISS (BLISS32)
;  161     0440  1 %THEN
;  162     0441  1
;  163     0442  1 EXTERNAL ROUTINE                                  ! File open error handling routine for BLISS32
;  164     0443  1      OPEN_ERROR;
;  165     0444  1
;  166     0445  1 %FI
```

```
;  168          0446  1 %SBTTL 'GLOBAL ROUTINE RNOTMS - Generate TMS-11 Output from RUNOFF text'
;  169          0447  1
;  170          0448  1 GLOBAL ROUTINE RNOTMS (RNO_LEN, RNO_PTR, DSC) : NOVALUE =
;  171          0449  1
;  172          0450  1 !++
;  173          0451  1 !
;  174          0452  1 ! FUNCTIONAL DESCRIPTION:
;  175          0453  1 !
;  176          0454  1 !       This routine converts RUNOFF input text to TMS-11 input text
;  177          0455  1 !
;  178          0456  1 ! FORMAL PARAMETERS:
;  179          0457  1 !
;  180          0458  1 !       RNO_LEN - Length of input text
;  181          0459  1 !       RNO_PTR - CH$PTR to input text
;  182          0460  1 !       DSC     - Address of output string descriptor
;  183          0461  1 !
;  184          0462  1 ! IMPLICIT INPUTS:
;  185          0463  1 !
;  186          0464  1 !       None
;  187          0465  1 !
;  188          0466  1 ! IMPLICIT OUTPUTS:
;  189          0467  1 !
;  190          0468  1 !       None
;  191          0469  1 !
;  192          0470  1 ! ROUTINE VALUE:
;  193          0471  1 ! COMPLETION CODES:
;  194          0472  1 !
;  195          0473  1 !       None
;  196          0474  1 !
;  197          0475  1 ! SIDE EFFECTS:
;  198          0476  1 !
;  199          0477  1 !       None
;  200          0478  1 !--
;  201          0479  2     BEGIN
;  202          0480  2     LOCAL
;  203          0481  2         LINE : VECTOR [CH$ALLOCATION (BUFFER_SIZE + 50)],
;  204          0482  2         LEADING_BLANKS,
;  205          0483  2         I_PTR,                          ! Copy of input line pointer
;  206          0484  2         I_LEN,                          ! Copy of input line length
;  207          0485  2         L_PTR,                          ! Pointer to output line
;  208          0486  2         L_LEN,                          ! Length of output line
;  209          0487  2         DOING_BOLD,                     ! Bold sequence in progress
;  210          0488  2         DOING_ITALICS,                  ! Italic sequence in progress
;  211          0489  2         BOLD_CHAR,                      ! Current character is bold
;  212          0490  2         ITALIC_CHAR,                    ! Current character is italic
;  213          0491  2         OPEN_QUOTE,                     ! Next '' is an open quote if TRUE
;  214          0492  2         CH;                             ! Current character
;  215          0493  2
;  216          0494  2     L_LEN = 0;                            No characters in output string
;  217          0495  2     L_PTR = CH$PTR (LINE);              ! Point to beginning of output string
;  218          0496  2     DOING_BOLD = FALSE;                 ! Not doing a bold string
;  219          0497  2     BOLD_CHAR = FALSE;                  ! Current character is not bold
;  220          0498  2     DOING_ITALICS = FALSE;              ! Not italicized string
;  221          0499  2     ITALIC_CHAR = FALSE;                ! Current character is not italic
;  222          0500  2     OPEN_QUOTE = TRUE;                  ! Next '' is an open quote
;  223          0501  2
;  224          0502  2     I_LEN = .RNO_LEN;
```

```
;   225     0503  2      I_PTR = CH$PLUS (.RNO_PTR, .RNO_LEN - 1);
;   226     0504  2
;   227     0505  2      DECR I FROM .RNO_LEN - 1 TO 0 DO
;   228     0506  2          BEGIN
;   229     0507  2          !
;   230     0508  2          ! Remove trailing whitespace
;   231     0509  2          !
;   232     0510  2          CH = CH$RCHAR (.I_PTR);
;   233     0511  2
;   234     0512  2          IF (.CH NEQ %C' ') AND (.CH NEQ TAB) THEN EXITLOOP;
;   235     0513  2
;   236     0514  2          I_LEN = .I;
;   237     0515  2          I_PTR = CH$PLUS (.I_PTR, -1);
;   238     0516  2          END;
;   239     0517  2
;   240     0518  2      I_PTR = .RNO_PTR;
;   241     0519  2
;   242     0520  2      DECR I FROM .I_LEN TO 1 DO
;   243     0521  2          !
;   244     0522  2          ! Convert leading tabs to "/u"'s
;   245     0523  2          !
;   246     0524  2          IF CH$RCHAR_A (I_PTR) EQL TAB
;   247     0525  2          THEN
;   248     0526  3              BEGIN
;   249     0527  3              CH$WCHAR_A (%C'/', L_PTR);
;   250     0528  3              CH$WCHAR_A (%C'u', L_PTR);
;   251     0529  3              L_LEN = .L_LEN + 2;
;   252     0530  3              END
;   253     0531  2          ELSE
;   254     0532  3              BEGIN
;   255     0533  3              I_LEN = .I;
;   256     0534  3              I_PTR = CH$PLUS (.I_PTR, -1);       ! Back up to account for overshoot
;   257     0535  3              EXITLOOP;
;   258     0536  2              END;
;   259     0537  2
;   260     0538  2      LEADING_BLANKS = 0;
;   261     0539  2      DECR I FROM .I_LEN TO 1 DO
;   262     0540  2          !
;   263     0541  2          ! Count the number of leading blanks
;   264     0542  2          !
;   265     0543  2          IF CH$RCHAR_A (I_PTR) EQL %C' '
;   266     0544  2          THEN
;   267     0545  2              LEADING_BLANKS = .LEADING_BLANKS + 1
;   268     0546  2          ELSE
;   269     0547  3              BEGIN
;   270     0548  3              I_LEN = .I;
;   271     0549  3              I_PTR = CH$PLUS (.I_PTR, -1);       ! Back up to account for overshoot
;   272     0550  3              EXITLOOP;
;   273     0551  2              END;
;   274     0552  2
;   275     0553  2      IF .LEADING_BLANKS
;   276     0554  2      THEN
;   277     0555  3          BEGIN
;   278     0556  3          !
;   279     0557  3          ! Odd number of leading blanks
;   280     0558  3          !
;   281     0559  3          I_LEN = .I_LEN + 1;
```

```
;  282          0560  3          I_PTR = CH$PLUS (.I_PTR, -1);
;  283          0561  3
;  284          0562  3          LEADING_BLANKS = .LEADING_BLANKS - 1;
;  285          0563  2          END;
;  286          0564  2
;  287          0565  2      INCR I FROM 1 TO .LEADING_BLANKS / 2 DO
;  288          0566  3          BEGIN
;  289          0567  3          !
;  290          0568  3          ! Insert indent sequence for each pair of leading blanks
;  291          0569  3          !
;  292          0570  3          CH$WCHAR_A (%C'/', L_PTR);
;  293          0571  3          CH$WCHAR_A (%C'm', L_PTR);
;  294          0572  3          L_LEN = .L_LEN + 2;
;  295          0573  2          END;
;  296          0574  2
;  297          0575  2      !
;  298          0576  2      ! Process text portion of line
;  299          0577  2      !
;  300          0578  2      INCR I FROM 1 TO .I_LEN DO
;  301          0579  3          BEGIN
;  302          0580  3          IF .L_LEN GEQ BUFFER_SIZE
;  303          0581  3          THEN
;  304          0582  4              BEGIN
;  305          0583  4              !
;  306          0584  4              ! String too long for buffer
;  307          0585  4              !
;  308        L 0586  4 %IF %BLISS (BLISS32)
;  309          0587  4 %THEN                                          ! Signal errors in BLISS32
;  310          0588  4
;  311          0589  4              SIGNAL (INDEX$_TRUNCATED, 0, INDEX$_TEXTD, 2, .L_LEN, CH$PTR (LINE));
;  312          0590  4
;  313        U 0591  4 %ELSE                                          ! Use $XPO_PUT_MSG otherwise
;  314        U 0592  4
;  315        U 0593  4              $XPO_PUT_MSG (SEVERITY = WARNING,
;  316        U 0594  4                  STRING = 'string too long - truncated',
;  317        U 0595  4                  STRING = $STR_CONCAT ('entry text: ', (.L_LEN, CH$PTR (LINE))));
;  318        U 0596  4
;  319          0597  4 %FI
;  320          0598  4
;  321          0599  4              EXITLOOP;
;  322          0600  3              END;
;  323          0601  3
;  324          0602  3          CH = CH$RCHAR_A (I_PTR);                ! Get next character
;  325          0603  3
;  326          0604  3          SELECTONE .CH OF
;  327          0605  3          SET
;  328          0606  3
;  329          0607  3          [%C'*']:
;  330          0608  3              !
;  331          0609  3              ! Bold next character
;  332          0610  3              !
;  333          0611  3              BOLD_CHAR = TRUE;
;  334          0612  3
;  335          0613  3          [%C'&']:
;  336          0614  3              !
;  337          0615  3              ! Next character is italic
;  338          0616  3              !
```

