```
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
```

```
NN       NN  DDDDDDDD    XX       XX  FFFFFFFFFF  MM       MM  TTTTTTTTTT
NN       NN  DDDDDDDD    XX       XX  FFFFFFFFFF  MM       MM  TTTTTTTTTT
NN       NN  DD       DD  XX       XX  FF          MMMM   MMMM      TT
NN       NN  DD       DD  XX       XX  FF          MMMM   MMMM      TT
NNNN     NN  DD       DD    XX   XX    FF          MM  MM  MM      TT
NNNN     NN  DD       DD    XX   XX    FF          MM  MM  MM      TT
NN  NN   NN  DD       DD      XX      FFFFFFFF    MM      MM      TT
NN  NN   NN  DD       DD      XX      FFFFFFFF    MM      MM      TT
NN    NNNN  DD       DD    XX   XX    FF          MM      MM      TT
NN    NNNN  DD       DD    XX   XX    FF          MM      MM      TT
NN      NN  DD       DD  XX       XX  FF          MM      MM      TT          ....
NN      NN  DD       DD  XX       XX  FF          MM      MM      TT          ....
NN      NN  DDDDDDDD    XX       XX  FF          MM      MM      TT          ....
NN      NN  DDDDDDDD    XX       XX  FF          MM      MM      TT          ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;  1     0001  0 %TITLE 'NDXFMT -- Binary index formatter'
;  2     0002  0 MODULE NDXFMT (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 !*                                                                           *
; 11     0011  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
; 12     0012  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
; 13     0013  1 !*  ALL RIGHTS RESERVED.                                                     *
; 14     0014  1 !*                                                                           *
; 15     0015  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 16     0016  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 17     0017  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 18     0018  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
; 19     0019  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
; 20     0020  1 !*  TRANSFERRED.                                                             *
; 21     0021  1 !*                                                                           *
; 22     0022  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
; 23     0023  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
; 24     0024  1 !*  CORPORATION.                                                             *
; 25     0025  1 !*                                                                           *
; 26     0026  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY  OR THE USE  OR  RELIABILITY OF ITS  *
; 27     0027  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
; 28     0028  1 !*                                                                           *
; 29     0029  1 !*                                                                           *
; 30     0030  1 !*****************************************************************************
; 31     0031  1 !
; 32     0032  1
; 33     0033  1 !++
; 34     0034  1 ! FACILITY:
; 35     0035  1 !   DSR (Digital Standard RUNOFF) /DSRPLUS DSRINDEX/INDEX Utility
; 36     0036  1 !
; 37     0037  1 ! ABSTRACT:
; 38     0038  1 !       This module contains the code to generate the output index lines
; 39     0039  1 !       from the internal binary index. The output lines are formatted
; 40     0040  1 !       into pages by module NDXPAG. This module was adopted from the TCX
; 41     0041  1 !       modules XPASSA and XPASSB.
; 42     0042  1 !
; 43     0043  1 ! ENVIRONMENT:   Transportable
; 44     0044  1 !
; 45     0045  1 ! AUTHOR:        JPK
; 46     0046  1 !
; 47     0047  1 ! CREATION DATE: January 1982
; 48     0048  1 !
; 49     0049  1 ! MODIFIED BY:
; 50     0050  1 !
; 51     0051  1 !       013     JPK00022        30-Mar-1983
; 52     0052  1 !               Modified NDXVMS, NDXFMT, NDXPAG, NDXVMSMSG and NDXVMSREQ
; 53     0053  1 !               to generate TEX output. Added module NDXTEX.
; 54     0054  1 !
; 55     0055  1 !       012     JPK00021        28-Mar-1983
; 56     0056  1 !               Modified NDXT20 to include E2.0 functionality.
; 57     0057  1 !               Modified NDXCLIDMP, NDXFMT, NDXPAG, NDXVRS to require RNODEF
```

```
;    58          0058  1 !                for BLISS36 and to remove any conditional require based on
;    59          0059  1 !                DSRPLUS_DEF.
;    60          0060  1 !
;    61          0061  1 !        011     JPK00020        17-Mar-1983
;    62          0062  1 !                Modified NDXFMT to generate a ".NO PAGING" in the prologue
;    63          0063  1 !                for /LAYOUT=GALLEY
;    64          0064  1 !
;    65          0065  1 !        010     JPK00018        09-Mar-1983
;    66          0066  1 !                Modified INDEX to handle new BRN format.
;    67          0067  1 !                Modified NDXOUT to handle specifyable levels on SORT= string.
;    68          0068  1 !                Modified NDXFMT to output new RUNOFF prologue.
;    69          0069  1 !                Modified NDXPAG to output new TMS prologue and RUNOFF epilogue.
;    70          0070  1 !
;    71          0071  1 !        009     JPK00017        23-Feb-1983
;    72          0072  1 !                Modified NDXINI to initialize the zero'th entries of LLINES,
;    73          0073  1 !                RLINES and TLINES which is where the telltale strings are
;    74          0074  1 !                stored by NDXFMT.
;    75          0075  1 !                Modified NDXFMT to write appropriate prologue for /TELLTALE,
;    76          0076  1 !                save the appropriate lines for left and right telltales, and
;    77          0077  1 !                to mark the end of every entry with a NULL.
;    78          0078  1 !                Modified NDXPAG to change the NULL following each entry to a
;    79          0079  1 !                space if LAYOUT is SEPARATE or to a comma otherwise and to
;    80          0080  1 !                generate and output telltales.
;    81          0081  1 !
;    82          0082  1 !        008     JPK00015        04-Feb-1983
;    83          0083  1 !                Cleaned up module names, modified revision history to
;    84          0084  1 !                conform with established standards. Updated copyright dates.
;    85          0085  1 !
;    86          0086  1 !        007     JPK00012        24-Jan-1983
;    87          0087  1 !                Modified NDXVMSMSG.MSG to define error messages for both
;    88          0088  1 !                DSRINDEX and INDEX.
;    89          0089  1 !                Added require of NDXVMSREQ.R32 to NDXOUT, NDXFMT, NDXDAT,
;    90          0090  1 !                INDEX, NDXMSG, NDXXTN, NDXTMS, NDXVMS and NDXPAG for BLISS32.
;    91          0091  1 !                Since this file defines the error message literals,
;    92          0092  1 !                the EXTERNAL REFERENCES for the error message literals
;    93          0093  1 !                have been removed.
;    94          0094  1 !
;    95          0095  1 !        006     JPK00011        24-Jan-1983
;    96          0096  1 !                Changed CMDBLK [NDX$G_LEVEL] to CMDBLK [NDX$H_LEVEL]
;    97          0097  1 !                Changed CMDBLK [NDX$H_FORMAT] to CMDBLK [NDX$R_LAYOUT]
;    98          0098  1 !                Changed CMDBLK [NDX$V_TMS11] and CMDBLK [NDX$V_TEX] to CMDBLK [NDX$H_FORMAT]
;    99          0099  1 !                Changed comparisons of (.CHRSIZ EQLA CHRSZA) to
;   100          0100  1 !                (.CMDBLK [NDX$H_FORMAT] EQL TMS11_A).
;   101          0101  1 !                Definitions were changed in NDXCLI and references to the
;   102          0102  1 !                effected fields were changed in NDXPAG, NDXFMT, INDEX, NDXVMS
;   103          0103  1 !                and NDXCLIDMP.
;   104          0104  1 !
;   105          0105  1 !        005     JPK00010        24-Jan-1983
;   106          0106  1 !                Removed routines GETDAT and UPDDAT from NDXDAT - they
;   107          0107  1 !                performed no useful function. Removed references to these
;   108          0108  1 !                routines from NDXOUT, NDXFMT, and NDXMSG.
;   109          0109  1 !                Removed reference to XPOOL in NDXOUT - not used.
;   110          0110  1 !
;   111          0111  1 !        004     JPK00009        24-Jan-1983
;   112          0112  1 !                Modified to enhance performance. The sort buckets have each
;   113          0113  1 !                been divided into 27 sub-buckets; 1 for each letter and 1
;   114          0114  1 !                for non-alphas. Removed reference to BUCKET from INDEX.
```

```
;  115          0115  1 !                 Definition of the structure was added to NDXPOL. References
;  116          0116  1 !                 to BUCKET were changed in modules NDXOUT, NDXINI, NDXFMT
;  117          0117  1 !                 and NDXDAT.
;  118          0118  1 !
;  119          0119  1 !         003     JPK00006        24-Sep-1982
;  120          0120  1 !                 Modified NDXFMT to ignore REQUIRE file if filename string
;  121          0121  1 !                 is null. Added bold guide headings for RUNOFF output.
;  122          0122  1 !
;  123          0123  1 !         002     JPK00004        24-Sep-1982
;  124          0124  1 !                 Modified NDXOUT, NDXMSG, NDXFMT, and NDXDAT for TOPS-20.
;  125          0125  1 !                 Strings stored in the index pool use the first fullword
;  126          0126  1 !                 for their length. References to these strings were incorrect.
;  127          0127  1 !
;  128          0128  1 !--
```

```
;  130          0129  1 %SBTTL 'Module level declarations'
;  131          0130  1 !
;  132          0131  1 ! TABLE OF CONTENTS:
;  133          0132  1 !
;  134          0133  1
;  135          0134  1 FORWARD ROUTINE
;  136          0135  1     NDXFMT       : NOVALUE,              ! Generate output index
;  137          0136  1     EXAM_BUCKET  : NOVALUE,              ! Examine bucket
;  138          0137  1     FORMAT_ENTRY: NOVALUE,               ! Format an index entry
;  139          0138  1     INS_PAGE_NO  : NOVALUE,              ! Insert page number(s) in print line
;  140          0139  1     BOOK_REF     : NOVALUE,              ! Insert book name in master index
;  141          0140  1     PAGE_REF     : NOVALUE,              ! Convert page reference to ASCII
;  142          0141  1     SAMEPG,                              ! Compare page references
;  143          0142  1     SAME_ATTR,                           ! Page references have same attributes
;  144          0143  1     INS_LINE     : NOVALUE,              ! Insert line into page
;  145          0144  1     SAVE_TELLTALE_HEADING : NOVALUE,     ! Save line as telltale heading
;  146          0145  1     LAST_LINE    : NOVALUE,              ! Process last line in column
;  147          0146  1     DO_CONTINUE  : NOVALUE,              ! Generate continuation heading if necessary
;  148          0147  1     FILL_LINE    : NOVALUE,              ! Pack the print line
;  149          0148  1     INDENT_LINE  : NOVALUE,              ! Indent a new line
;  150          0149  1     PADLIN       : NOVALUE;              ! Pad line to column width
;  151          0150  1
;  152          0151  1 !
;  153          0152  1 ! INCLUDE FILES:
;  154          0153  1 !
;  155          0154  1 LIBRARY 'NXPORT:XPORT';
;  156          0155  1
;  157          0156  1 SWITCHES LIST (REQUIRE);
;  158          0157  1
;  159          0158  1 REQUIRE 'REQ:NDXPOL';
```

```
;  R0159  1
;  R0160  1        Version:      'V04-000'
;  R0161  1
;  R0162  1        ****************************************************************************
;  R0163  1        *                                                                          *
;  R0164  1        *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                *
;  R0165  1        *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
;  R0166  1        *  ALL RIGHTS RESERVED.                                                    *
;  R0167  1        *                                                                          *
;  R0168  1        *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;  R0169  1        *  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;  R0170  1        *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;  R0171  1        *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;  R0172  1        *  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;  R0173  1        *  TRANSFERRED.                                                            *
;  R0174  1        *                                                                          *
;  R0175  1        *  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;  R0176  1        *  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;  R0177  1        *  CORPORATION.                                                            *
;  R0178  1        *                                                                          *
;  R0179  1        *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;  R0180  1        *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;  R0181  1        *                                                                          *
;  R0182  1        *                                                                          *
;  R0183  1        ****************************************************************************
;  R0184  1
;  R0185  1
;  R0186  1        !++
;  R0187  1        ! FACILITY:
;  R0188  1        !   DSR (Digital Standard RUNOFF) /DSRPLUS DSRINDEX/INDEX Utility
;  R0189  1        !