```
;  339          0617  3              ITALIC_CHAR = TRUE;
;  340          0618  3
;  341          0619  3          [%C'%']:
;  342          0620  4              BEGIN
;  343          0621  4              !
;  344          0622  4              ! Overstrike previous character
;  345          0623  4              !
;  346        L 0624  4  %IF %BLISS (BLISS32)
;  347          0625  4  %THEN                                          ! Signal errors for BLISS32
;  348          0626  4
;  349          0627  4              SIGNAL (INDEX$_OVERSTRK, 0, INDEX$_TEXTD, 2, .RNO_LEN, .RNO_PTR);
;  350          0628  4
;  351        U 0629  4  %ELSE                                          ! Use $XPO_PUT_MSG otherwise
;  352        U 0630  4
;  353        U 0631  4              $XPO_PUT_MSG (SEVERITY = WARNING,
;  354        U 0632  4                  STRING = 'The following line contains an overstrike sequence',
;  355        U 0633  4                  STRING = $STR_CONCAT ('entry text: ', (.RNO_LEN, .RNO_PTR)));
;  356        U 0634  4
;  357          0635  4  %FI
;  358          0636  4
;  359          0637  4              CH$WCHAR_A (%C'[', L_PTR);
;  360          0638  4              CH$WCHAR_A (%C'e', L_PTR);
;  361          0639  4              CH$WCHAR_A (%C';', L_PTR);
;  362          0640  4              CH$WCHAR_A (%C']', L_PTR);
;  363          0641  4              L_LEN = .L_LEN + 4;
;  364          0642  3              END;
;  365          0643  3
;  366          0644  3          [OTHERWISE]:
;  367          0645  4              BEGIN
;  368          0646  4              !
;  369          0647  4              ! A 'normal' character
;  370          0648  4              !
;  371          0649  4              IF .CH EQL %C'_'
;  372          0650  4              THEN
;  373          0651  5                  BEGIN
;  374          0652  5                  !
;  375          0653  5                  ! Character is quoted.
;  376          0654  5                  ! Get character.
;  377          0655  5                  !
;  378          0656  5                  CH = CH$RCHAR_A (I_PTR);
;  379          0657  5                  I = .I + 1;
;  380          0658  4                  END;
;  381          0659  4
;  382          0660  4              IF NOT .BOLD_CHAR
;  383          0661  4              THEN
;  384          0662  5                  BEGIN
;  385          0663  5                  !
;  386          0664  5                  ! Do not bold this character
;  387          0665  5                  !
;  388          0666  5                  IF  .DOING_BOLD
;  389          0667  6                  AND (.CH NEQ %C' ')
;  390          0668  5                  THEN
;  391          0669  6                      BEGIN
;  392          0670  6                      !
;  393          0671  6                      ! Bold is turned on and the current character is non-blank
;  394          0672  6                      !
;  395          0673  6                      CH$WCHAR_A (%C'[', L_PTR);  ! Turn off bold
```

```
;  396          0674  6                    CH$WCHAR_A (%C'f', L_PTR);
;  397          0675  6                    CH$WCHAR_A (%C'r', L_PTR);
;  398          0676  6                    L_LEN = .L_LEN + 3;
;  399          0677  6
;  400          0678  6                    IF .DOING_ITALICS
;  401          0679  6                    THEN
;  402          0680  7                        BEGIN
;  403          0681  7                        !
;  404          0682  7                        ! Must turn italics off too on since both bold
;  405          0683  7                        ! and italics use the same termination sequence
;  406          0684  7                        !
;  407          0685  7                        CH$WCHAR_A (%C'f', L_PTR);
;  408          0686  7                        CH$WCHAR_A (%C'r', L_PTR);
;  409          0687  7                        L_LEN = .L_LEN + 2;
;  410          0688  7
;  411          0689  7                        IF .ITALIC_CHAR
;  412          0690  7                        THEN
;  413          0691  8                            BEGIN
;  414          0692  8                            !
;  415          0693  8                            ! This character is italicized.
;  416          0694  8                            ! Turn italics back on.
;  417          0695  8                            !
;  418          0696  8                            CH$WCHAR_A (%C'f', L_PTR);
;  419          0697  8                            CH$WCHAR_A (%C'i', L_PTR);
;  420          0698  8                            L_LEN = .L_LEN + 2;
;  421          0699  8                            END
;  422          0700  7                        ELSE
;  423          0701  7                            !
;  424          0702  7                            ! Character is not italicized.
;  425          0703  7                            ! Note that we've turned off italics
;  426          0704  7                            !
;  427          0705  7                            DOING_ITALICS = FALSE;
;  428          0706  6                        END;
;  429          0707  6
;  430          0708  6                    CH$WCHAR_A (%C']', L_PTR);
;  431          0709  6                    L_LEN = .L_LEN + 1;
;  432          0710  6
;  433          0711  6                    DOING_BOLD = FALSE;
;  434          0712  5                    END;
;  435          0713  5                END
;  436          0714  4            ELSE
;  437          0715  5                BEGIN
;  438          0716  5                !
;  439          0717  5                ! Bold next character
;  440          0718  5                !
;  441          0719  5                IF NOT .DOING_BOLD
;  442          0720  5                THEN
;  443          0721  6                    BEGIN
;  444          0722  6                    !
;  445          0723  6                    ! Turn on bolding
;  446          0724  6                    !
;  447          0725  6                    CH$WCHAR_A (%C'[', L_PTR);
;  448          0726  6                    CH$WCHAR_A (%C'f', L_PTR);
;  449          0727  6                    CH$WCHAR_A (%C'b', L_PTR);
;  450          0728  6                    CH$WCHAR_A (%C']', L_PTR);
;  451          0729  6                    L_LEN = .L_LEN + 4;
;  452          0730  6
```

```
;  453     0731  6                    DOING_BOLD = TRUE;
;  454     0732  5                    END;
;  455     0733  5
;  456     0734  5                BOLD_CHAR = FALSE;              ! Reset bold character flag
;  457     0735  4                END;
;  458     0736  4
;  459     0737  4            IF NOT .ITALIC_CHAR
;  460     0738  4            THEN
;  461     0739  5                BEGIN
;  462     0740  5                !
;  463     0741  5                ! Do not italicize this character
;  464     0742  5                !
;  465     0743  5                IF  .DOING_ITALICS
;  466     0744  6                AND (.CH NEQ %C' ')
;  467     0745  5                THEN
;  468     0746  6                    BEGIN
;  469     0747  6                    !
;  470     0748  6                    ! Italics are turned on and the current character is non-blank
;  471     0749  6                    !
;  472     0750  6                    CH$WCHAR_A (%C'[', L_PTR);  ! Turn off italics
;  473     0751  6                    CH$WCHAR_A (%C'f', L_PTR);
;  474     0752  6                    CH$WCHAR_A (%C'r', L_PTR);
;  475     0753  6                    L_LEN = .L_LEN + 3;
;  476     0754  6
;  477     0755  6                    IF .DOING_BOLD
;  478     0756  6                    THEN
;  479     0757  7                        BEGIN
;  480     0758  7                        !
;  481     0759  7                        ! Must turn bold off and back on since both
;  482     0760  7                        ! bold and italics use the same termination sequence
;  483     0761  7                        !
;  484     0762  7                        CH$WCHAR_A (%C'f', L_PTR);
;  485     0763  7                        CH$WCHAR_A (%C'r', L_PTR);
;  486     0764  7                        CH$WCHAR_A (%C'f', L_PTR);
;  487     0765  7                        CH$WCHAR_A (%C'b', L_PTR);
;  488     0766  7                        L_LEN = .L_LEN + 4;
;  489     0767  6                        END;
;  490     0768  6
;  491     0769  6                    CH$WCHAR_A (%C']', L_PTR);
;  492     0770  6                    L_LEN = .L_LEN + 1;
;  493     0771  6
;  494     0772  6                    DOING_ITALICS = FALSE;
;  495     0773  5                    END;
;  496     0774  5                END
;  497     0775  4            ELSE
;  498     0776  5                BEGIN
;  499     0777  5                !
;  500     0778  5                ! Italicize next character
;  501     0779  5                !
;  502     0780  5                IF NOT .DOING_ITALICS
;  503     0781  5                THEN
;  504     0782  6                    BEGIN
;  505     0783  6                    !
;  506     0784  6                    ! Turn on italics
;  507     0785  6                    !
;  508     0786  6                    CH$WCHAR_A (%C'[', L_PTR);
;  509     0787  6                    CH$WCHAR_A (%C'f', L_PTR);
```

```
;  510          0788  6                    CH$WCHAR_A (%C'i', L_PTR);
;  511          0789  6                    CH$WCHAR_A (%C']', L_PTR);
;  512          0790  6                    L_LEN = .L_LEN + 4;
;  513          0791  6
;  514          0792  6                    DOING_ITALICS = TRUE;
;  515          0793  5                    END;
;  516          0794  5
;  517          0795  5                ITALIC_CHAR = FALSE;            ! Reset flag
;  518          0796  4                END;
;  519          0797  4
;  520          0798  5            IF (.CH LSS %C' ') OR (.CH GTR %O'176')
;  521          0799  4            THEN
;  522          0800  5                BEGIN
;  523          0801  5                !
;  524          0802  5                ! Character is a control character
;  525          0803  5                !
;  526          0804  5                IF .CH EQL TAB
;  527          0805  5                THEN
;  528          0806  6                    BEGIN
;  529          0807  6                    !
;  530          0808  6                    ! TAB character
;  531          0809  6                    !
;  532          0810  6                    CH$WCHAR_A (%C'/', L_PTR);
;  533          0811  6                    CH$WCHAR_A (%C'u', L_PTR);
;  534          0812  6                    L_LEN = .L_LEN + 2;
;  535          0813  6                    END
;  536          0814  5                ELSE
;  537          0815  6                    BEGIN
;  538          0816  6                    !
;  539          0817  6                    ! Other control characters are illegal
;  540          0818  6                    !
;  541        L 0819  6 %IF %BLISS (BLISS32)
;  542          0820  6 %THEN                                           ! Signal errors in BLISS32
;  543          0821  6
;  544          0822  6                    SIGNAL (INDEX$_CTRLCHAR, 0, INDEX$_TEXTD, 2, .RNO_LEN, .RNO_PTR);
;  545          0823  6
;  546        U 0824  6 %ELSE                                           ! Use $XPO_PUT_MSG otherwise
;  547        U 0825  6
;  548        U 0826  6                    $XPO_PUT_MSG (SEVERITY = WARNING,
;  549        U 0827  6                        STRING = 'the following line contains control characters - ignored',
;  550        U 0828  6                        STRING = $STR_CONCAT ('entry text: ', (.RNO_LEN, .RNO_PTR)));
;  551        U 0829  6
;  552          0830  6 %FI
;  553          0831  6
;  554          0832  5                    END;
;  555          0833  5                END
;  556          0834  4            ELSE
;  557          0835  4                SELECTONE .CH OF
;  558          0836  4                SET
;  559          0837  4
;  560          0838  4                [%C' ']:
;  561          0839  5                    BEGIN
;  562          0840  5                    CH$MOVE (5, CH$PTR (UPLIT ('*n10*')), .L_PTR);
;  563          0841  5                    L_PTR = CH$PLUS (.L_PTR, 5);
;  564          0842  5                    L_LEN = .L_LEN + 5;
;  565          0843  4                    END;
;  566          0844  4
```