;  R0190  1        ! ABSTRACT:
;  R0191  1        !   This file contains literals and macros defining the data structures
;  R0192  1        !   found in the internal index pool
;  R0193  1        !
;  R0194  1        ! ENVIRONMENT:    Transportable
;  R0195  1        !
;  R0196  1        ! AUTHOR:         JPK
;  R0197  1        !
;  R0198  1        ! CREATION DATE: January 1982
;  R0199  1        !
;  R0200  1        ! MODIFIED BY:
;  R0201  1        !
;  R0202  1        !       003     JPK00015        04-Feb-1983
;  R0203  1        !               Cleaned up module names, modified revision history to
;  R0204  1        !               conform with established standards. Updated copyright dates.
;  R0205  1        !
;  R0206  1        !       002     JPK00009        24-Jan-1983
;  R0207  1        !               Modified to enhance performance. The sort buckets have each
;  R0208  1        !               been divided into 27 sub-buckets; 1 for each letter and 1
;  R0209  1        !               for non-alphas. Removed reference to BUCKET from INDEX.
;  R0210  1        !               Definition of the structure was added to NDXPOL. References
;  R0211  1        !               to BUCKET were changed in modules NDXOUT, NDXINI, NDXFMT
;  R0212  1        !               and NDXDAT.
;  R0213  1        !
;  R0214  1        !--
;  R0215  1
```

```
;  R0216  1     ! Index entry
;  R0217  1
;  R0218  1     $FIELD XE_FIELDS =
;  R0219  1         SET
;  R0220  1
;  R0221  1         XE$A_PREV           = [$ADDRESS],           ! Link to previous item
;  R0222  1         XE$A_NEXT           = [$ADDRESS],           ! Link to next item
;  R0223  1         XE$A_SUBX           = [$ADDRESS],           ! Sub index pointer
;  R0224  1         XE$A_REF            = [$ADDRESS],           ! Reference pointer
;  R0225  1         XE$A_TEXT           = [$ADDRESS],           ! Pointer to text of index item
;  R0226  1         XE$A_SORT_AS        = [$ADDRESS],           ! Pointer to SORT_AS string
;  R0227  1         XE$H_SUBC           = [$SHORT_INTEGER],     ! Sub index level
;  R0228  1
;  R0229  1         XE$V_FLAGS          = [$SHORT_INTEGER],     ! Entry flags
;  R0230  1
;  R0231  1             $OVERLAY (XE$V_FLAGS)
;  R0232  1
;  R0233  1             XE$V_BARS               = [$BIT],       ! Change bar flag
;  R0234  1
;  R0235  1             $CONTINUE
;  R0236  1
;  R0237  1         XE$A_BOOK_LIST      = [$ADDRESS]            ! Master index book name list
;  R0238  1
;  R0239  1         $ALIGN (FULLWORD)
;  R0240  1
;  R0241  1         TES;
;  R0242  1
;  R0243  1     LITERAL
;  R0244  1         XE$K_LENGTH = $FIELD_SET_SIZE;
;  R0245  1
;  R0246  1     MACRO
;  R0247  1         $XE_BLOCK = BLOCK [XE$K_LENGTH] FIELD (XE_FIELDS) %;
;  R0248  1
;  R0249  1     ! End of Index entry
;  R0250  1
;  R0251  1
;  R0252  1     ! Reference entry
;  R0253  1
;  R0254  1     $FIELD XX_FIELDS =
;  R0255  1         SET
;  R0256  1
;  R0257  1         XX$A_LINK           = [$ADDRESS],           ! Link to additional entries
;  R0258  1         XX$A_APPEND         = [$ADDRESS],           ! APPEND text pointer
;  R0259  1         XX$H_PAGE           = [$SHORT_INTEGER],     ! Transaction number
;  R0260  1
;  R0261  1         XX$V_FLAGS          = [$SHORT_INTEGER],     ! Display attributes
;  R0262  1
;  R0263  1             $OVERLAY (XX$V_FLAGS)
;  R0264  1
;  R0265  1             XX$V_BOLD       = [$BIT],               ! Bold page reference
;  R0266  1             XX$V_UNDERLINE  = [$BIT],               ! Underline page reference
;  R0267  1             XX$V_BEGIN      = [$BIT],               ! Begin page range
;  R0268  1             XX$V_END        = [$BIT],               ! End page range
;  R0269  1
;  R0270  1             $CONTINUE
;  R0271  1
;  R0272  1         XX$A_BOOK           = [$ADDRESS]            ! Master index book name
```

```
;  R0273  1
;  R0274  1          $ALIGN (FULLWORD)
;  R0275  1
;  R0276  1          TES;
;  R0277  1
;  R0278  1      LITERAL
;  R0279  1          XX$K_LENGTH = $FIELD_SET_SIZE;
;  R0280  1
;  R0281  1      MACRO
;  R0282  1          $XX_BLOCK = BLOCK [XX$K_LENGTH] FIELD (XX_FIELDS) %;
;  R0283  1
;  R0284  1      ! End of Reference entry
;  R0285  1
;  R0286  1
;  R0287  1      ! Master index book reference entry
;  R0288  1
;  R0289  1      $FIELD XM_FIELDS =
;  R0290  1          SET
;  R0291  1
;  R0292  1          XM$A_LINK           = [$ADDRESS],           ! Link to additional entries
;  R0293  1          XM$A_BOOK           = [$ADDRESS]            ! Pointer to book name
;  R0294  1
;  R0295  1          TES;
;  R0296  1
;  R0297  1      LITERAL
;  R0298  1          XM$K_LENGTH = $FIELD_SET_SIZE;
;  R0299  1
;  R0300  1      MACRO
;  R0301  1          $XM_BLOCK = BLOCK [XM$K_LENGTH] FIELD (XM_FIELDS) %;
;  R0302  1
;  R0303  1      ! End of Master index book reference entry
;  R0304  1
;  R0305  1
;  R0306  1      ! Current Entry
;  R0307  1
;  R0308  1      $FIELD C_FIELDS =
;  R0309  1          SET
;  R0310  1
;  R0311  1          C$A_CURR            = [$ADDRESS],           ! Pointer to current cell
;  R0312  1          C$A_PREV            = [$ADDRESS],           ! Pointer to previous cell
;  R0313  1          C$A_HEAD            = [$ADDRESS],           ! Pointer to head of chain
;  R0314  1
;  R0315  1          $ALIGN (FULLWORD)
;  R0316  1
;  R0317  1          C$V_FLAGS           = [$INTEGER],           ! Current cell flags
;  R0318  1
;  R0319  1              $OVERLAY (C$V_FLAGS)
;  R0320  1
;  R0321  1              C$V_IDNS        = [$BIT]                ! Identical string flag
;  R0322  1
;  R0323  1              $CONTINUE
;  R0324  1
;  R0325  1          TES;
;  R0326  1
;  R0327  1      LITERAL
;  R0328  1          C$K_LENGTH = $FIELD_SET_SIZE;
;  R0329  1
```

```
;  R0330  1     MACRO
;  R0331  1         $C_BLOCK = BLOCK [C$K_LENGTH] FIELD (C_FIELDS) %;
;  R0332  1
;  R0333  1     ! End of current entry
;  R0334  1
;  R0335  1
;  R0336  1     !
;  R0337  1     ! Dummy datasets
;  R0338  1     !
;  R0339  1     LITERAL
;  R0340  1         DS_X_ENTRY  = XE$K_LENGTH,
;  R0341  1         DS_XX_ENTRY = XX$K_LENGTH,
;  R0342  1         DS_XM_ENTRY = XM$K_LENGTH,
;  R0343  1         DS_X_STRING = 0;
;  R0344  1
;  R0345  1     !
;  R0346  1     ! Structure definition for bucket array.
;  R0347  1     !
;  R0348  1     !       Buckets are arranged so that each row represents the first letter of
;  R0349  1     !       the string and each column represents the second letter of the string.
;  R0350  1     !
;  R0351  1     !       This approach is used only for master indexes as no performance
;  R0352  1     !       improvement is realised until about 10 input files have been processed.
;  R0353  1     !
;  R0354  1     !       Indexes which are not master indexes use only the first element of
;  R0355  1     !       each row, i.e., [0, 0] ... [26, 0].
;  R0356  1     !
;  R0357  1     !       The only exception is for nonalphabetic characters which use only
;  R0358  1     !       element [0, 0]. Elements [0, 1] ... [0, 26] are not used since mapping
;  R0359  1     !       all nonalphabetics into one row loses the sort order of the first
;  R0360  1     !       character in the string. For nonalphabetics to work correctly in a two
;  R0361  1     !       dimensional bucket scheme, the array would have to be at least 127 x 127
;  R0362  1     !
;  R0363  1     !            0   1   .   .   .  26
;  R0364  1     !          0 **  not used  . . .
;  R0365  1     !          1 A?  AA          AZ
;  R0366  1     !          .                 .
;  R0367  1     !          .                 .
;  R0368  1     !          .                 .
;  R0369  1     !         26 Z?  ZA  .   .   .  ZZ
;  R0370  1     !
;  R0371  1     STRUCTURE
;  R0372  1         $BUCKET_ARRAY [ROW_IDX, COL_IDX; M, N] =
;  R0373  1             [M * N * %UPVA[] ($BUCKET_ARRAY + (ROW_IDX * N + COL_IDX) * %UPVAL);
;  R0374  1
;  R0375  1     !--     End of NDXPOL.REQ
```

```
;  160          0376  1
;  161          0377  1 REQUIRE 'REQ:NDXCLI';
```

```
;  R0378  1        !                                  IDENT = 0V04-00004
;  R0379  1        !
;  R0380  1        !*****************************************************************************
;  R0381  1        !*                                                                           *
;  R0382  1        !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;  R0383  1        !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
;  R0384  1        !*  ALL RIGHTS RESERVED.                                                     *
;  R0385  1        !*                                                                           *
;  R0386  1        !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
;  R0387  1        !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
;  R0388  1        !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
;  R0389  1        !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
;  R0390  1        !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
;  R0391  1        !*  TRANSFERRED.                                                             *
;  R0392  1        !*                                                                           *
;  R0393  1        !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
;  R0394  1        !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
;  R0395  1        !*  CORPORATION.                                                             *
;  R0396  1        !*                                                                           *
;  R0397  1        !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
;  R0398  1        !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
;  R0399  1        !*                                                                           *
;  R0400  1        !*                                                                           *
;  R0401  1        !*****************************************************************************
;  R0402  1        !
;  R0403  1
;  R0404  1        !++
;  R0405  1        ! FACILITY:
;  R0406  1        !   DSR (Digital Standard RUNOFF) /DSRPLUS DSRINDEX/INDEX Utility
;  R0407  1        !
;  R0408  1        ! ABSTRACT:       INDEX command line definitions
;  R0409  1        !
;  R0410  1        ! ENVIRONMENT:    Transportable
;  R0411  1        !
;  R0412  1        ! AUTHOR:         JPK
;  R0413  1        !
;  R0414  1        ! CREATION DATE: January 1982
;  R0415  1        !
;  R0416  1        ! MODIFIED BY:
;  R0417  1        !
;  R0418  1        !       004     JPK00015        04-Feb-1983
;  R0419  1        !               Cleaned up module names, modified revision history to
;  R0420  1        !               conform with established standards. Updated copyright dates.
;  R0421  1        !
;  R0422  1        !       003     JPK00011        24-Jan-1983
;  R0423  1        !               Changed CMDBLK [NDX$G_LEVEL] to CMDBLK [NDX$H_LEVEL]
;  R0424  1        !               Changed CMDBLK [NDX$H_FORMAT] to CMDBLK [NDX$H_LAYOUT]
;  R0425  1        !               Changed CMDBLK [NDX$V_TMS11] and CMDBLK [NDX$V_TEX] to CMDBLK [NDX$H_FORMAT]
;  R0426  1        !               Changed comparisons of (.CHRSIZ EQLA CHRSZA) to
;  R0427  1        !               (.CMDBLK [NDX$H_FORMAT] EQL TMS11_A).
;  R0428  1        !               Definitions were changed in NDXCLI and references to the
;  R0429  1        !               effected fields were changed in NDXPAG, NDXFMT, INDEX, NDXVMS
;  R0430  1        !               and NDXCLIDMP.
;  R0431  1        !
;  R0432  1        !       002     RER00002        20-Jan-1983
;  R0433  1        !               Modified VMS command line interface module NDXVMS:
;  R0434  1        !                 - changed /FORMAT qualifier to /LAYOUT.
```

```
;  R0435  1        !                  - changed use of /RESERVE and /REQUIRE for DSRPLUS.
;  R0436  1        !                  - added code for new DSRPLUS qualifiers /FORMAT and
;  R0437  1        !                    /TELLTALE_HEADINGS.
;  R0438  1        !                Added fields to NDXCLI for new qualifiers: NDX$V_TELLTALE
;  R0439  1        !                and NDX$V_TEX.
;  R0440  1        !                Conditionalized output of NDX$V_PAGE_MERGE in NDXCLIDMP to
;  R0441  1        !                account for different DSR and DSRPLUS default values.
;  R0442  1        !
;  R0443  1        !--
;  R0444  1
```

```
;  R0445  1      !
;  R0446  1      !  NDXCMD_FIELDS
;  R0447  1      !
;  R0448  1      $FIELD ndxcmd_fields =
;  R0449  1          SET
;  R0450  1
;  R0451  1          NDX$V_OPTIONS       = [$INTEGER],          ! Command option indicators:
;  R0452  1
;  R0453  1              $OVERLAY (NDX$V_OPTIONS)
;  R0454  1
;  R0455  1              NDX$V_INPUT_CONCAT      = [$BIT],      ! Input file concatenated to previous
;  R0456  1              NDX$V_OUTPUT            = [$BIT],      ! Generate output file
;  R0457  1              NDX$V_REQUIRE           = [$BIT],      ! Require file specified
;  R0458  1              NDX$V_PAGES             = [$BIT],      ! Include page references in index
;  R0459  1              NDX$V_OVERRIDE          = [$BIT],      ! Override master index information
;  R0460  1              NDX$V_STANDARD_PAGE     = [$BIT],      ! Generate standard page numbers
;  R0461  1              NDX$V_CONTINUATION      = [$BIT],      ! Generate continuation headings
;  R0462  1              NDX$V_GUIDE             = [$BIT],      ! Generate guide headings
;  R0463  1              NDX$V_WORD_SORT         = [$BIT],      ! Sort entries word by word
;  R0464  1              NDX$V_LOG               = [$BIT],      ! Generate /LOG message
;  R0465  1              NDX$V_MASTER            = [$BIT],      ! Generate a master index
;  R0466  1              NDX$V_PAGE_MERGE        = [$BIT],      ! Merge adjacent page references
;  R0467  1              NDX$V_TELLTALE          = [$BIT],      ! Generate telltale headings
;  R0468  1
;  R0469  1              $CONTINUE
;  R0470  1
;  R0471  1          NDX$H_FORMAT        = [$SHORT_INTEGER],    ! Output format: DSR, TMS, TEX
;  R0472  1          NDX$H_LAYOUT        = [$SHORT_INTEGER],    ! Output layout type
;  R0473  1          NDX$H_NONALPHA      = [$SHORT_INTEGER],    ! Treatment of leading nonalphas during sort
;  R0474  1          NDX$H_LEVEL         = [$SHORT_INTEGER],    ! Deepest level to include in index
;  R0475  1          NDX$G_COLUMN_WID    = [$INTEGER],          ! Column width
;  R0476  1          NDX$G_GUTTER_WID    = [$INTEGER],          ! Gutter width
;  R0477  1          NDX$G_LINES_PAGE    = [$INTEGER],          ! Lines per page
;  R0478  1          NDX$G_RESERVE_LINES = [$INTEGER],          ! Number of lines to reserve when requiring a file
;  R0479  1          NDX$G_SEPARATE_WIDTH= [$INTEGER],          ! Width of reference portion of entry
;  R0480  1          NDX$T_MASTER_BOOK   = [$DESCRIPTOR(DYNAMIC)], ! Book name descriptor for Master indexing
;  R0481  1          NDX$T_INPUT_FILE    = [$DESCRIPTOR(DYNAMIC)], ! Input file name descriptor
;  R0482  1          NDX$T_OUTPUT_FILE   = [$DESCRIPTOR(DYNAMIC)], ! Output file name descriptor
;  R0483  1          NDX$T_REQUIRE_FILE  = [$DESCRIPTOR(DYNAMIC)], ! Require file name descriptor
;  R0484  1          NDX$T_RELATED_FILE  = [$DESCRIPTOR(DYNAMIC)], ! Related file name descriptor is saved here
;  R0485  1                                                        ! by NDXINP for later use by MAKNDX
;  R0486  1          NDX$T_COMMAND_LINE  = [$DESCRIPTOR(DYNAMIC)]  ! Copy of entire command line
;  R0487  1
;  R0488  1              TES;
;  R0489  1      !
;  R0490  1      !  End of NDXCMD_FIELDS
;  R0491  1      !
;  R0492  1
;  R0493  1      LITERAL
;  R0494  1          NDXCMD$K_LENGTH = $FIELD_SET_SIZE;
;  R0495  1
;  R0496  1      MACRO
;  R0497  1          $NDXCMD = BLOCK [NDXCMD$K_LENGTH] FIELD (NDXCMD_FIELDS) %;
;  R0498  1
;  R0499  1      $LITERAL                                       ! Output formats (NDX$H_FORMAT)
;  R0500  1          DSR                 = $DISTINCT,         ! Runoff
;  R0501  1          TMS11_A             = $DISTINCT,         ! TMS=A
```

```
;  R0502  1          TMS11_E                 = $DISTINCT,    ! TMS=E
;  R0503  1          TEX                     = $DISTINCT;    ! TEX
;  R0504  1
;  R0505  1      $LITERAL                                    ! Output layouts (NDX$H_LAYOUT)
;  R0506  1          TWO_COLUMN              = $DISTINCT,    ! Normal two column format
;  R0507  1          ONE_COLUMN              = $DISTINCT,    ! Normal one column format
;  R0508  1          SEPARATE                = $DISTINCT,    ! Separate reference format
;  R0509  1          GALLEY                  = $DISTINCT;    ! TMS11 Galley format
;  R0510  1
;  R0511  1      $LITERAL                                    ! Treatment of leading nonalphas during sort (NDX$H_NONALPHA)
;  R0512  1          BEFORE                  = $DISTINCT,    ! Leading nonalphas sort before alphas
;  R0513  1          AFTER                   = $DISTINCT,    ! Leading nonalphas sort after alphas
;  R0514  1          IGNORE                  = $DISTINCT;    ! Leading nonalphas are ignored
;  R0515  1
;  R0516  1      !
;  R0517  1      !--     End of NDXCLI.REQ
```

```
;  162          0518  1
;  163          0519  1 REQUIRE 'REQ:NDXLIN';
```

```
;  R0520  1        !                                IDENT = 0V04-00002
;  R0521  1        !
;  R0522  1        !*****************************************************************************
;  R0523  1        !*                                                                           *
;  R0524  1        !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;  R0525  1        !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
;  R0526  1        !*  ALL RIGHTS RESERVED.                                                     *
;  R0527  1        !*                                                                           *
;  R0528  1        !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
;  R0529  1        !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
;  R0530  1        !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
;  R0531  1        !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
;  R0532  1        !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
;  R0533  1        !*  TRANSFERRED.                                                             *
;  R0534  1        !*                                                                           *
;  R0535  1        !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
;  R0536  1        !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
;  R0537  1        !*  CORPORATION.                                                             *
;  R0538  1        !*                                                                           *
;  R0539  1        !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
;  R0540  1        !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
;  R0541  1        !*                                                                           *
;  R0542  1        !*                                                                           *
;  R0543  1        !*****************************************************************************
;  R0544  1        !
;  R0545  1
;  R0546  1        !++
;  R0547  1        ! FACILITY:
;  R0548  1        !   DSR (Digital Standard RUNOFF) /DSRPLUS DSRINDEX/INDEX Utility
;  R0549  1        !
;  R0550  1        ! ABSTRACT:
;  R0551  1        !   Contains output line type definitions.
;  R0552  1        !
;  R0553  1        ! ENVIRONMENT:    Transportable
;  R0554  1        !
;  R0555  1        ! AUTHOR:         JPK
;  R0556  1        !
;  R0557  1        ! CREATION DATE: January 1982
;  R0558  1        !
;  R0559  1        ! MODIFIED BY:
;  R0560  1        !
;  R0561  1        !       002     JPK00010        04-Feb-1983
;  R0562  1        !               Cleaned up module names, modified revision history to
;  R0563  1        !               conform with established standards. Updated copyright dates.
;  R0564  1        !
;  R0565  1        !--
;  R0566  1
;  R0567  1        LITERAL                                         !Output line types:
;  R0568  1            BKT_E = 1,                                  !Bucket end blank line
;  R0569  1            FILL = 2,                                   !Fill line
;  R0570  1            GUIDE = 3,                                  !Guide heading
;  R0571  1            GUIDE_FILL = 4,                             !Blank line after guide heading
;  R0572  1            ENTRY_B = 5,                                !Beginning of top level entry
;  R0573  1            ENTRY_W = 6,                                !Wrap line of top level entry
;  R0574  1            ENTRY_E = 7,                                !Last line of top level entry
;  R0575  1            SUB_B = 8,                                  !Beginning of subentry
;  R0576  1            SUB_W = 9,                                  !Wrap line of subentry
```

```
;  R0577  1          SUB_E = 10,                                 .Last line of subentry
;  R0578  1          CONT_HEAD = 11;                             !Continuation heading
;  R0579  1
;  R0580  1      !
;  R0581  1      !--     End of NDXLIN.REQ
```

```
;  164          0582  1
;  165          0583  1 REQUIRE 'REQ:PAGEN';
```

```
;  R0584  1     !
;  R0585  1     ! Version:      'V04-000'
;  R0586  1     !
;  R0587  1     !****************************************************************************
;  R0588  1     !*                                                                          *
;  R0589  1     !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
;  R0590  1     !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
;  R0591  1     !*  ALL RIGHTS RESERVED.                                                    *
;  R0592  1     !*                                                                          *
;  R0593  1     !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;  R0594  1     !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
;  R0595  1     !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;  R0596  1     !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;  R0597  1     !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;  R0598  1     !*  TRANSFERRED.                                                            *
;  R0599  1     !*                                                                          *
;  R0600  1     !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;  R0601  1     !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;  R0602  1     !*  CORPORATION.                                                            *
;  R0603  1     !*                                                                          *
;  R0604  1     !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;  R0605  1     !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;  R0606  1     !*                                                                          *
;  R0607  1     !*                                                                          *
;  R0608  1     !****************************************************************************
;  R0609  1     !