```
;  567     0845  4                    [%C'-']:
;  568     0846  5                        BEGIN
;  569     0847  5                        CH$WCHAR_A (%C'+', L_PTR);
;  570     0848  5                        CH$WCHAR_A (%C'n', L_PTR);
;  571     0849  5                        L_LEN = .L_LEN + 2;
;  572     0850  4                        END;
;  573     0851  4
;  574     0852  4                    [%C'*']:
;  575     0853  5                        BEGIN
;  576     0854  5                        CH$WCHAR_A (%C'+', L_PTR);
;  577     0855  5                        CH$WCHAR_A (%C'a', L_PTR);
;  578     0856  5                        L_LEN = .L_LEN + 2;
;  579     0857  4                        END;
;  580     0858  4
;  581     0859  4                    [%C'=']:
;  582     0860  5                        BEGIN
;  583     0861  5                        CH$WCHAR_A (%C'+', L_PTR);
;  584     0862  5                        CH$WCHAR_A (%C'e', L_PTR);
;  585     0863  5                        L_LEN = .L_LEN + 2;
;  586     0864  4                        END;
;  587     0865  4
;  588     0866  4                    [%C'+']:
;  589     0867  5                        BEGIN
;  590     0868  5                        CH$WCHAR_A (%C'+', L_PTR);
;  591     0869  5                        CH$WCHAR_A (%C'p', L_PTR);
;  592     0870  5                        L_LEN = .L_LEN + 2;
;  593     0871  4                        END;
;  594     0872  4
;  595     0873  4                    [%C'\']:
;  596     0874  5                        BEGIN
;  597     0875  5                        CH$WCHAR_A (%C'+', L_PTR);
;  598     0876  5                        CH$WCHAR_A (%C's', L_PTR);
;  599     0877  5                        L_LEN = .L_LEN + 2;
;  600     0878  4                        END;
;  601     0879  4
;  602     0880  4                    [%C'@']:
;  603     0881  5                        BEGIN
;  604     0882  5                        CH$WCHAR_A (%C'+', L_PTR);
;  605     0883  5                        CH$WCHAR_A (%C't', L_PTR);
;  606     0884  5                        L_LEN = .L_LEN + 2;
;  607     0885  4                        END;
;  608     0886  4
;  609     0887  4                    [%C'/']:
;  610     0888  5                        BEGIN
;  611     0889  5                        CH$WCHAR_A (%C'+', L_PTR);
;  612     0890  5                        CH$WCHAR_A (%C'.', L_PTR);
;  613     0891  5                        L_LEN = .L_LEN + 2;
;  614     0892  4                        END;
;  615     0893  4
;  616     0894  4                    [%C'|']:
;  617     0895  5                        BEGIN
;  618     0896  5                        CH$WCHAR_A (%C'+', L_PTR);
;  619     0897  5                        CH$WCHAR_A (%C'v', L_PTR);
;  620     0898  5                        L_LEN = .L_LEN + 2;
;  621     0899  4                        END;
;  622     0900  4
;  623     0901  4                    [%C'{']:
```

```
;  624     0902  5                    BEGIN
;  625     0903  5                    CH$WCHAR_A (%C'+', L_PTR);
;  626     0904  5                    CH$WCHAR_A (%C'w', L_PTR);
;  627     0905  5                    L_LEN = .L_LEN + 2;
;  628     0906  4                    END;
;  629     0907  4
;  630     0908  4                [%C'}']:
;  631     0909  5                    BEGIN
;  632     0910  5                    CH$WCHAR_A (%C'+', L_PTR);
;  633     0911  5                    CH$WCHAR_A (%C'x', L_PTR);
;  634     0912  5                    L_LEN = .L_LEN + 2;
;  635     0913  4                    END;
;  636     0914  4
;  637     0915  4                [%C'<']:
;  638     0916  5                    BEGIN
;  639     0917  5                    CH$WCHAR_A (%C'+', L_PTR);
;  640     0918  5                    CH$WCHAR_A (%C'y', L_PTR);
;  641     0919  5                    L_LEN = .L_LEN + 2;
;  642     0920  4                    END;
;  643     0921  4
;  644     0922  4                [%C'>']:
;  645     0923  5                    BEGIN
;  646     0924  5                    CH$WCHAR_A (%C'+', L_PTR);
;  647     0925  5                    CH$WCHAR_A (%C'z', L_PTR);
;  648     0926  5                    L_LEN = .L_LEN + 2;
;  649     0927  4                    END;
;  650     0928  4
;  651     0929  4                [%C'[']:
;  652     0930  5                    BEGIN
;  653     0931  5                    CH$WCHAR_A (%C'+', L_PTR);
;  654     0932  5                    CH$WCHAR_A (%C'(', L_PTR);
;  655     0933  5                    L_LEN = .L_LEN + 2;
;  656     0934  4                    END;
;  657     0935  4
;  658     0936  4                [%C']']:
;  659     0937  5                    BEGIN
;  660     0938  5                    CH$WCHAR_A (%C'+', L_PTR);
;  661     0939  5                    CH$WCHAR_A (%C')', L_PTR);
;  662     0940  5                    L_LEN = .L_LEN + 2;
;  663     0941  4                    END;
;  664     0942  4
;  665     0943  4                [%C'""']:
;  666     0944  5                    BEGIN
;  667     0945  5                    IF .OPEN_QUOTE
;  668     0946  5                    THEN
;  669     0947  6                        BEGIN
;  670     0948  6                        !
;  671     0949  6                        ! Opening quote of quoted string
;  672     0950  6                        !
;  673     0951  6                        CH$WCHAR_A (%C'""', L_PTR);
;  674     0952  6                        CH$WCHAR_A (%C'""', L_PTR);
;  675     0953  6
;  676     0954  6                        OPEN_QUOTE = FALSE;     ! Next quote is not an open quote
;  677     0955  6                        END
;  678     0956  5                    ELSE
;  679     0957  6                        BEGIN
;  680     0958  6                        !
```

```
;   681     0959  6                         ! Closing quote
;   682     0960  6                         !
;   683     0961  6                         CH$WCHAR_A (%C'''', L_PTR);
;   684     0962  6                         CH$WCHAR_A (%C'''', L_PTR);
;   685     0963  6
;   686     0964  6                         OPEN_QUOTE = TRUE;      ! Next quote is open quote
;   687     0965  5                         END;
;   688     0966  5
;   689     0967  5                     L_LEN = .L_LEN + 2;
;   690     0968  4                     END;
;   691     0969  4
;   692     0970  4                 [OTHERWISE]:
;   693     0971  5                     BEGIN
;   694     0972  5                     !
;   695     0973  5                     ! A real normal character
;   696     0974  5                     !
;   697     0975  5                     CH$WCHAR_A (.CH, L_PTR);
;   698     0976  5                     L_LEN = .L_LEN + 1;
;   699     0977  4                     END;
;   700     0978  4
;   701     0979  4                 TES;
;   702     0980  4
;   703     0981  3             END;
;   704     0982  3
;   705     0983  3         TES;
;   706     0984  3
;   707     0985  2         END;
;   708     0986  2
;   709     0987  2     IF .DOING_BOLD OR .DOING_ITALICS
;   710     0988  2     THEN
;   711     0989  3         BEGIN
;   712     0990  3         !
;   713     0991  3         ! Turn off bold and italics
;   714     0992  3         !
;   715     0993  3         CH$WCHAR_A (%C'[', L_PTR);
;   716     0994  3         L_LEN = .L_LEN + 1;
;   717     0995  3
;   718     0996  3         IF .DOING_BOLD
;   719     0997  3         THEN
;   720     0998  4             BEGIN
;   721     0999  4             CH$WCHAR_A (%C'f', L_PTR);
;   722     1000  4             CH$WCHAR_A (%C'r', L_PTR);
;   723     1001  4             L_LEN = .L_LEN + 2;
;   724     1002  3             END;
;   725     1003  3
;   726     1004  3         IF .DOING_ITALICS
;   727     1005  3         THEN
;   728     1006  4             BEGIN
;   729     1007  4             CH$WCHAR_A (%C'f', L_PTR);
;   730     1008  4             CH$WCHAR_A (%C'r', L_PTR);
;   731     1009  4             L_LEN = .L_LEN + 2;
;   732     1010  3             END;
;   733     1011  3
;   734     1012  3         CH$WCHAR_A (%C']', L_PTR);
;   735     1013  3         L_LEN = .L_LEN + 1;
;   736     1014  2         END;
;   737     1015  2
```