;  R0610  1
;  R0611  1     !++
;  R0612  1     ! FACILITY:      DSR (Digital Standard RUNOFF) / DSRPLUS
;  R0613  1     !
;  R0614  1     ! ABSTRACT:
;  R0615  1     !       A page number carries with it not only its current value, but also
;  R0616  1     !       codes as to how those values are to be displayed when they are finally
;  R0617  1     !       output. It was decided to do it this way rather than have a separate
;  R0618  1     !       table so that the program TCX would have less trouble.
;  R0619  1     !
;  R0620  1     ! ENVIRONMENT:   Transportable BLISS
;  R0621  1     !
;  R0622  1     ! AUTHOR:        Rich Friday
;  R0623  1     !
;  R0624  1     ! CREATION DATE: 1978
;  R0625  1     !
;  R0626  1     ! MODIFIED BY:
;  R0627  1     !
;  R0628  1     !       004     KAD00004        Keith Dawson    07-Mar-1983
;  R0629  1     !               Global edit of all modules. Updated module names, idents,
;  R0630  1     !               copyright dates. Changed require files to BLISS library.
;  R0631  1     !
;  R0632  1     !--
;  R0633  1
;  R0634  1     LITERAL
;  R0635  1         page_sct_size = 4;
;  R0636  1
;  R0637  1     LITERAL                                         !Type of section:
;  R0638  1         sct_chapt     = 1,                         ! Chapter section.
;  R0639  1         sct_index     = 2,                         ! Index section.
;  R0640  1         sct_append    = 3;                         ! Appendix section.
```

```
;  R0641  1
;  R0642  1     LITERAL
;  R0643  1         sct_low       = 1,                          ! Lowest section type key.
;  R0644  1         sct_high      = 3;                          ! Highest section type key.
;  R0645  1
;  R0646  1     MACRO
;  R0647  1         sct_typ       = 0, 0,         4,        0 %, ! Section Type (zero if none).
;  R0648  1         sct_page_d    = 0, 4,         4,        0 %, ! Display code for page number.
;  R0649  1         sct_sub_page  = 0, %BPVAL/2,  %BPVAL/2, 0 %, ! Subpage, if any (zero if none).
;  R0650  1         sct_number    = 1, 0,         %BPVAL,   0 %, ! Type of section number.
;  R0651  1         sct_page      = 2, 0,         %BPVAL,   0 %, ! Page number.
;  R0652  1         sct_subpg_d   = 3, 0,         4,        0 %, ! Display code for subpages.
;  R0653  1         sct_chapt_d   = 3, 4,         4,        0 %, ! Display code for chapters.
;  R0654  1         sct_appen_d   = 3, 8,         4,        0 %, ! Display code for appendices.
;  R0655  1         sct_index_d   = 3, 12,        4,        0 %; ! Display code if indexes.
;  R0656  1
;  R0657  1     MACRO
;  R0658  1         sct_run_page  = 3, %BPVAL/2, %BPVAL/2, 0 %; ! Running page number.
;  R0659  1
;  R0660  1     MACRO
;  R0661  1         page_definition = BLOCK [page_sct_size] %;
;  R0662  1
;  R0663  1     !                          End of PAGEN.REQ
```

```
;  166          0664  1
;  167        L 0665  1 %IF %BLISS (BLISS32)
;  168          0666  1 %THEN
;  169          0667  1
;  170          0668  1 REQUIRE 'REQ:NDXVMSREQ';
```

```
;  R0669  1
;  R0670  1        Version:      'V04-000'
;  R0671  1
;  R0672  1        ****************************************************************************
;  R0673  1        *                                                                          *
;  R0674  1        *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                *
;  R0675  1        *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
;  R0676  1        *  ALL RIGHTS RESERVED.                                                    *
;  R0677  1        *                                                                          *
;  R0678  1        *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;  R0679  1        *  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;  R0680  1        *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;  R0681  1        *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;  R0682  1        *  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;  R0683  1        *  TRANSFERRED.                                                            *
;  R0684  1        *                                                                          *
;  R0685  1        *  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;  R0686  1        *  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;  R0687  1        *  CORPORATION.                                                            *
;  R0688  1        *                                                                          *
;  R0689  1        *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;  R0690  1        *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;  R0691  1        *                                                                          *
;  R0692  1        *                                                                          *
;  R0693  1        ****************************************************************************
;  R0694  1
;  R0695  1
;  R0696  1        ++
;  R0697  1        FACILITY:
;  R0698  1          DSR (Digital Standard RUNOFF) /DSRPLUS DSRINDEX/INDEX Utility
;  R0699  1
;  R0700  1        ABSTRACT:
;  R0701  1              This file contains external references to the error message numbers
;  R0702  1              for DSRINDEX/INDEX.
;  R0703  1
;  R0704  1              New messages must be defined in NDXVMSMSG.MSG and referenced here
;  R0705  1              both in the MACRO section (for DSRINDEX) and the EXTERNAL LITERAL
;  R0706  1              section (for INDEX)
;  R0707  1
;  R0708  1        ENVIRONMENT:   VAX/VMS User Mode
;  R0709  1
;  R0710  1        AUTHOR:        JPK
;  R0711  1
;  R0712  1        CREATION DATE: 01-Feb-1983
;  R0713  1
;  R0714  1        MODIFIED BY:
;  R0715  1
;  R0716  1              004      JPK00022        30-Mar-1983
;  R0717  1                       Modified NDXVMS, NDXFMT, NDXPAG, NDXVMSMSG and NDXVMSREQ
;  R0718  1                       to generate TEX output. Added module NDXTEX.
;  R0719  1
;  R0720  1              003      JPK00021        28-Mar-1983
;  R0721  1                       Modified NDXT20 to include E2.0 functionality.
;  R0722  1                       Modified NDXCLIDMP, NDXFMT, NDXPAG, NDXVRS to require RNODEF
;  R0723  1                       for BLISS36 and to remove any conditional require based on
;  R0724  1                       DSRPLUS_DEF.
;  R0725  1
```

```
;  R0726  1      !       002     JPK00010        04-Feb-1983
;  R0727  1      !               Cleaned up module names, modified revision history to
;  R0728  1      !               conform with established standards. Updated copyright dates.
;  R0729  1      !
;  R0730  1      !--
;  R0731  1
;  R0732  1      REQUIRE 'REQ:RNODEF';
```

```
;  R0733  1     !
;  R0734  1     ! Version:      'V04-000'
;  R0735  1     !
;  R0736  1     !****************************************************************************
;  R0737  1     !*                                                                          *
;  R0738  1     !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
;  R0739  1     !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
;  R0740  1     !*  ALL RIGHTS RESERVED.                                                    *
;  R0741  1     !*                                                                          *
;  R0742  1     !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;  R0743  1     !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
;  R0744  1     !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;  R0745  1     !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;  R0746  1     !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;  R0747  1     !*  TRANSFERRED.                                                            *
;  R0748  1     !*                                                                          *
;  R0749  1     !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;  R0750  1     !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;  R0751  1     !*  CORPORATION.                                                            *
;  R0752  1     !*                                                                          *
;  R0753  1     !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;  R0754  1     !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;  R0755  1     !*                                                                          *
;  R0756  1     !*                                                                          *
;  R0757  1     !****************************************************************************
;  R0758  1     !
;  R0759  1
;  R0760  1     !++
;  R0761  1     ! FACILITY:      DSR (Digital Standard RUNOFF) / DSRPLUS
;  R0762  1     !
;  R0763  1     ! ABSTRACT:
;  R0764  1     !       Converts BLISS/VARIANT values into useful names.
;  R0765  1     !
;  R0766  1     ! ENVIRONMENT:   Transportable BLISS
;  R0767  1     !
;  R0768  1     ! AUTHOR:        Rich Friday
;  R0769  1     !
;  R0770  1     ! CREATION DATE: 1978
;  R0771  1     !
;  R0772  1     ! MODIFIED BY:
;  R0773  1     !
;  R0774  1     !       016     KAD00016        Ray Marshall    19-Mar-1984
;  R0775  1     !               Added GERMAN, FRENCH, & ITALIAN.
;  R0776  1     !
;  R0777  1     !       015     KAD00015        Keith Dawson    18-Apr-1983
;  R0778  1     !               Made the LN01 conditional the default for vanilla DSR --
;  R0779  1     !                 its value is 0 (no variant supplied).
;  R0780  1     !
;  R0781  1     !       014     KAD00014        Keith Dawson    22-Mar-1983
;  R0782  1     !               Asserted the LN01 conditional when DSRPLUS is asserted.
;  R0783  1     !
;  R0784  1     !       013     KAD00013        Keith Dawson    20-Mar-1983
;  R0785  1     !               Removed all references to .BIX and .BTC files.
;  R0786  1     !
;  R0787  1     !       012     KAD00012        Keith Dawson    07-Mar-1983
;  R0788  1     !               Global edit of all modules. Updated module names, idents,
;  R0789  1     !               copyright dates. Changed require files to BLISS library.
```

```
;  R0790  1     !
;  R0791  1     !--
;  R0792  1
;  R0793  1     !++
;  R0794  1     !       D E F I N I T I O N   O F  /VARIANT   B I T S
;  R0795  1     !
;  R0796  1     !       The bit assignments are as follows:
;  R0797  1     !
;  R0798  1     !       Bit  Weight   Meaning
;  R0799  1     !-------------------------------------------------------------------------
;  R0800  1     !       --      0     I  o /VARIANT is supplied (as for vanilla DSR),
;  R0801  1     !                     compile with LN01 support. LN01 support is also
;  R0802  1     !                     implied by the DSRPLUS variant.
;  R0803  1     !
;  R0804  1     !        0      1     CLEAR =   Unassigned
;  R0805  1     !                     SET   =   Unassigned
;  R0806  1     !
;  R0807  1     !        1      2     CLEAR =   Normal compile
;  R0808  1     !                     SET   =   Compile for DSRPLUS
;  R0809  1     !
;  R0810  1     !       4-6    16     CLEAR =   English (American) version
;  R0811  1     !                     SET   =   16 = German (Austrian)
;  R0812  1     !                               32 = French
;  R0813  1     !                               48 = Italian
;  R0814  1     !--
;  R0815  1
;  R0816  1     !-------------------------------------------------------------------------
;  R0817  1     !   This variable (LN01) controls whether or not to compile an LN01-flavored
;  R0818  1     !   DSR. It is asserted by default, and also whenever DSRPLUS is asserted.
;  R0819  1     !
;  R0820  1     !   Modules utilizing LN01 are:
;  R0821  1     !
;  R0822  1     !       DOOPTS  NOUT
;  R0823  1
;  R0824  1     COMPILETIME
;  R0825  1         ln01 =
;  R0826  2             ( (%VARIANT EQL 0) OR  %VARIANT/2 )
;  R0827  1         ;
;  R0828  1
;  R0829  1     !-------------------------------------------------------------------------
;  R0830  1     !   This variable (DSRPLUS) controls compilation for the DSRPLUS program.
;  R0831  1     !
;  R0832  1     !   All modules utilize DSRPLUS.
;  R0833  1
;  R0834  1     COMPILETIME
;  R0835  1         dsrplus =
;  R0836  2             ( %VARIANT/2 )
;  R0837  1         ;
;  R0838  1
;  R0839  1     !-------------------------------------------------------------------------
;  R0840  1     !   This variable (FLIP) controls compilation of FLIP features of DSRPLUS.
;  R0841  1     !   It assures that FLIP features are compiled only on VMS systems.
;  R0842  1     !
;  R0843  1     !   Modules utilizing FLIP are many and various.
;  R0844  1
;  R0845  1     COMPILETIME
;  R0846  1         flip =
```

```
;  R0847  2              ( %VARIANT/2 AND %BLISS(BLISS32) )
;  R0848  1          ;
;  R0849  1
;  R0850  1      !----------------------------------------------------------------------
;  R0851  1      !       4-6    16     CLEAR =   English (American) version
;  R0852  1      !                     SET   =   16 = German (Austrian)
;  R0853  1      !                               32 = French
;  R0854  1      !                               48 = Italian
;  R0855  1      COMPILETIME
;  R0856  1          German  = ( %VARIANT/16 AND NOT %VARIANT/32 AND NOT %VARIANT/64 ) ;
;  R0857  1      COMPILETIME
;  R0858  1          French  = ( NOT %VARIANT/16 AND %VARIANT/32 AND NOT %VARIANT/64 ) ;
;  R0859  1      COMPILETIME
;  R0860  1          Italian = ( %VARIANT/16 AND %VARIANT/32 AND NOT %VARIANT/64 ) ;
;  R0861  1      !----------------------------------------------------------------------
;  R0862  1      !                       End of RNODEF.REQ
```

```
;  R0863  1
; LR0864  1     %IF NOT DSRPLUS
;  R0865  1     %THEN
;  R0866  1
;  R0867  1     MACRO
;  R0868  1          INDEX$_BADLOGIC   = DSRINDEX$_BADLOGIC   %,
;  R0869  1          INDEX$_BADVALUE   = DSRINDEX$_BADVALUE   %,
;  R0870  1          INDEX$_INSVIRMEM  = DSRINDEX$_INSVIRMEM  %,
;  R0871  1          INDEX$_LINELENG   = DSRINDEX$_LINELENG   %,
;  R0872  1          INDEX$_NOREF      = DSRINDEX$_NOREF      %,
;  R0873  1          INDEX$_OPENIN     = DSRINDEX$_OPENIN     %,
;  R0874  1          INDEX$_OPENOUT    = DSRINDEX$_OPENOUT    %,
;  R0875  1          INDEX$_TOOMANY    = DSRINDEX$_TOOMANY    %,
;  R0876  1          INDEX$_VALERR     = DSRINDEX$_VALERR     %,
;  R0877  1          INDEX$_CANTBAL    = DSRINDEX$_CANTBAL    %,
;  R0878  1          INDEX$_CLOSEQUOT  = DSRINDEX$_CLOSEQUOT  %,
;  R0879  1          INDEX$_CONFQUAL   = DSRINDEX$_CONFQUAL   %,
;  R0880  1          INDEX$_CTRLCHAR   = DSRINDEX$_CTRLCHAR   %,
;  R0881  1          INDEX$_DOESNTFIT  = DSRINDEX$_DOESNTFIT  %,
;  R0882  1          INDEX$_DUPBEGIN   = DSRINDEX$_DUPBEGIN   %,
;  R0883  1          INDEX$_EMPTYIN    = DSRINDEX$_EMPTYIN    %,
;  R0884  1          INDEX$_IGNORED    = DSRINDEX$_IGNORED    %,
;  R0885  1          INDEX$_INVINPUT   = DSRINDEX$_INVINPUT   %,
;  R0886  1          INDEX$_INVRECORD  = DSRINDEX$_INVRECORD  %,
;  R0887  1          INDEX$_LASTCONT   = DSRINDEX$_LASTCONT   %,
;  R0888  1          INDEX$_NOBEGIN    = DSRINDEX$_NOBEGIN    %,
;  R0889  1          INDEX$_NOEND      = DSRINDEX$_NOEND      %,
;  R0890  1          INDEX$_NOINDEX    = DSRINDEX$_NOINDEX    %,
;  R0891  1          INDEX$_NOLIST     = DSRINDEX$_NOLIST     %,
;  R0892  1          INDEX$_OVERSTRK   = DSRINDEX$_OVERSTRK   %,
;  R0893  1          INDEX$_SKIPPED    = DSRINDEX$_SKIPPED    %,
;  R0894  1          INDEX$_SYNTAX     = DSRINDEX$_SYNTAX     %,
;  R0895  1          INDEX$_TEXFILE    = DSRINDEX$_TEXFILE    %,
;  R0896  1          INDEX$_TOODEEP    = DSRINDEX$_TOODEEP    %,
;  R0897  1          INDEX$_TOOFEW     = DSRINDEX$_TOOFEW     %,
;  R0898  1          INDEX$_TRUNCATED  = DSRINDEX$_TRUNCATED  %,
;  R0899  1          INDEX$_COMPLETE   = DSRINDEX$_COMPLETE   %,
;  R0900  1          INDEX$_CREATED    = DSRINDEX$_CREATED    %,
;  R0901  1          INDEX$_IDENT      = DSRINDEX$_IDENT      %,
;  R0902  1          INDEX$_PROCFILE   = DSRINDEX$_PROCFILE   %,
;  R0903  1          INDEX$_TEXT       = DSRINDEX$_TEXT       %,
;  R0904  1          INDEX$_TEXTD      = DSRINDEX$_TEXTD      %,
;  R0905  1          INDEX$_TMS11      = DSRINDEX$_TMS11      %;
;  R0906  1
;  R0907  1     %FI
;  R0908  1
;  R0909  1     EXTERNAL LITERAL
;  R0910  1          INDEX$_BADLOGIC,     ! <internal logic error detected>
;  R0911  1          INDEX$_BADVALUE,     ! <'!AS' is an invalid keyword value>
;  R0912  1          INDEX$_INSVIRMEM,    ! <insufficient virtual memory>
;  R0913  1          INDEX$_LINELENG,     ! <maximum line length is 120>
;  R0914  1          INDEX$_NOREF,        ! <page reference not found>
;  R0915  1          INDEX$_OPENIN,       ! <error opening '!AS' for input>
;  R0916  1          INDEX$_OPENOUT,      ! <error opening '!AS' for output>
;  R0917  1          INDEX$_TOOMANY,      ! <too many values supplied>
;  R0918  1          INDEX$_VALERR,       ! <specified value is out of legal range>
;  R0919  1          INDEX$_CANTBAL,      ! <can't balance last page>
```

```
;  R0920  1          INDEX$_CLOSEQUOT,   ! <missing close quote>
;  R0921  1          INDEX$_CONFQUAL,    ! <conflicting qualifiers>
;  R0922  1          INDEX$_CTRLCHAR,    ! <the following line contains control characters - ignored>
;  R0923  1          INDEX$_DOESNTFIT,   ! <'!AD' will not fit at the current indentation level>
;  R0924  1          INDEX$_DUPBEGIN,    ! <duplicate .XPLUS (BEGIN) - inserted as .XPLUS ()>
;  R0925  1          INDEX$_EMPTYIN,     ! <empty input file '!AS'>
;  R0926  1          INDEX$_IGNORED,     ! <'!AS' ignored>
;  R0927  1          INDEX$_INVINPUT,    ! <invalid input file format in file '!AS'>
;  R0928  1          INDEX$_INVRECORD,   ! <invalid record type in file '!AS'>
;  R0929  1          INDEX$_LASTCONT,    ! <can't generate continuation heading on last page>
;  R0930  1          INDEX$_NOBEGIN,     ! <.XPLUS (END) with no .XPLUS (BEGIN) - inserted as .XPLUS ()>
;  R0931  1          INDEX$_NOEND,       ! <.XPLUS (BEGIN) has no corresponding .XPLUS (END)>
;  R0932  1          INDEX$_NOINDEX,     ! <no index information in file '!AS'>
;  R0933  1          INDEX$_NOLIST,      ! <parameter list not allowed>
;  R0934  1          INDEX$_OVERSTRK,    ! <the following line contains an overstrike sequence>
;  R0935  1          INDEX$_SKIPPED,     ! <!UL reference!%S inside page range - ignored>
;  R0936  1          INDEX$_SYNTAX,      ! <error parsing '!AS'>
;  R0937  1          INDEX$_TEXFILE,     ! <error processing line !UL of TEX character file '!AS'>
;  R0938  1          INDEX$_TOODEEP,     ! <maximum subindex depth exceeded>
;  R0939  1          INDEX$_TOOFEW,      ! <not enough values supplied>
;  R0940  1          INDEX$_TRUNCATED,   ! <string too long - truncated>
;  R0941  1          INDEX$_COMPLETE,    ! <processing complete '!AS'>
;  R0942  1          INDEX$_CREATED,     ! <'!AS' created>
;  R0943  1          INDEX$_IDENT,       ! <INDEX version !AD>
;  R0944  1          INDEX$_PROCFILE,    ! <processing file '!AS'>
;  R0945  1          INDEX$_TEXT,        ! <!AS>
;  R0946  1          INDEX$_TEXTD,       ! <entry text: '!AD'>
;  R0947  1          INDEX$_TMS11;       ! <output file full - continuing with file '!AS'>
;  R0948  1
```

```
;  171          0949  1
;  172          0950  1 %FI
;  173          0951  1
;  174        L 0952  1 %IF %BLISS (BLISS36)
;  175        U 0953  1 %THEN
;  176        U 0954  1
;  177        U 0955  1 REQUIRE 'REQ:RNODEF';
;  178        U 0956  1
;  179          0957  1 %FI
;  180          0958  1
;  181          0959  1 SWITCHES LIST (NOREQUIRE);
;  182          0960  1
;  183          0961  1 !
;  184          0962  1 ! MACROS:
;  185          0963  1 !
;  186          0964  1 MACRO
;  187          0965  1     !
;  188          0966  1     ! Don't quote the character, and it doesn't count visually.
;  189          0967  1     !
;  190        M 0968  1     NQCHAR_NC (X) =
;  191        M 0969  1         BEGIN
;  192        M 0970  1         CH$WCHAR_A(X, WORD_POINTER);
;  193        M 0971  1         INT_WORD_LENGTH = .INT_WORD_LENGTH + 1;
;  194          0972  1         END     %,
;  195          0973  1
;  196          0974  1     !
;  197          0975  1     ! Don't quote the character, and it does count visually.
;  198          0976  1     !
;  199        M 0977  1     NQCHAR_C (X) =
;  200        M 0978  1         BEGIN
;  201        M 0979  1         NQCHAR_NC (X);
;  202        M 0980  1         EXT_WORD_LENGTH = .EXT_WORD_LENGTH + 1;
;  203        M 0981  1         TMS_WORD_LENGTH = .TMS_WORD_LENGTH + .CHRSIZ [X];
;  204          0982  1         END     %,
;  205          0983  1
;  206          0984  1     !
;  207          0985  1     ! Quote the character; it doesn't count visually.
;  208          0986  1     !
;  209        M 0987  1     QCHAR_NC (X) =
;  210        M 0988  1         BEGIN
;  211        M 0989  1         NQCHAR_NC (%C'_');
;  212        M 0990  1         NQCHAR_NC (X);
;  213          0991  1         END     %,
;  214          0992  1
;  215          0993  1     !
;  216          0994  1     ! Quote the character; it does count visually.
;  217          0995  1     !
;  218        M 0996  1     QCHAR_C (X) =
;  219        M 0997  1         BEGIN
;  220        M 0998  1         NQCHAR_NC (%C'_');
;  221        M 0999  1         NQCHAR_C (X);
;  222          1000  1         END     %,
;  223          1001  1
;  224          1002  1     !
;  225          1003  1     ! Special macro to keep track of the backspace control character
;  226          1004  1     !
;  227        M 1005  1     BACKSPACE (X) =
```

```
;  228        M 1006  1          BEGIN
;  229        M 1007  1          QCHAR_NC (X);
;  230        M 1008  1          EXT_WORD_LENGTH = .EXT_WORD_LENGTH - 1;
;  231          1009  1          END     %,
;  232          1010  1
;  233          1011  1      !