```
;  738          1016  2      IF NOT .OPEN_QUOTE
;  739          1017  2      THEN
;  740          1018  2          !
;  741          1019  2          ! Saw an open quote but no close quote
;  742          1020  2          !
;  743        L 1021  2 %IF %BLISS (BLISS32)
;  744          1022  2 %THEN                                   ! Signal errors for BLISS32
;  745          1023  2
;  746          1024  2          SIGNAL (INDEX$_CLOSEQUOT, 0, INDEX$_TEXTD, 2, .RNO_LEN, .RNO_PTR);
;  747          1025  2
;  748        U 1026  2 %ELSE                                   ! Use $XPO_PUT_MSG otherwise
;  749        U 1027  2
;  750        U 1028  2          $XPO_PUT_MSG (SEVERITY = WARNING,
;  751        U 1029  2              STRING = 'missing close quote',
;  752        U 1030  2              STRING = $STR_CONCAT ('entry text: ', (.RNO_LEN, .RNO_PTR)));
;  753        U 1031  2
;  754          1032  2 %FI
;  755          1033  2
;  756          1034  2      $STR_COPY (STRING = (.L_LEN, CH$PTR (LINE)), TARGET = .DSC);
;  757          1035  1      END;


                                                                .TITLE  NDXTMS NDXTMS -- RUNOFF to TMS-11 conversion
                                                                .IDENT  \V04-000\

                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                        00  00  00  2A  30  31  6E  2A  00000 P.AAA:  .ASCII  \*n10*\<0><0><0>                                ;

                                                                .PSECT  $OWN$,NOEXE,2

                                                    0000  00000 OLD_FILE:
                                                                .WORD   0                                                   ;
                                                      02  0E  00002         .BYTE   14, 2                                   ;
                                                00000000  00004         .LONG   0                                           ;
                                                00000000  00008 TMSCHR:  .LONG   0                                           ;
                                                00000000  0000C FILE_NO: .LONG   0                                           ;

                                                                .PSECT  $GLOBAL$,NOEXE,2

                                                    0000  00000 TMSTOF::.WORD   0                                           ;
                                                      02  0E  00002         .BYTE   14, 2                                   ;
                                                00000000  00004         .LONG   0                                           ;
                                                00000028  00008 TMSSIZ::.LONG   40                                          ;
                                                          0000C CHRSIZ::.BLKB   4                                           ;
                                                00000000# 00010 CHRSZA::.LONG   0[32]                                       ;
0000001E  00000012  00000012  00000014  0000000E  0000000D  00090         .LONG   13, 14, 20, 18, 18, 30, 28, 10, 14, 14, -   ;
00000018  00000012  0000000E  0000000E  0000000A  0000001C  000A8                 18, 24, 11, 18, 11, 12                      ;
                    0000000C  0000000B  00000012  0000000B  000C0                                                           ;
                                                  00000012# 000D0         .LONG   18[10]                                    ;
00000012  00000018  00000018  00000018  0000000C  0000000C  000F8         .LONG   12, 12, 24, 24, 24, 18, 30, 26, 26, 26, -   ;
0000001A  0000001C  0000001A  0000001A  0000001A  0000001E  00110                 28, 26, 23, 28, 30, 13, 19, 30, 24, 36, -   ;
0000001E  00000013  0000000D  0000001E  0000001C  0000001C  00128                 30, 28, 24, 28, 26, 23, 26, 28, 26, 36, -   ;
0000001C  00000018  0000001C  0000001E  00000024  00000018  00140                 30, 26, 24, 10, 18, 10, 19, 26, 19, 19, -   ;
00000024  0000001A  0000001C  0000001A  00000017  0000001A  00158                 21, 17, 21, 17, 12, 18, 21, 11, 11, 20, -   ;
0000000A  00000012  0000000A  00000018  0000001A  0000001E  00170                 11, 32, 21, 18, 21, 20, 14, 15, 14, 21, -   ;
00000011  00000015  00000013  00000013  0000001A  00000013  00188                 19, 26, 19, 19, 17, 11, 9, 11, 19, 0        ;
```

```
0000000B  00000015  00000012  0000000C  00000011  00000015  001A0
00000012  00000015  00000020  0000000B  00000014  0000000B  001B8
00000015  0000000E  0000000F  0000000E  00000014  00000015  001D0
0000000B  00000011  00000013  00000013  0000001A  00000013  001E8
                    00000000  00000013  0000000B  00000009  00200
                                                  00000000# 00210          .LONG   0[128]
                                                  00000000# 00410 CHRSZE::.LONG   0[32]
00000020  00000014  0000001C  0000000C  0000000C  0000000D  00490          .LONG   13, 12, 12, 28, 20, 32, 26, 12, 10, 10, -
00000020  00000010  0000000A  0000000A  0000000C  0000001A  004A8                  16, 32, 12, 20, 12, 10
                    0000000A  0000000C  00000014  0000000C  004C0
                                                  000J0014# 004D0          .LONG   20[10]
0000000E  0000001E  00000024  0000001E  0000000C  0000000C  004F8          .LONG   12, 12, 30, 36, 30, 14, 36, 24, 22, 24, -
00000012  0000001C  00000018  00000016  00000018  00000024  00510                  28, 18, 18, 28, 28, 10, 10, 22, 18, 32, -
00000016  0000000A  0000000A  0000001C  0000001C  00000012  00528                  28, 30, 20, 30, 22, 18, 20, 28, 24, 36, -
0000001E  00000014  0000001E  0000001C  00000020  00000012  00540                  22, 22, 22, 12, 12, 12, 12, 20, 12, 18, -
00000024  00000018  0000001C  00000014  00000012  00000016  00558                  20, 18, 20, 18, 10, 18, 20, 10, 10, 18, -
0000000C  0000000C  0000000C  00000016  00000016  00000016  00570                  10, 30, 20, 20, 20, 20, 12, 14, 10, 20, -
00000012  00000014  00000012  0000000C  00000014  000C000C  00588                  18, 28, 18, 18, 18, 12, 8, 12, 12, 0
0000000A  00000014  00000012  0000000A  00000012  00000014  005A0
00000014  00000014  0000001E  0000000A  00000012  0000000A  005B8
00000014  0000000A  0000000E  0000000C  00000014  00000014  005D0
0000000C  00000012  00000012  00000012  0000001C  00000012  005E8
                    00000000  0000000C  0000000C  00000008  00600
                                                  00000000# 00610          .LONG   0[128]

                                                  TMSSTD==            20
                                                  MSPACE==            32
                                                          .EXTRN  DSRINDEX$_BADLOGIC
                                                          .EXTRN  DSRINDEX$_BADVALUE
                                                          .EXTRN  DSRINDEX$_INSVIRMEM
                                                          .EXTRN  DSRINDEX$_LINELENG
                                                          .EXTRN  DSRINDEX$_NOREF
                                                          .EXTRN  DSRINDEX$_OPENIN
                                                          .EXTRN  DSRINDEX$_OPENOUT
                                                          .EXTRN  DSRINDEX$_TOOMANY
                                                          .EXTRN  DSRINDEX$_VALERR
                                                          .EXTRN  DSRINDEX$_CANTBAL
                                                          .EXTRN  DSRINDEX$_CLOSEQUOT
                                                          .EXTRN  DSRINDEX$_CONFQUAL
                                                          .EXTRN  DSRINDEX$_CTRLCHAR
                                                          .EXTRN  DSRINDEX$_DOESNTFIT
                                                          .EXTRN  DSRINDEX$_DUPBEGIN
                                                          .EXTRN  DSRINDEX$_EMPTYIN
                                                          .EXTRN  DSRINDEX$_IGNORED
                                                          .EXTRN  DSRINDEX$_INVINPUT
                                                          .EXTRN  DSRINDEX$_INVRECORD
                                                          .EXTRN  DSRINDEX$_LASTCONT
                                                          .EXTRN  DSRINDEX$_NOBEGIN
                                                          .EXTRN  DSRINDEX$_NOEND
                                                          .EXTRN  DSRINDEX$_NOINDEX
                                                          .EXTRN  DSRINDEX$_NOLIST
                                                          .EXTRN  DSRINDEX$_OVERSTRK
                                                          .EXTRN  DSRINDEX$_SKIPPED
                                                          .EXTRN  DSRINDEX$_SYNTAX
                                                          .EXTRN  DSRINDEX$_TEXFILE
                                                          .EXTRN  DSRINDEX$_TOODEEP
                                                          .EXTRN  DSRINDEX$_TOOFEW
```

```
                                                  .EXTRN   DSRINDEX$_TRUNCATED
                                                  .EXTRN   DSRINDEX$_COMPLETE
                                                  .EXTRN   DSRINDEX$_CREATED
                                                  .EXTRN   DSRINDEX$_IDENT
                                                  .EXTRN   DSRINDEX$_PROCFILE
                                                  .EXTRN   DSRINDEX$_TEXT, DSRINDEX$_TEXTD
                                                  .EXTRN   DSRINDEX$_TMS11
                                                  .EXTRN   TAB, OPEN_ERROR
                                                  .EXTRN   XST$COPY, STR$FAILURE