;  234          1012  1      ! Clear the text lines being built up.
;  235          1013  1      !
;  236        M 1014  1      CLR_RNO_LINE (_) =
;  237        M 1015  1          BEGIN
;  238        M 1016  1          LINE_POINTER = CH$PTR (RNO_LINE);
;  239        M 1017  1          INT_LINE_LENGTH = 0;
;  240        M 1018  1          EXT_LINE_LENGTH = 0;
;  241        M 1019  1          TMS_LINE_LENGTH = 0;
;  242          1020  1          END     %,
;  243          1021  1
;  244          1022  1      !
;  245          1023  1      ! Clear the word being built
;  246          1024  1      !
;  247        M 1025  1      CLR_RNO_WORD (_) =
;  248        M 1026  1          BEGIN
;  249        M 1027  1          WORD_POINTER = CH$PTR (RNO_WORD);
;  250        M 1028  1          INT_WORD_LENGTH = 0;
;  251        M 1029  1          EXT_WORD_LENGTH = 0;
;  252        M 1030  1          TMS_WORD_LENGTH = 0;
;  253        M 1031  1          HYPHENATE = FALSE;
;  254          1032  1          END     %,
;  255          1033  1
;  256          1034  1      !
;  257          1035  1      ! Write a line to the output file
;  258          1036  1      !
;  259        M 1037  1      PUT_LINE (S) =
;  260          1038  1          $XPO_PUT (IOB = OUTIOB, STRING = S) %;
;  261          1039  1
;  262          1040  1  !
;  263          1041  1  ! EQUATED SYMBOLS:
;  264          1042  1  !
;  265          1043  1  LITERAL
;  266          1044  1      TRUE = 1,
;  267          1045  1      FALSE = 0;
;  268          1046  1
;  269          1047  1  LITERAL
;  270          1048  1      TMS_SAFETY_MARGIN = 4;                      ! Subtracted from # chars / line to avoid wrapping
;  271          1049  1
;  272          1050  1  !
;  273          1051  1  ! OWN STORAGE:
;  274          1052  1  !
;  275          1053  1  OWN
;  276          1054  1      BLANKS : INITIAL (CH$PTR (UPLIT ('    '))); ! Pointer to blanks
;  277          1055  1
;  278          1056  1  OWN
;  279          1057  1      LINE_POINTER,
;  280          1058  1      RNO_LINE : VECTOR [CH$ALLOCATION (1200)],
;  281          1059  1      INT_LINE_LENGTH,
;  282          1060  1      EXT_LINE_LENGTH,
;  283          1061  1      TMS_LINE_LENGTH,
;  284          1062  1      LINE_TYPE,
```

```
;  285          1063  1      PAGE_WIDTH;
;  286          1064  1
;  287          1065  1  OWN
;  288          1066  1      DOING_LAST;                          ! TRUE if processing last line
;  289          1067  1
;  290          1068  1  OWN
;  291          1069  1      SAVE_ENTRY : INITIAL (FALSE);        ! TRUE if saw guide head at top of left page
;  292          1070  1
;  293          1071  1  !
;  294          1072  1  ! EXTERNAL REFERENCES:
;  295          1073  1  !
;  296          1074  1
;  297          1075  1  EXTERNAL LITERAL
;  298          1076  1      TAB    : UNSIGNED (8),
;  299          1077  1      MAXLST,
;  300          1078  1      TMSSTD,                              ! Average TMS character size
;  301          1079  1      MSPACE,                              ! TMS 'em' space size
;  302          1080  1      RINTES : UNSIGNED (8);
;  303          1081  1
;  304          1082  1  EXTERNAL
;  305          1083  1      CMDBLK : $NDXCMD,                    ! Command line information block
;  306          1084  1      NDXVRL,                              ! Length of version string
;  307          1085  1      NDXVRP,                              ! CH$PTR to version string
;  308          1086  1      OUTIOB : $XPO_IOB (),                ! Output file IOB
;  309          1087  1      LSTPTR : REF $XE_BLOCK,
;  310          1088  1      INDLVL,                              ! Index level
;  311          1089  1      LSTSTK : VECTOR,                     ! Temporary stack for storing back links
;  312          1090  1      BUCKET : $BUCKET_ARRAY [27, 27],     ! Hashing buckets
;  313          1091  1      CHRSIZ : REF VECTOR,                 ! TMS11 character size vector
;  314          1092  1      ALLOWD,                              ! Usuable lines per page
;  315          1093  1      LCOUNT,                              ! Number of lines in left column
;  316          1094  1      RCOUNT,                              ! Number of lines in right column
;  317          1095  1      TCOUNT,                              ! Number of lines in temp column
;  318          1096  1  !
;  319          1097  1  ! NOTE: The vectors and blockvectors below have two extra entries allocated
;  320          1098  1  !       to avoid needing to subtract 1 all the time (for entry zero),
;  321          1099  1  !       and so that there will always be an available line at the end of the column
;  322          1100  1  !
;  323          1101  1      LTYPE  : VECTOR,                     ! Left column line types
;  324          1102  1      LLINES : BLOCKVECTOR [, STR$K_D_BLN], ! Left column string descriptors
;  325          1103  1      RTYPE  : VECTOR,                     ! Right column line types
;  326          1104  1      RLINES : BLOCKVECTOR [, STR$K_D_BLN], ! Right column string descriptors
;  327          1105  1      TTYPE  : VECTOR,                     ! Temp column line types
;  328          1106  1      TLINES : BLOCKVECTOR [, STR$K_D_BLN]; ! Right column for last page
;  329          1107  1
;  330          1108  1  EXTERNAL ROUTINE
;  331          1109  1
;  332        L 1110  1  %IF %BLISS (BLISS32)
;  333          1111  1  %THEN
;  334          1112  1
;  335          1113  1      OPEN_ERROR,                          ! File open error processor
;  336          1114  1
;  337          1115  1  %FI
;  338          1116  1
;  339          1117  1      ENTMSG   : NOVALUE,
;  340          1118  1      GENTRY,
;  341          1119  1      LASTPG   : NOVALUE,
```

```
;  342          1120  1      PACBAS,
;  343          1121  1      PACPAG,
;  344          1122  1      PAGEQL,
;  345          1123  1      PAGMRG,
;  346          1124  1      PUTPAG    : NOVALUE,
;  347          1125  1      TMSINI    : NOVALUE,
;  348          1126  1      TMSPUT    : NOVALUE,
;  349          1127  1      XTNPAG;
```

```
;  351          1128  1 %SBTTL 'NDXFMT -- Build output index from internal index'
;  352          1129  1 GLOBAL ROUTINE NDXFMT : NOVALUE =
;  353          1130  1
;  354          1131  1 !++
;  355          1132  1 ! FUNCTIONAL DESCRIPTION:
;  356          1133  1 !
;  357          1134  1 !       This routine is the top-level control routine of the formatting
;  358          1135  1 !       procedure.
;  359          1136  1 !
;  360          1137  1 !       If output is being generated for RUNOFF, a "standard" RUNOFF
;  361          1138  1 !       environment is set up.
;  362          1139  1 !
;  363          1140  1 !       If a require file was specified, it is copied to the output file.
;  364          1141  1 !
;  365          1142  1 !       Each bucket of the binary index is then examined to form the
;  366          1143  1 !       output index. The nonalpha bucket is examined first if either
;  367          1144  1 !       /SORT=NONALPHA=BEFORE or /SORT=NONALPHA=IGNORE were specified
;  368          1145  1 !       on the command line. Otherwise, the nonalpha bucket is examined last.
;  369          1146  1 !
;  370          1147  1 ! FORMAL PARAMETERS:
;  371          1148  1 !
;  372          1149  1 !       None
;  373          1150  1 !
;  374          1151  1 ! IMPLICIT INPUTS:
;  375          1152  1 !
;  376          1153  1 !       CMDBLK          - Command line information block
;  377          1154  1 !
;  378          1155  1 ! IMPLICIT OUTPUTS:
;  379          1156  1 !
;  380          1157  1 !       INDLVL          - subindex level
;  381          1158  1 !       PAGE_WIDTH      - width of output page
;  382          1159  1 !       ALLOWD          - number of lines allowed on first page of index
;  383          1160  1 !
;  384          1161  1 ! ROUTINE VALUE:
;  385          1162  1 ! COMPLETION CODES:
;  386          1163  1 !
;  387          1164  1 !       None
;  388          1165  1 !
;  389          1166  1 ! SIDE EFFECTS:
;  390          1167  1 !
;  391          1168  1 !       None
;  392          1169  1 !
;  393          1170  1 !--
;  394          1171  1
;  395          1172  2     BEGIN
;  396          1173  2     LOCAL
;  397          1174  2         DSC_PTR : REF $STR_DESCRIPTOR (),
;  398          1175  2         CMD_LEN,
;  399          1176  2         CMD_PTR,
;  400          1177  2         PTR;
;  401          1178  2
;  402          1179  2     !
;  403          1180  2     ! Initialization
;  404          1181  2     !
;  405          1182  2     LCOUNT = 0;
;  406          1183  2     RCOUNT = 0;
;  407          1184  2     DOING_LAST = FALSE;
```

```
;  408          1185  2
;  409          1186  2      SELECTONE .CMDBLK [NDX$H_LAYOUT] OF
;  410          1187  2      SET
;  411          1188  2
;  412          1189  2      [SEPARATE]:
;  413          1190  2          PAGE_WIDTH = .CMDBLK [NDX$G_COLUMN_WID] + .CMDBLK [NDX$G_GUTTER_WID] + .CMDBLK [NDX$G_SEPARATE_WIDTH
;  414          1191  2
;  415          1192  2      [TWO_COLUMN]:
;  416          1193  2          PAGE_WIDTH = (2 * .CMDBLK [NDX$G_COLUMN_WID]) + .CMDBLK [NDX$G_GUTTER_WID];
;  417          1194  2
;  418          1195  2      [OTHERWISE]:
;  419          1196  2          PAGE_WIDTH = .CMDBLK [NDX$G_COLUMN_WID];
;  420          1197  2
;  421          1198  2      TES;
;  422          1199  2
;  423          1200  2      !
;  424          1201  2      ! Reformat command line before outputting.
;  425          1202  2      !   Change '<' to '['
;  426          1203  2      !   Change '>' to ']'
;  427          1204  2      !   Change ';' to '.'
;  428          1205  2      !
;  429          1206  2      DSC_PTR = CMDBLK [NDX$T_COMMAND_LINE];
;  430          1207  2      CMD_LEN = .DSC_PTR [STR$H_LENGTH];
;  431          1208  2      CMD_PTR = .DSC_PTR [STR$A_POINTER];
;  432          1209  2
;  433          1210  2      WHILE NOT CH$FAIL (PTR = CH$FIND_CH (.CMD_LEN, .CMD_PTR, %C'<')) DO
;  434          1211  2          CH$WCHAR (%C'[', .PTR);
;  435          1212  2
;  436          1213  2      WHILE NOT CH$FAIL (PTR = CH$FIND_CH (.CMD_LEN, .CMD_PTR, %C'>')) DO
;  437          1214  2          CH$WCHAR (%C']', .PTR);
;  438          1215  2
;  439          1216  2      WHILE NOT CH$FAIL (PTR = CH$FIND_CH (.CMD_LEN, .CMD_PTR, %C';')) DO
;  440          1217  2          CH$WCHAR (%C'.', .PTR);
;  441          1218  2
;  442          1219  2      !
;  443          1220  2      ! Compute number of lines allowed on first page
;  444          1221  2      !