                                                  .PSECT   $CODE$,NOWRT,2

                          0FFC 00000              .ENTRY   RNOTMS, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-   ; 0448
                                                           R11                                          ;
              5E     F7F0   CE   9E 00002         MOVAB    -2064(SP), SP                                ;
                            58   D4 00007         CLRL     L_LEN                                        ; 0494
              57       0C   AE   9E 00009         MOVAB    LINE, L_PTR                                  ; 0495
                            7E   7C 0000D         CLRQ     BOLD_CHAR                                    ; 0497
                            7E   7C 0000F         CLRQ     ITALIC_CHAR                                  ; 0499
                            01   DD 00011         PUSHL    #1                                           ; 0500
              56       04   AC   D0 00013         MOVL     RNO_LEN, I_LEN                               ; 0502
              5A       08   AC   D0 00017         MOVL     RNO_PTR, RTO                                 ; 0503
   50    04   AC            01   C3 0001B         SUBL3    #1, RNO_LEN, RO                              ;
   5B         5A            50   C1 00020         ADDL3    RO, R10, I_PTR                               ;
              50       04   AC   D0 00024         MOVL     RNO_LEN, I                                   ; 0505
                            16   11 00028         BRB      3$                                           ;
              59            6B   9A 0002A 1$:     MOVZBL   (I_PTR), CH                                  ; 0510
              20            59   D1 0002D         CMPL     CH, #32                                      ; 0512
                            09   13 00030         BEQL     2$                                           ;
      00000000G 8F          59   D1 00032         CMPL     CH, #TAB                                     ;
                            08   12 00039         BNEQ     4$                                           ;
              56            50   D0 0003B 2$:     MOVL     I, I_LEN                                     ; 0514
                            5B   D7 0003E         DECL     I_PTR                                        ; 0515
              E7            50   F4 00040 3$:     SOBGEQ   I, 1$                                        ; 0505
              5B            5A   D0 00043 4$:     MOVL     R10, I_PTR                                   ; 0518
              50       01   A6   9E 00046         MOVAB    1(R6), I                                     ; 0520
                            1D   11 0004A         BRB      7$                                           ;
              51            8B   9A 0004C 5$:     MOVZBL   (I_PTR)+, R1                                 ; 0524
      00000000G 8F          51   D1 0004F         CMPL     R1, #TAB                                     ;
                            0A   12 00056         BNEQ     6$                                           ;
              87     752F   8F   B0 00058         MOVW     #29999, (L_PTR)+                             ; 0527
              58            02   C0 0005D         ADDL2    #2, L_LEN                                    ; 0529
                            07   11 00060         BRB      7$                                           ; 0524
              56            50   D0 00062 6$:     MOVL     I, I_LEN                                     ; 0533
                            5B   D7 00065         DECL     I_PTR                                        ; 0534
                            03   11 00067         BRB      8$                                           ; 0532
              E0            50   F5 00069 7$:     SOBGTR   I, 5$                                        ; 0524
                            51   D4 0006C 8$:     CLRL     LEADING_BLANKS                               ; 0538
              50       01   A6   9E 0006E         MOVAB    1(R6), I                                     ; 0539
                            13   11 00072         BRB      11$                                          ;
              52            8B   9A 00074 9$:     MOVZBL   (I_PTR)+, R2                                 ; 0543
              20            52   91 00077         CMPB     R2, #32                                      ;
                            04   12 0007A         BNEQ     10$                                          ;
                            51   D6 0007C         INCL     LEADING_BLANKS                               ; 0545
                            07   11 0007E         BRB      11$                                          ;
              56            50   D0 00080 10$:    MOVL     I, I_LEN                                     ; 0548
                            5B   D7 00083         DECL     I_PTR                                        ; 0549
```

```
                                03 11 00085        BRB     12$                             ; 0547
                EA              50 F5 00087 11$:   SOBGTR  I, 9$                           ; 0543
                06              51 E9 0008A 12$:   BLBC    LEADING_BLANKS, 13$             ; 0553
                                56 D6 0008D        INCL    I_LEN                           ; 0559
                                5B D7 0008F        DECL    I_PTR                           ; 0560
                                51 D7 00091        DECL    LEADING_BLANKS                  ; 0562
                51              02 C6 00093 13$:   DIVL2   #2, R1                          ; 0565
                                50 D4 00096        CLRL    I                               ;
                                08 11 00098        BRB     15$                             ;
                87        6D2F  8F B0 0009A 14$:   MOVW    #27951, (L_PTR)+                ; 0570
                58              02 C0 0009F        ADDL2   #2, L_LEN                       ; 0572
         F4     50              51 F3 000A2 15$:   AOBLEQ  R1, I, 14$                      ; 0565
                            14  AE D4 000A6        CLRL    I                               ; 0578
                                6C 11 000A9        BRB     20$                             ;
   000007D0     8F              58 D1 000AB 16$:   CMPL    L_LEN, #2000                    ; 0580
                                1F 19 000B2        BLSS    17$                             ;
                            20  AE 9F 000B4        PUSHAB  LINE                            ; 0589
                                58 DD 000B7        PUSHL   L_LEN                           ;
                                02 DD 000B9        PUSHL   #2                              ;
                     00000000G  8F DD 000BB        PUSHL   #DSRINDEX$_TEXTD                ;
                                7E D4 000C1        CLRL    -(SP)                           ;
                     00000000G  8F DD 000C3        PUSHL   #DSRINDEX$_TRUNCATED            ;
   00000000G    00              06 FB 000C9        CALLS   #6, LIB$SIGNAL                  ;
                              025C 31 000D0        BRW     62$                             ; 0582
                59              8B 9A 000D3 17$:   MOVZBL  (I_PTR)+, CH                    ; 0602
                2A              59 D1 000D6        CMPL    CH, #42                         ; 0607
                                06 12 000D9        BNEQ    18$                             ;
         0C     AE              01 D0 000DB        MOVL    #1, BOLD_CHAR                   ; 0611
                                36 11 000DF        BRB     20$                             ;
                26              59 D1 000E1 18$:   CMPL    CH, #38                         ; 0613
                                06 12 000E4        BNEQ    19$                             ;
         04     AE              01 D0 000E6        MOVL    #1, ITALIC_CHAR                 ; 0617
                                2B 11 000EA        BRB     20$                             ;
                25              59 D1 000EC 19$:   CMPL    CH, #37                         ; 0619
                                29 12 000EF        BNEQ    21$                             ;
                                5A DD 000F1        PUSHL   R10                             ; 0627
                            04  AC DD 000F3        PUSHL   RNO_LEN                         ;
                                02 DD 000F6        PUSHL   #2                              ;
                     00000000G  8F DD 000F8        PUSHL   #DSRINDEX$_TEXTD                ;
                                7E D4 000FE        CLRL    -(SP)                           ;
                     00000000G  8F DD 00100        PUSHL   #DSRINDEX$_OVERSTRK             ;
   00000000G    00              06 FB 00106        CALLS   #6, LIB$SIGNAL                  ;
                87    5D63655B  8F D0 0010D        MOVL    #1566795099, (L_PTR)+           ; 0637
                58              04 C0 00114        ADDL2   #4, L_LEN                       ; 0641
                              020E 31 00117 20$:   BRW     61$                             ; 0604
   0000005F     8F              59 D1 0011A 21$:   CMPL    CH, #95                         ; 0649
                                06 12 00121        BNEQ    22$                             ;
                59              8B 9A 00123        MOVZBL  (I_PTR)+, CH                    ; 0656
                            14  AE D6 00126        INCL    I                               ; 0657
                3D          0C  AE E8 00129 22$:   BLBS    BOLD_CHAR, 25$                  ; 0660
                4E          10  AE E9 0012D        BLBC    DOING_BOLD, 27$                 ; 0666
                20              59 D1 00131        CMPL    CH, #32                         ; 0667
                                49 13 00134        BEQL    27$                             ;
                87        665B  8F B0 00136        MOVW    #26203, (L_PTR)+                ; 0673
                87          72  8F 90 0013B        MOVB    #114, (L_PTR)+                  ; 0675
                58              03 C0 0013F        ADDL2   #3, L_LEN                       ; 0676
                19          08  AE E9 00142        BLBC    DOING_ITALICS, 24$              ; 0678
```

```
                     87      7266   8F  B0 00146          MOVW    #29286, (L_PTR)+                ; 0685
                     58             02  C0 0014B          ADDL2   #2, L_LEN                       ; 0687
                     0A       04    AE  E9 0014E          BLBC    ITALIC_CHAR, 23$                ; 0689
                     87      6966   8F  B0 00152          MOVW    #26982, (L_PTR)+                ; 0696
                     58             02  C0 00157          ADDL2   #2, L_LEN                       ; 0698
                                    03  11 0015A          BRB     24$                             ; 0689
                               08   AE  D4 0015C 23$:      CLRL    DOING_ITALICS                   ; 0705
                     87        5D   8F  90 0015F 24$:      MOVB    #93, (L_PTR)+                   ; 0708
                                    58  D6 00163          INCL    L_LEN                           ; 0709
                               10   AE  D4 00165          CLRL    DOING_BOLD                      ; 0711
                                    15  11 00168          BRB     27$                             ; 0660
                     0E        10   AE  E8 0016A 25$:      BLBS    DOING_BOLD, 26$                 ; 0719
                     87 5D62665B    8F  D0 0016E          MOVL    #1566729819, (L_PTR)+           ; 0725
                     58             04  C0 00175          ADDL2   #4, L_LEN                       ; 0729
                  10 AE             01  D0 00178          MOVL    #1, DOING_BOLD                  ; 0731
                               0C   AE  D4 0017C 26$:      CLRL    BOLD_CHAR                       ; 0734
                     2E        04   AE  E8 0017F 27$:      BLBS    ITALIC_CHAR, 29$                ; 0737
                     3F        08   AE  E9 00183          BLBC    DOING_ITALICS, 31$              ; 0743
                     20             59  D1 00187          CMPL    CH, #32                         ; 0744
                                    3A  13 0018A          BEQL    31$                             ;
                     87      665B   8F  B0 0018C          MOVW    #26203, (L_PTR)+                ; 0750
                     87        72   8F  90 00191          MOVB    #114, (L_PTR)+                  ; 0752
                     58             03  C0 00195          ADDL2   #3, L_LEN                       ; 0753
                     0A       10    AE  E9 00198          BLBC    DOING_BOLD, 28$                 ; 0755
                     87 62667266    8F  D0 0019C          MOVL    #1650881126, (L_PTR)+           ; 0762
                     58             04  C0 001A3          ADDL2   #4, L_LEN                       ; 0766
                     87        5D   8F  90 001A6 28$:      MOVB    #93, (L_PTR)+                   ; 0769
                                    58  D6 001AA          INCL    L_LEN                           ; 0770
                               08   AE  D4 001AC          CLRL    DOING_ITALICS                   ; 0772
                                    15  11 001AF          BRB     31$                             ; 0737
                     0E        08   AE  E8 001B1 29$:      BLBS    DOING_ITALICS, 30$              ; 0780
                     87 5D69665B    8F  D0 001B5          MOVL    #1567188571, (L_PTR)+           ; 0786
                     58             04  C0 001BC          ADDL2   #4, L_LEN                       ; 0790
                  08 AE             01  D0 001BF          MOVL    #1, DOING_ITALICS               ; 0792
                               04   AE  D4 001C3 30$:      CLRL    ITALIC_CHAR                     ; 0795
                     20             59  D1 001C6 31$:      CMPL    CH, #32                         ; 0798
                                    09  19 001C9          BLSS    32$                             ;
            0000007E 8F             59  D1 001CB          CMPL    CH, #126                        ;
                                    31  15 001D2          BLEQ    34$                             ;
            00000000G 8F            59  D1 001D4 32$:      CMPL    CH, #TAB                        ; 0804
                                    0A  12 001DB          BNEQ    33$                             ;
                     87             2F  90 001DD          MOVB    #47, (L_PTR)+                   ; 0810
                     67        75   8F  90 001E0          MOVB    #117, (L_PTR)                   ; 0811
                                 0084 31  001E4          BRW     41$                             ;
                                    5A  DD 001E7 33$:      PUSHL   R10                             ; 0822
                               04   AC  DD 001E9          PUSHL   RNO_LEN                         ;
                                    02  DD 001EC          PUSHL   #2                              ;
                        00000000G   8F  DD 001EE          PUSHL   #DSRINDEX$_TEXTD                ;
                                    7E  D4 001F4          CLRL    -(SP)                           ;
                        00000000G   8F  DD 001F6          PUSHL   #DSRINDEX$_CTRLCHAR             ;
            00000000G   00              06  FB 001FC          CALLS   #6, LIB$SIGNAL                  ;
                                    17  11 00203          BRB     35$                             ; 0798
            0000005F    8F              59  D1 00205 34$:      CMPL    CH, #95                         ; 0838
                                    11  12 0020C          BNEQ    36$                             ;
         67 00000000'   EF              05  28 0020E          MOVC3   #5, P.AAA, (L_PTR)              ; 0840
                                    57  05  C0 00216          ADDL2   #5, L_PTR                       ; 0841
                                    58  05  C0 00219          ADDL2   #5, L_LEN                       ; 0842
```

```
                                 0109  31 0021C 35$:     BRW     61$                             ; 0835
                           2D      59  D1 0021F 36$:     CMPL    CH, #45                         ; 0845
                                   0A  12 00222          BNEQ    37$                             ;
                           87      2B  90 00224          MOVB    #43, (L_PTR)+                   ; 0847
                           67  6E   8F  90 00227          MOVB    #110, (L_PTR)                   ; 0848
                                 0081  31 0022B          BRW     46$                             ;
                           2A      59  D1 0022E 37$:     CMPL    CH, #42                         ; 0852
                                   0A  12 00231          BNEQ    38$                             ;
                           87      2B  90 00233          MOVB    #43, (L_PTR)+                   ; 0854
                           67  61   8F  90 00236          MOVB    #97, (L_PTR)                    ; 0855
                                 0084  31 0023A          BRW     48$                             ;
                           3D      59  D1 0023D 38$:     CMPL    CH, #61                         ; 0859
                                   0A  12 00240          BNEQ    39$                             ;
                           87      2B  90 00242          MOVB    #43, (L_PTR)+                   ; 0861
                           67  65   8F  90 00245          MOVB    #101, (L_PTR)                   ; 0862
                                 0083  31 00249          BRW     50$                             ;
                           2B      59  D1 0024C 39$:     CMPL    CH, #43                         ; 0866
                                   0A  12 0024F          BNEQ    40$                             ;
                           87      2B  90 00251          MOVB    #43, (L_PTR)+                   ; 0868
                           67  70   8F  90 00254          MOVB    #112, (L_PTR)                   ; 0869
                                 0082  31 00258          BRW     52$                             ;
                0000005C   8F      59  D1 0025B 40$:     CMPL    CH, #92                         ; 0873
                                   0A  12 00262          BNEQ    42$                             ;
                           87      2B  90 00264          MOVB    #43, (L_PTR)+                   ; 0875
                           67  73   8F  90 00267          MOVB    #115, (L_PTR)                   ; 0876
                                 0080  31 0026B 41$:     BRW     54$                             ;
                00000040   8F      59  D1 0026E 42$:     CMPL    CH, #64                         ; 0880
                                   09  12 00275          BNEQ    43$                             ;
                           87      2B  90 00277          MOVB    #43, (L_PTR)+                   ; 0882
                           67  74   8F  90 0027A          MOVB    #116, (L_PTR)                   ; 0883
                                   7F  11 0027E          BRB     56$                             ;
                           2F      59  D1 00280 43$:     CMPL    CH, #47                         ; 0887
                                   08  12 00283          BNEQ    44$                             ;
                           87      2B  90 00285          MOVB    #43, (L_PTR)+                   ; 0889
                           67      2E  90 00288          MOVB    #46, (L_PTR)                    ; 0890
                                   72  11 0028B          BRB     56$                             ;
                0000007C   8F      59  D1 0028D 44$:     CMPL    CH, #124                        ; 0894
                                   09  12 00294          BNEQ    45$                             ;
                           87      2B  90 00296          MOVB    #43, (L_PTR)+                   ; 0896
                           67  76   8F  90 00299          MOVB    #118, (L_PTR)                   ; 0897
                                   7D  11 00297          BRB     59$                             ;
                0000007B   8F      59  D1 0029F 45$:     CMPL    CH, #123                        ; 0901
                                   09  12 002A6          BNEQ    47$                             ;
                           87      2B  90 002A8          MOVB    #43, (L_PTR)+                   ; 0903
                           67  77   8F  90 002AB          MOVB    #119, (L_PTR)                   ; 0904
                                   6B  11 002AF 46$:     BRB     59$                             ;
                0000007D   8F      59  D1 002B1 47$:     CMPL    CH, #125                        ; 0908
                                   09  12 002B8          BNEQ    49$                             ;
                           87      2B  90 002BA          MOVB    #43, (L_PTR)+                   ; 0910
                           67  78   8F  90 002BD          MOVB    #120, (L_PTR)                   ; 0911
                                   59  11 002C1 48$:     BRB     59$                             ;
                           3C      59  D1 002C3 49$:     CMPL    CH, #60                         ; 0915
                                   09  12 002C6          BNEQ    51$                             ;
                           87      2B  90 002C8          MOVB    #43, (L_PTR)+                   ; 0917
                           67  79   8F  90 002CB          MOVB    #121, (L_PTR)                   ; 0918
                                   4B  11 002CF 50$:     BRB     59$                             ;
                           3E      59  D1 002D1 51$:     CMPL    CH, #62                         ; 0922
```

```
                                                09  12 002D4          BNEQ    53$                                     ;
                                          87    2B  90 002D6          MOVB    #43, (L_PTR)+                           ; 0924
                                          67  7A  8F  90 002D9          MOVB    #122, (L_PTR)                           ; 0925
                                                3D  11 002DD 52$:       BRB     59$                                     ;
                                0000005B  8F    59  D1 002DF 53$:       CMPL    CH, #91                                 ; 0929
                                                08  12 002E6          BNEQ    55$                                     ;
                                          87    2B  90 002E8          MOVB    #43, (L_PTR)+                           ; 0931
                                          67    28  90 002EB          MOVB    #40, (L_PTR)                            ; 0932
                                                2C  11 002EE 54$:       BRB     59$                                     ;
                                0000005D  8F    59  D1 002F0 55$:       CMPL    CH, #93                                 ; 0936
                                                08  12 002F7          BNEQ    57$                                     ;
                                          87    2B  90 002F9          MOVB    #43, (L_PTR)+                           ; 0938
                                          67    29  90 002FC          MOVB    #41, (L_PTR)                            ; 0939
                                                1B  11 002FF 56$:       BRB     59$                                     ;
                                          22    59  D1 00301 57$:       CMPL    CH, #34                                 ; 0943
                                                1D  12 00304          BNEQ    60$                                     ;
                                          0A    6E  E9 00306          BLBC    OPEN_QUOTE, 58$                         ; 0945
                                          87    22  90 00309          MOVB    #34, (L_PTR)+                           ; 0951
                                          67    22  90 0030C          MOVB    #34, (L_PTR)                            ; 0952
                                                6E  D4 0030F          CLRL    OPEN_QUOTE                              ; 0954
                                                09  11 00311          BRB     59$                                     ; 0945
                                          87    27  90 00313 58$:       MOVB    #39, (L_PTR)+                           ; 0961
                                          67    27  90 00316          MOVB    #39, (L_PTR)                            ; 0962
                                          6E    01  D0 00319          MOVL    #1, OPEN_QUOTE                          ; 0964
                                                57  D6 0031C 59$:       INCL    L_PTR                                   ; 0952
                                          58    02  C0 0031E          ADDL2   #2, L_LEN                               ; 0967
                                                05  11 00321          BRB     61$                                     ; 0835
                                          87    59  90 00323 60$:       MOVB    CH, (L_PTR)+                            ; 0975
                                                58  D6 00326          INCL    L_LEN                                   ; 0976
                FD7C    14  AE          01    56  F1 00328 61$:       ACBL    I_LEN, #1, I, 16$                       ; 0578
                                          04  10  AE  E8 0032F 62$:       BLBS    DOING_BOLD, 63$                         ; 0987
                                          24  08  AE  E9 00333          BLBC    DOING_ITALICS, 66$                      ;
                                          87  5B  8F  90 00337 63$:       MOVB    #91, (L_PTR)+                           ; 0993
                                                58  D6 0033B          INCL    L_LEN                                   ; 0994
                                          08  10  AE  E9 0033D          BLBC    DOING_BOLD, 64$                         ; 0996
                                          87  7266  8F  B0 00341          MOVW    #29286, (L_PTR)+                        ; 0999
                                          58    02  C0 00346          ADDL2   #2, L_LEN                               ; 1001
                                          08  08  AE  E9 00349 64$:       BLBC    DOING_ITALICS, 65$                      ; 1004
                                          87  7266  8F  B0 0034D          MOVW    #29286, (L_PTR)+                        ; 1007
                                          58    02  C0 00352          ADDL2   #2, L_LEN                               ; 1009
                                          87  5D  8F  90 00355 65$:       MOVB    #93, (L_PTR)+                           ; 1012
                                                58  D6 00359          INCL    L_LEN                                   ; 1013
                                          1C    6E  E8 0035B 66$:       BLBS    OPEN_QUOTE, 67$                         ; 1016
                                                5A  DD 0035E          PUSHL   R10                                     ; 1024
                                              04  AC  DD 00360          PUSHL   RNO_LEN                                 ;
                                                02  DD 00363          PUSHL   #2                                      ;
                                    00000000G   8F  DD 00365          PUSHL   #DSRINDEX$_TEXTD                        ;
                                                7E  D4 0036B          CLRL    -(SP)                                   ;
                                    00000000G   8F  DD 0036D          PUSHL   #DSRINDEX$_CLOSEQUOT                    ;
                00000000G   00                  C6  FB 00373          CALLS   #6, LIB$SIGNAL                          ;
                      18    AE                  58  B0 0037A 67$:       MOVW    L_LEN, $STR$STRING                      ; 1034
                      1A    AE                  0E  90 0037E          MOVB    #14, $STR$STRING+2                      ;
                      1B    AE                  01  90 00382          MOVB    #1, $STR$STRING+3                       ;
                      1C    AE              20  AE  9E 00386          MOVAB   LINE, $STR$STRING+4                     ;
                                    00000000G   EF  9F 0038B          PUSHAB  STR$FAILURE                             ;
                                                7E  D4 00391          CLRL    -(SP)                                   ;
                                              0C  AC  DD 00393          PUSHL   DSC                                     ;
```