;  445          1222  2      IF .CMDBLK [NDX$G_RESERVE_LINES] NEQ 0
;  446          1223  2      THEN
;  447          1224  2          !
;  448          1225  2          ! Number of lines on first page is lines per page
;  449          1226  2          ! minus number of reserved lines.
;  450          1227  2          !
;  451          1228  2          ALLOWD = .CMDBLK [NDX$G_LINES_PAGE] - .CMDBLK [NDX$G_RESERVE_LINES]
;  452          1229  2      ELSE
;  453          1230  3          BEGIN
;  454          1231  3
;  455          1232  3          IF .CMDBLK [NDX$H_FORMAT] EQL DSR
;  456          1233  3          THEN
;  457          1234  3              ALLOWD = .CMDBLK [NDX$G_LINES_PAGE] - 4      ! Default for RUNOFF
;  458          1235  3          ELSE
;  459          1236  3              ALLOWD = .CMDBLK [NDX$G_LINES_PAGE] - 22;    ! Default for TMS and TEX
;  460          1237  3
;  461          1238  2          END;
;  462          1239  2
;  463          1240  2      !
;  464          1241  2      ! Write output dependant prologue
```

```
;  465          1242  2     !
;  466          1243  2     SELECTONE .CMDBLK [NDX$H_FORMAT] OF
;  467          1244  2         SET
;  468          1245  2
;  469          1246  2         [DSR]:
;  470          1247  3             BEGIN
;  471          1248  3             !
;  472          1249  3             ! RUNOFF output
;  473          1250  3             !
;  474          1251  3             ! Generate a "standard" RUNOFF environment.
;  475          1252  3             !
;  476        L 1253  3 %IF DSRPLUS
;  477        U 1254  3 %THEN
;  478        U 1255  3
;  479        U 1256  3             PUT_LINE ($STR_CONCAT ('.! INDEX version ', (.NDXVRL, .NDXVRP)));
;  480        U 1257  3
;  481          1258  3 %ELSE
;  482          1259  3
;  483          1260  3             PUT_LINE ($STR_CONCAT ('.! DSRINDEX version ', (.NDXVRL, .NDXVRP)));
;  484          1261  3
;  485          1262  3 %FI
;  486          1263  3
;  487          1264  3             PUT_LINE ($STR_CONCAT ('.! ', .DSC_PTR));
;  488          1265  3             PUT_LINE ('.SAVE');
;  489          1266  3             PUT_LINE ('.NO FLAGS ALL');
;  490          1267  3             PUT_LINE ('.NO FLAGS BREAK          .NO FLAGS CAPITALIZE    .NO FLAGS ENDFOOTNOTE');
;  491          1268  3             PUT_LINE ('.NO FLAGS HYPHENATE      .NO FLAGS INDEX         .NO FLAGS LOWERCASE');
;  492          1269  3             PUT_LINE ('.NO FLAGS PERIOD .NO FLAGS SPACE         .NO FLAGS SUBSTITUTE');
;  493          1270  3             PUT_LINE ('.NO FLAGS UPPERCASE');
;  494          1271  3             PUT_LINE ('.FLAGS ACCEPT            .FLAGS BOLD *           .FLAGS COMMENT !');
;  495          1272  3             PUT_LINE ('.FLAGS OVERSTRIKE %      .FLAGS UNDERLINE &');
;  496          1273  3             PUT_LINE ('.FLAGS ALL');
;  497          1274  3             PUT_LINE ('.NO FILL         .NO JUSTIFY');
;  498          1275  3
;  499        P 1276  3             PUT_LINE (
;  500        P 1277  3                 $STR_CONCAT (
;  501        P 1278  3                     '.LEFT MARGIN 0             .RIGHT MARGIN ',
;  502        P 1279  3                     $STR_ASCII (.PAGE_WIDTH),
;  503        P 1280  3                     '    .PAGE SIZE , ',
;  504        P 1281  3                     $STR_ASCII (.PAGE_WIDTH)
;  505        P 1282  3                 )
;  506          1283  3             );
;  507          1284  3
;  508          1285  3             PUT_LINE ('.NUMBER INDEX            .NUMBER PAGE 1          .FIRST TITLE');
;  509          1286  3
;  510          1287  3             IF .CMDBLK [NDX$V_TELLTALE]
;  511          1288  3             THEN
;  512          1289  4                 BEGIN
;  513          1290  4                 !
;  514          1291  4                 ! Generate prologue to set up for /TELLTALE
;  515          1292  4                 !
;  516          1293  4                 PUT_LINE ('.LAYOUT 2,2');
;  517          1294  4                 PUT_LINE ('.TITLE');
;  518          1295  4                 PUT_LINE ('.SUBTITLE');
;  519          1296  3                 END;
;  520          1297  3
;  521          1298  3             IF .CMDBLK [NDX$H_LAYOUT] EQL GALLEY THEN PUT_LINE ('.NO PAGING');
```

```
;   522          1299  2              END;
;   523          1300  2
;   524          1301  2          [TMS11_A, TMS11_E]:
;   525          1302  2              !
;   526          1303  2              ! Generate TMS11 heading
;   527          1304  2              !
;   528          1305  2              TMSINI ();
;   529          1306  2
;   530          1307  2          [TEX]:
;   531          1308  3              BEGIN
;   532          1309  3              !
;   533          1310  3              ! Generate a heading for TEX
;   534          1311  3              !
;   535          1312  3              PUT_LINE ($STR_CONCAT ('% INDEX version ', (.NDXVRL, .NDXVRP)));
;   536          1313  3              PUT_LINE ($STR_CONCAT ('% ', .DSC_PTR));
;   537          1314  3              PUT_LINE ('\twocolindex');
;   538          1315  2              END;
;   539          1316  2
;   540          1317  2          TES;
;   541          1318  2
;   542          1319  2      DSC_PTR = CMDBLK [NDX$T_REQUIRE_FILE];
;   543          1320  2
;   544          1321  3      IF .CMDBLK [NDX$V_REQUIRE] AND (.DSC_PTR [STR$H_LENGTH] GTR 0)
;   545          1322  2      THEN
;   546          1323  3          BEGIN
;   547          1324  3          !
;   548          1325  3          ! User specified a require file. Insert it here.
;   549          1326  3          !
;   550          1327  3          LOCAL
;   551          1328  3              REQUIRE_IOB : $XPO_IOB ();
;   552          1329  3
;   553          1330  3          $XPO_IOB_INIT (IOB = REQUIRE_IOB);
;   554        P 1331  3          $XPO_OPEN (IOB = REQUIRE_IOB, FILE_SPEC = .DSC_PTR
;   555        P 1332  3              %IF %BLISS (BLISS32) %THEN , FAILURE = OPEN_ERROR %FI
;   556          1333  3              );
;   557          1334  3
;   558          1335  3          WHILE $XPO_GET (IOB = REQUIRE_IOB) EQL XPO$_NORMAL DO
;   559          1336  4              BEGIN
;   560          1337  4
;   561          1338  4              SELECTONE .CMDBLK [NDX$H_FORMAT] OF
;   562          1339  4                  SET
;   563          1340  4
;   564          1341  4                  [TMS11_A, TMS11_E]:
;   565          1342  4                      TMSPUT (.REQUIRE_IOB [IOB$H_STRING], .REQUIRE_IOB [IOB$A_STRING], OUTIOB, FALSE);
;   566          1343  4
;   567          1344  4                  [DSR, TEX]:
;   568          1345  4                      PUT_LINE (REQUIRE_IOB [IOB$T_STRING]);
;   569          1346  4
;   570          1347  4                  TES;
;   571          1348  4
;   572          1349  3              END;
;   573          1350  3
;   574          1351  3          $XPO_CLOSE (IOB = REQUIRE_IOB);
;   575          1352  3          END
;   576          1353  2      ELSE
;   577          1354  2          IF .CMDBLK [NDX$H_FORMAT] EQL DSR
;   578          1355  2          THEN
```

```
;   579          1356  3              BEGIN
;   580          1357  3              !
;   581          1358  3              ! User didn't specify a require file and output is for RUNOFF.
;   582          1359  3              ! Therefore, insert default commands.
;   583          1360  3              !
;   584          1361  3              PUT_LINE ('.CENTER;INDEX');
;   585          1362  3              PUT_LINE ('.BLANK3');
;   586          1363  2              END;
;   587          1364  2
;   588          1365  2          IF .CMDBLK [NDX$H_NONALPHA] NEQ AFTER
;   589          1366  2          THEN
;   590          1367  2              !
;   591          1368  2              ! Examine nonalpha bucket first if user specified
;   592          1369  2              !
;   593          1370  2              ! /SORT=NONALPHA=BEFORE or /SORT=NONALPHA=IGNORE
;   594          1371  2              !
;   595          1372  2              EXAM_BUCKET (0);
;   596          1373  2
;   597          1374  2          !
;   598          1375  2          ! Examine all alpha buckets
;   599          1376  2          !
;   600          1377  2          INCR I FROM 1 TO 26 DO
;   601          1378  2              EXAM_BUCKET (.I);
;   602          1379  2
;   603          1380  2          IF .CMDBLK [NDX$H_NONALPHA] EQL AFTER
;   604          1381  2          THEN
;   605          1382  2              !
;   606          1383  2              ! Examine nonalpha characters last
;   607          1384  2              !
;   608          1385  2              EXAM_BUCKET (0);
;   609          1386  2
;   610          1387  2          !
;   611          1388  2          ! Write last page
;   612          1389  2          !