```
                                 24    AE  9F 00396         PUSHAB  $STR$STRING                      ;
                                       7E  D4 00399         CLRL    -(SP)                           ;
                     00000000G   EF          05  FB 0039B  CALLS   #5, XST$COPY                    ;
                                             04  003A2  RET                                         ; 1035

; Routine Size:  931 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  759          1036  1 %SBTTL 'GLOBAL ROUTINE TMSPUT -- Write a line to output file'
;  760          1037  1 GLOBAL ROUTINE TMSPUT (LEN, PTR, IOB, IDEAL) : NOVALUE =
;  761          1038  1 !++
;  762          1039  1 !
;  763          1040  1 ! FUNCTIONAL DESCRIPTION:
;  764          1041  1 !
;  765          1042  1 !       This routine is called to write a line to the output file.
;  766          1043  1 !
;  767          1044  1 !       If the output file is TMSSIZ blocks, and IDEAL
;  768          1045  1 !       is TRUE, creates a new output file before writing line.
;  769          1046  1 !
;  770          1047  1 !       If the output file is TMSSIZ + 10 blocks, creates
;  771          1048  1 !       a new output file unconditionally.
;  772          1049  1 !
;  773          1050  1 !       New output file names are of the form
;  774          1051  1 !               {first 3 characters of original file name}{3 digits}.TMS
;  775          1052  1 !
;  776          1053  1 ! FORMAL PARAMETERS:
;  777          1054  1 !
;  778          1055  1 !       LEN     - Length of line
;  779          1056  1 !       PTR     - CH$PTR to line
;  780          1057  1 !       IOB     - Address of IOB
;  781          1058  1 !       IDEAL   - TRUE if file can be broken on this line
;  782          1059  1 !
;  783          1060  1 ! IMPLICIT INPUTS:
;  784          1061  1 !
;  785          1062  1 !       TMSCHR  - Number of characters written to output file
;  786          1063  1 !       TMSSIZ  - Ideal file size in blocks
;  787          1064  1 !
;  788          1065  1 ! IMPLICIT OUTPUTS:
;  789          1066  1 !
;  790          1067  1 !       TMSCHR  - Number of characters written to output file
;  791          1068  1 !
;  792          1069  1 ! ROUTINE VALUE:
;  793          1070  1 ! COMPLETION CODES:
;  794          1071  1 !
;  795          1072  1 !       None
;  796          1073  1 !
;  797          1074  1 ! SIDE EFFECTS:
;  798          1075  1 !
;  799          1076  1 !       None
;  800          1077  1 !--
;  801          1078  2     BEGIN
;  802          1079  2     MAP
;  803          1080  2         IOB : REF $XPO_IOB ();
;  804          1081  2
;  805          1082  3     IF ((.TMSCHR GEQ (.TMSSIZ * 512)) AND .IDEAL)       ! Best place to start a new file
;  806          1083  3     OR  (.TMSCHR GEQ (.TMSSIZ * 512 + 5120))            ! Unconditionally start a new file
;  807          1084  2     THEN
;  808          1085  3         BEGIN
;  809          1086  3         !
;  810          1087  3         ! Must start a new output file.
;  811          1088  3         ! TMS11 can process only 40-50 block files maximum.
;  812          1089  3         !
;  813          1090  3         LOCAL
;  814          1091  3             L_NAME,
;  815          1092  3             SPEC_BLK : $XPO_SPEC_BLOCK;
```

```
;  816     1093  3
;  817     1094  3          IF .OLD_FILE [STR$H_LENGTH] EQL 0
;  818     1095  3          THEN
;  819     1096  3              !
;  820     1097  3              ! Save first output file name.
;  821     1098  3              ! We use it to build the new file name and
;  822     1099  3              ! for a related file specification.
;  823     1100  3              !
;  824     1101  3              $STR_COPY (STRING = IOB [IOB$T_RESULTANT], TARGET = OLD_FILE);
;  825     1102  3
;  826     1103  3          !
;  827     1104  3          ! Parse old file name
;  828     1105  3          !
;  829     1106  3          $XPO_PARSE_SPEC (FILE_SPEC = OLD_FILE, SPEC_BLOCK = SPEC_BLK);
;  830     1107  3
;  831     1108  3          !
;  832     1109  3          ! Use up to 6 characters of old file name for new file name
;  833     1110  3          !
;  834     1111  3          IF .SPEC_BLK [XPO$H_FILE_NAME] GEQ 6
;  835     1112  3          THEN
;  836     1113  3              L_NAME = 6
;  837     1114  3          ELSE
;  838     1115  3              L_NAME = .SPEC_BLK [XPO$H_FILE_NAME];
;  839     1116  3
;  840     1117  3          !
;  841     1118  3          ! Close current file
;  842     1119  3          !
;  843     1120  3          $XPO_CLOSE (IOB = .IOB);
;  844     1121  3
;  845     1122  3          !
;  846     1123  3          ! Open new file.
;  847     1124  3          !
;  848     1125  3          FILE_NO = .FILE_NO + 1;
;  849     1126  3          $XPO_IOB_INIT (IOB = .IOB);
;  850   P 1127  3          $XPO_OPEN (IOB = .IOB, OPTIONS = OUTPUT, DEFAULT = OLD_FILE,
;  851   P 1128  3              FILE_SPEC = $STR_CONCAT ((.L_NAME, .SPEC_BLK [XPO$A_FILE_NAME]),
;  852   P 1129  3                  $STR_ASCII (.FILE_NO, UNSIGNED, LEADING_ZERO, LENGTH = 3))
;  853   P 1130  3              %IF %BLISS (BLISS32) %THEN , FAILURE = OPEN_ERROR %FI
;  854     1131  3              );
;  855     1132  3
;  856   L 1133  3 %IF %BLISS (BLISS32)
;  857     1134  3 %THEN                                                  ! Signal messages in BLISS32
;  858     1135  3
;  859     1136  3          SIGNAL (INDEX$_TMS11, 1, IOB [IOB$T_RESULTANT]);
;  860     1137  3
;  861   U 1138  3 %ELSE                                                  ! Use $XPO_PUT_MSG otherwise
;  862   U 1139  3
;  863   U 1140  3          $XPO_PUT_MSG (SEVERITY = SUCCESS,
;  864   U 1141  3              STRING = $STR_CONCAT ('output file full - continuing with file ''', IOB [IOB$T_RESULTANT], ''''))
;  865   U 1142  3
;  866     1143  3 %FI
;  867     1144  3
;  868     1145  3          !
;  869     1146  3          ! Write initial lines
;  870     1147  3          !
;  871     1148  3          $XPO_PUT (IOB = .IOB, STRING = TMSTOF);
;  872     1149  3          $XPO_PUT (IOB = .IOB, STRING = ' ');
```

```
;  873          1150  3          TMSCHR = .TMSTOF [STR$H_LENGTH] + 1;
;  874          1151  2          END;
;  875          1152  2
;  876          1153  2      !
;  877          1154  2      ! Write line to file
;  878          1155  2      !
;  879          1156  2      TMSCHR = .TMSCHR + .LEN;
;  880          1157  2      $XPO_PUT (IOB = .IOB, STRING = (.LEN, .PTR));
;  881          1158  1      END;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                              20  00008 P.AAB:  .ASCII  \ \                                             ;

                                                        .PSECT  $OWN$,NOEXE,2

                                            0001  00010 $IOB$OUTPUT:
                                                        .WORD   1                                                       ;
                                           01  0E 00012         .BYTE   14, 1                                           ;
                                  00000000' 00014               .ADDRESS P.AAB                                          ;

                                                        $STR$TARGET=        OLD_FILE
                                                        $STR$FILE_SPEC=     OLD_FILE
                                                        $IOB$DEFAULT=       OLD_FILE
                                                        $IOB$OUTPUT=        TMSTOF
                                                        .EXTRN  XPO$PARSE_SPEC, XPO$FAILURE
                                                        .EXTRN  XPO$CLOSE, XST$JOIN
                                                        .EXTRN  XST$ASCII, XPO$OPEN
                                                        .EXTRN  XPO$PUT

                                                        .PSECT  $CODE$,NOWRT,2

                                                    0FFC 00000          .ENTRY  TMSPUT, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-      ; 1037
                                                                                R11
                                          5B 00000000'  EF  9E 00002            MOVAB   $IOB$OUTPUT, R11                    ;
                                          5A 00000000G  EF  9E 00009            MOVAB   XPO$PUT, R10                        ;
                                          59 00000000G  EF  9E 00010            MOVAB   XPO$FAILURE, R9                     ;
                                          58 00000000'  EF  9E 00017            MOVAB   TMSCHR, R8                          ;
                                          5E       B0   AE  9E 0001E            MOVAB   -80(SP), SP                         ;
                                50    08  AB            09  78 00022            ASHL    #9, TMSSIZ, R0                      ; 1082
                                          50            68  D1 00027            CMPL    TMSCHR, R0                          ;
                                                        04  19 0002A            BLSS    1$                                  ;
                                          0D       10   AC  E8 0002C            BLBS    IDEAL, 2$                           ;
                                          50     1400   C0  9E 00030 1$:        MOVAB   5120(R0), R0                        ; 1083
                                          50            68  D1 00035            CMPL    TMSCHR, R0                          ;
                                                        03  18 00038            BGEQ    2$                                  ;
                                                  00F9  31  00 0003A            BRW     6$                                  ;
                                                F8      A8  B5 0003D 2$:        TSTW    OLD_FILE                            ; 1094
                                                        1B  12 00040            BNEQ    3$                                  ;
                                50    0C  AC            1C  C1 00042            ADDL3   #28, IOB, R0                        ; 1101
                                          00000000G     EF  9F 00047            PUSHAB  STR$FAILURE                         ;
                                                        7E  D4 0004D            CLRL    -(SP)                               ;
                                                F8      A8  9F 0004F            PUSHAB  $STR$TARGET                         ;
                                                        50  DD 00052            PUSHL   R0                                  ;
                                                        7E  D4 00054            CLRL    -(SP)                               ;
```

```
                          00000000G  EF                05  FB 00056          CALLS   #5, XST$COPY
                                                       59  DD 0005D 3$:       PUSHL   R9                                    ;  1106
                                                       7E  D4 0005F           CLRL    -(SP)
                                     7E                01  CE 00061          MNEGL   #1, -(SP)
                                           14          AE  9F 00064          PUSHAB  SPEC_BLK
                                           F8          A8  9F 00067          PUSHAB  $STR$FILE_SPEC
                          00000000G  EF                05  FB 0006A          CALLS   #5, XPO$PARSE_SPEC
                                     06          28    AE  B1 00071          CMPW    SPEC_BLK+32, #6                      ;  1111
                                                       05  1F 00075          BLSSU   4$
                                     57                06  D0 00077          MOVL    #6, L_NAME                           ;  1113
                                                       04  11 0007A          BRB     5$
                                     57          28    AE  3C 0007C 4$:       MOVZWL  SPEC_BLK+32, L_NAME                  ;  1115
                                     56          0C    AC  D0 00080 5$:       MOVL    IOB, R6                              ;  1120
                                2C   A6                02  90 00084          MOVB    #2, 44(R6)
                                                       59  DD 00088          PUSHL   R9
                                                       7E  D4 0008A          CLRL    -(SP)
                                                       56  DD 0008C          PUSHL   R6
                          00000000G  EF                03  FB 0008E          CALLS   #3, XPO$CLOSE
                                           04          A8  D6 00095          INCL    FILE_NO                              ;  1125
00F4  8F            00                6E                00  2C 00098          MOVC5   #0, (SP), #0, #244, (R6)             ;  1126
                                                       66      0009F
                                     66 0301003D       8F  D0 000A0          MOVL    #50397245, (R6)
                                1E   A6      020E       8F  B0 000A7          MOVW    #526, 30(R6)
                                                       03  DD 000AD          PUSHL   #3                                   ;  1131
                                           04          A8  DD 000AF          PUSHL   FILE_NO
                                     7E          0603  8F  3C 000B2          MOVZWL  #1539, -(SP)
                          00000000G  EF                03  FB 000B7          CALLS   #3, XST$ASCII
                                     6E                57  B0 000BE          MOVW    L_NAME, $STR$STRING0
                                02   AE                0E  90 000C1          MOVB    #14, $STR$STRING0+2
                                03   AE                01  90 000C5          MOVB    #1, $STR$STRING0+3
                                04   AE          2C    AE  D0 000C9          MOVL    SPEC_BLK+36, $STR$STRING0+4
                                                       50  DD 000CE          PUSHL   R0
                                           04          AE  9F 000D0          PUSHAB  $STR$STRING0
                          00000000G  EF                02  FB 000D3          CALLS   #2, XST$JOIN
                                04   A6                50  D0 000DA          MOVL    R0, 4(R6)
                                08   A6          F8    A8  9E 000DE          MOVAB   $IOB$DEFAULT, 8(R6)
                                2E   A6                02  88 000E3          BISB2   #2, 46(R6)
                                2C   A6                01  90 000E7          MOVB    #1, 44(R6)
                                           00000000G   EF  9F 000EB          PUSHAB  OPEN_ERROR
                                                       7E  D4 000F1          CLRL    -(SP)
                                                       56  DD 000F3          PUSHL   R6
                          00000000G  EF                03  FB 000F5          CALLS   #3, XPO$OPEN
                                           1C          A6  9F 000FC          PUSHAB  28(R6)                               ;  1136
                                                       01  DD 000FF          PUSHL   #1
                                           00000000G   8F  DD 00101          PUSHL   #DSRINDEX$_TMS11
                          00000000G  00                03  FB 00107          CALLS   #3, LIB$SIGNAL
                                44   A6                6B  9E 0010E          MOVAB   $IOB$OUTPUT, 68(R6)                  ;  1148
                                2C   A6                07  90 00112          MOVB    #7, 44(R6)
                                                       59  DD 00116          PUSHL   R9
                                                       7E  D4 00118          CLRL    -(SP)
                                                       56  DD 0011A          PUSHL   R6
                                     6A                03  FB 0011C          CALLS   #3, XPO$PUT
                                44   A6          08    A8  9E 0011F          MOVAB   $IOB$OUTPUT, 68(R6)                  ;  1149
                                2C   A6                07  90 00124          MOVB    #7, 44(R6)
                                                       59  DD 00128          PUSHL   R9
                                                       7E  D4 0012A          CLRL    -(SP)
                                                       56  DD 0012C          PUSHL   R6
```

```
                                 6A              03  FB 0012E           CALLS   #3, XPO$PUT                          ;
                                 68              6B  3C 00131           MOVZWL  TMSTOF, TMSCHR                       ; 1150
                                                 6B  D6 00134           INCL    TMSCHR                               ;
                                 6B        04    AC  C0 00136 6$:       ADDL2   LEN, TMSCHR                          ; 1156
                                 50        0C    AC  D0 0013A           MOVL    IOB, R0                              ; 1157
                           48    AE        04    AC  B0 0013E           MOVW    LEN, $IOB$OUTPUT                     ;
                           4A    AE              0E  90 00143           MOVB    #14, $IOB$OUTPUT+2                   ;
                           4B    AE              01  90 00147           MOVB    #1, $IOB$OUTPUT+3                    ;
                           4C    AE        08    AC  D0 0014B           MOVL    PTR, $IOB$OUTPUT+4                   ;
                           44    A0        48    AE  9E 00150           MOVAB   $IOB$OUTPUT, 68(R0)                  ;
                           2C    A0              07  90 00155           MOVB    #7, 44(R0)                           ;
                                                 59  DD 00159           PUSHL   R9                                   ;
                                                 7E  D4 0015B           CLRL    -(SP)                                ;
                                                 50  DD 0015D           PUSHL   R0                                   ;
                                 6A              03  FB 0015F           CALLS   #3, XPO$PUT                          ;
                                                     04 00162           RET                                          ; 1158

; Routine Size:  355 bytes,    Routine Base:  $CODE$ + 03A3

;  882          1159  1
;  883          1160  ; END                                 ! End of module
;  884          1161  0 ELUDOM



                                                                        .EXTRN  LIB$SIGNAL

;                                  PSECT SUMMARY

;      Name                        Bytes                           Attributes

;  $GLOBAL$                         2064  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                              24  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                              9  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                           1286  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  .  ABS  .                           0  NOVEC,NOWRT,NORD ,NOEXE,NOSHR,  LCL,  ABS,  CON,NOPIC,ALIGN(0)



;                               Library Statistics

;                                        --------- Symbols ---------      Pages       Processing
;      File                              Total    Loaded   Percent      Mapped       Time

;  _$255$DUA28:[SYSLIB]XPORT.L32;1         590       175        29         252       00:00.1



;                                  COMMAND QUALIFIERS
```

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

; Size:          1286 code + 2097 data bytes
; Run Time:        00:47.0
; Elapsed Time:    01:25.4
; Lines/CPU Min:    1482
; Lexemes/CPU-Min: 51731
; Memory Used:  531 pages
; Compilation Complete
```