;   613          1390  2          LASTPG ();
;   614          1391  2
;   615          1392  1          END;                            !End of NDXFMT


                                                                                .TITLE  NDXFMT NDXFMT -- Binary index formatter
                                                                                .IDENT  \V04-000\

                                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                        20  20  20  20  00000 P.AAA:  .ASCII  \    \
72  65  76  20  58  45  44  4E  49  52  53  44  20  21  2E  00004 P.AAC:  .ASCII  \.! DSRINDEX version \
                                            20  6E  6F  69  73  00013
                                                        20  21  2E  00018 P.AAE:  .ASCII  \.! \
                                            45  56  41  53  2E  0001B P.AAH:  .ASCII  \.SAVE\
        4C  4C  41  20  53  47  41  4C  46  20  4F  4E  2E  00020 P.AAI:  .ASCII  \.NO FLAGS ALL\
4B  41  45  52  42  20  53  47  41  4C  46  20  4F  4E  2E  0002D P.AAJ:  .ASCII  \.NO FLAGS BREAK\<9><9>\.NO FLAGS CA\
    41  43  20  53  47  41  4C  46  20  4F  4E  2E  09  09  0003C
4C  46  20  4F  4E  2E  09  45  5A  49  4C  41  54  49  50  0004A           .ASCII  \PITALIZE\<9>\.NO FLAGS ENDFOOTNOTE\
45  54  4F  4E  54  4F  4F  46  44  4E  45  20  53  47  41  00059
45  48  50  59  48  20  53  47  41  4C  46  20  4F  4E  2E  00068 P.AAK:  .ASCII  \.NO FLAGS HYPHENATE\<9>\.NO FLAGS INDE\
20  53  47  41  4C  46  20  4F  4E  2E  09  45  54  41  4E  00077
                                            45  44  4E  49  00086
```

```
4F 4C 20 53 47 41 4C 46 20 4F 4E 2E 09 09 58 0008A                .ASCII  \X\<9><9>\.NO FLAGS LOWERCASE\
                        45 53 41 43 52 45 57 00099
4F 49 52 45 50 20 53 47 41 4C 46 20 4F 4E 2E 000A0 P.AAL:  .ASCII  \.NO FLAGS PERIOD\<9>\.NO FLAGS SPACE\<9>
41 50 53 20 53 47 41 4C 46 20 4F 4E 2E 09 44 000AF
                                       09 45 43 000BE
53 42 55 53 20 53 47 41 4C 46 20 4F 4E 2E 09 000C1                .ASCII  <9>\.NO FLAGS SUBSTITUTE\
                        45 54 55 54 49 54 000D0
52 45 50 50 55 20 53 47 41 4C 46 20 4F 4E 2E 000D6 P.AAM:  .ASCII  \.NO FLAGS UPPERCASE\
                                    45 53 41 43 000E5
5F 20 54 50 45 43 43 41 20 53 47 41 4C 46 2E 000E9 P.AAN:  .ASCII  \.FLAGS ACCEPT _\<9><9>\.FLAGS BOLD \
   20 44 4C 4F 42 20 53 47 41 4C 46 2E 09 09 000F8
45 4D 4D 4F 43 20 53 47 41 4C 46 2E 09 09 2A 00106                .ASCII  \*\<9><9>\.FLAGS COMMENT !\
                                    21 20 54 4E 00115
49 52 54 53 52 45 56 4F 20 53 47 41 4C 46 2E 00119 P.AAO:  .ASCII  \.FLAGS OVERSTRIKE %\<9>\.FLAGS UNDERLI\
44 4E 55 20 53 47 41 4C 46 2E 09 25 20 45 4B 00128
                                    49 4C 52 45 00137
                                    26 20 45 4E 0013B                .ASCII  \NE &\
                  4C 4C 41 20 53 47 41 4C 46 2E 0013F P.AAP:  .ASCII  \.FLAGS ALL\
4A 20 4F 4E 2E 09 09 4C 4C 49 46 20 4F 4E 2E 00149 P.AAQ:  .ASCII  \.NO FILL\<9><9>\.NO JUSTIFY\
                        59 46 49 54 53 55 00158
09 30 20 4E 49 47 52 41 4D 20 54 46 45 4C 2E 0015E P.AAT:  .ASCII  \.LEFT MARGIN 0\<9><9>\.RIGHT MARGIN\
   4E 49 47 52 41 4D 20 54 48 47 49 52 2E 09 0016D
                                       20 0017B                .ASCII  \ \
   20 2C 20 45 5A 49 53 20 45 47 41 50 2E 09 0017C P.AAU:  .ASCII  <9>\.PAGE SIZE , \
09 09 58 45 44 4E 49 20 52 45 42 4D 55 4E 2E 0018A P.AAX:  .ASCII  \.NUMBER INDEX\<9><9>\.NUMBER PAGE 1\
   31 20 45 47 41 50 20 52 45 42 4D 55 4E 2E 00199
   45 4C 54 49 54 20 54 53 52 49 46 2E 09 09 001A7                .ASCII  <9><9>\.FIRST TITLE\
            32 2C 32 20 54 55 4F 59 41 4C 2E 001B5 P.AAY:  .ASCII  \.LAYOUT 2,2\
                        45 4C 54 49 54 2E 001C0 P.AAZ:  .ASCII  \.TITLE\
               45 4C 54 49 54 42 55 53 2E 001C6 P.ABA:  .ASCII  \.SUBTITLE\
            47 4E 49 47 41 50 20 4F 4E 2E 001CF P.ABB:  .ASCII  \.NO PAGING\
6E 6F 69 73 72 65 76 20 58 45 44 4E 49 20 25 001D9 P.ABD:  .ASCII  \% INDEX version \
                                       20 001E8
                                    20 25 001E9 P.ABG:  .ASCII  \% \
   78 65 64 6E 69 6C 6F 63 6F 77 74 5C 001EB P.ABI:  .ASCII  <92>\twocolindex\
58 45 44 4E 49 3B 52 45 54 4E 45 43 2E 001F7 P.ABJ:  .ASCII  \.CENTER;INDEX\
                  33 4B 4E 41 4C 42 2E 00204 P.ABK:  .ASCII  \.BLANK3\

                                             .PSECT  $OWN$,NOEXE,2

                                 00000000' 00000 BLANKS:  .ADDRESS P.AAA
                                           00004 LINE_POINTER:
                                                 .BLKB   4
                                           00008 RNO_LINE:
                                                 .BLKB   1200
                                           004B8 INT_LINE_LENGTH:
                                                 .BLKB   4
                                           004BC EXT_LINE_LENGTH:
                                                 .BLKB   4
                                           004C0 TMS_LINE_LENGTH:
                                                 .BLKB   4
                                           004C4 LINE_TYPE:
                                                 .BLKB   4
                                           004C8 PAGE_WIDTH:
                                                 .BLKB   4
                                           004CC DOING_LAST:
                                                 .BLKB   4
```

```
                                00000000  004D0 SAVE_ENTRY:
                                                      .LONG   0                       ;
                                    0014  004D4 $STR$STRING0:
                                                      .WORD   20                      ;
                                  01  0E  004D6          .BYTE   14, 1               ;
                                00000000' 004D8          .ADDRESS P.AAC              ;
                                    0003  004DC $STR$STRING0:
                                                      .WORD   3                       ;
                                  01  0E  004DE          .BYTE   14, 1               ;
                                00000000' 004E0          .ADDRESS P.AAF              ;
                                    0005  004E4 $IOB$OUTPUT:
                                                      .WORD   5                       ;
                                  01  0E  004E6          .BYTE   14, 1               ;
                                00000000' 004E8          .ADDRESS P.AAH              ;
                                    000D  004EC $IOB$OUTPUT:
                                                      .WORD   13                      ;
                                  01  0E  004EE          .BYTE   14, 1               ;
                                00000000' 004F0          .ADDRESS P.AAI              ;
                                    003B  004F4 $IOB$OUTPUT:
                                                      .WORD   59                      ;
                                  01  0E  004F6          .BYTE   14, 1               ;
                                00000000' 004F8          .ADDRESS P.AAJ              ;
                                    0038  004FC $IOB$OUTPUT:
                                                      .WORD   56                      ;
                                  01  0E  004FE          .BYTE   14, 1               ;
                                00000000' 00500          .ADDRESS P.AAK              ;
                                    0036  00504  IOB$OUTPUT:
                                                      .WORD   54                      ;
                                  01  0E  00506          .BYTE   14, 1               ;
                                00000000' 00508          .ADDRESS P.AAL              ;
                                    0013  0050C $IOB$OUTPUT:
                                                      .WORD   19                      ;
                                  01  0E  0050E          .BYTE   14, 1               ;
                                00000000' 00510          .ADDRESS P.AAM              ;
                                    0030  00514 $IOB$CUTPUT:
                                                      .WORD   48                      ;
                                  01  0E  00516          .BYTE   14, 1               ;
                                00000000' 00518          .ADDRESS P.AAN              ;
                                    0026  0051C $IOB$OUTPUT:
                                                      .WORD   38                      ;
                                  01  0E  0051E          .BYTE   14, 1               ;
                                00000000' 00520          .ADDRESS P.AAO              ;
                                    000A  00524 $IOB$OUTPUT:
                                                      .WORD   10                      ;
                                  01  0E  00526          .BYTE   14, 1               ;
                                00000000' 00528          .ADDRESS P.AAP              ;
                                    0015  0052C $IOB$OUTPUT:
                                                      .WORD   21                      ;
                                  01  0E  0052E          .BYTE   14, 1               ;
                                00000000' 00530          .ADDRESS P.AAQ              ;
                                    001E  00534 $STR$STRING0:
                                                      .WORD   30                      ;
                                  01  0E  00536          .BYTE   14, 1               ;
                                00000000' 00538          .ADDRESS P.AAT              ;
                                    000E  0053C $STR$STRING2:
                                                      .WORD   14                      ;
                                  01  0E  0053E          .BYTE   14, 1               ;
```

```
                    00000000' 00540          .ADDRESS P.AAU                          ;
                        002B  00544 $IOB$OUTPUT:
                                             .WORD   43                              ;
                      01  0E  00546          .BYTE   14, 1                           ;
                    00000000' 00548          .ADDRESS P.AAX                          ;
                        000B  0054C $IOB$OUTPUT:
                                             .WORD   11                              ;
                      01  0E  0054E          .BYTE   14, 1                           ;
                    00000000' 00550          .ADDRESS P.AAY                          ;
                        0006  00554 $IOB$OUTPUT:
                                             .WORD   6                               ;
                      01  0E  00556          .BYTE   14, 1                           ;
                    00000000' 00558          .ADDRESS P.AAZ                          ;
                        0009  0055C $IOB$OUTPUT:
                                             .WORD   9                               ;
                      01  0E  0055E          .BYTE   14, 1                           ;
                    00000000' 00560          .ADDRESS P.ABA                          ;
                        000A  00564 $IOB$OUTPUT:
                                             .WORD   10                              ;
                      01  0E  00566          .BYTE   14, 1                           ;
                    00000000' 00568          .ADDRESS P.ABB                          ;
                        0010  0056C $STR$STRING0:
                                             .WORD   16                              ;
                      01  0E  0056E          .BYTE   14, 1                           ;
                    00000000' 00570          .ADDRESS P.ABD                          ;
                        0002  00574 $STR$STRING0:
                                             .WORD   2                               ;
                      01  0E  00576          .BYTE   14, 1                           ;
                    00000000' 00578          .ADDRESS P.ABG                          ;
                        000C  0057C $IOB$OUTPUT:
                                             .WORD   12                              ;
                      01  0E  0057E          .BYTE   14, 1                           ;
                    00000000' 00580          .ADDRESS P.ABI                          ;
                        000D  00584 $IOB$OUTPUT:
                                             .WORD   13                              ;
                      01  0E  00586          .BYTE   14, 1                           ;
                    00000000' 00588          .ADDRESS P.ABJ                          ;
                        0007  0058C $IOB$OUTPUT:
                                             .WORD   7                               ;
                      01  0E  0058E          .BYTE   14, 1                           ;
                    00000000' 00590          .ADDRESS P.ABK                          ;

                                             .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$_TMS11
                                                        .EXTRN  TAB, MAXLST, TMSSTD
                                                        .EXTRN  MSPACE, RINTES, CMDBLK
                                                        .EXTRN  NDXVRL, NDXVRP, OUTIOB
                                                        .EXTRN  LSTPTR, INDLVL, LSTSTK
                                                        .EXTRN  BUCKET, CHRSIZ, ALLOWD
                                                        .EXTRN  LCOUNT, RCOUNT, TCOUNT
                                                        .EXTRN  LTYPE, LLINES, RTYPE
                                                        .EXTRN  RLINES, TTYPE, TLINES
                                                        .EXTRN  OPEN_ERROR, ENTMSG
                                                        .EXTRN  GENTRY, LASTPG, PACBAS
                                                        .EXTRN  PACPAG, PAGEQL, PAGMRG
                                                        .EXTRN  PUTPAG, TMSINI, TMSPUT
                                                        .EXTRN  XTNPAG, XST$JOIN
                                                        .EXTRN  XPO$PUT, XPO$FAILURE
                                                        .EXTRN  XST$ASCII, XPO$OPEN
                                                        .EXTRN  XPO$GET, XPO$CLOSE

                                                        .PSECT  $CODE$,NOWRT,2

                                    0FFC 00000          .ENTRY  NDXFMT, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-      ; 1129
                                                                R11
                                5B 00000000G   EF   9E 00002    MOVAB   CMDBLK+4, 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   PAGE_WIDTH, R8                         ;
                                57 00000000G   EF   9E 0001E    MOVAB   IOB$T68, R7                            ;
                                5E     FF0C    CE   9E 00025    MOVAB   -244(SP), SP                           ;
                                   00000000G   EF   D4 0002A    CLRL    LCOUNT                                 ; 1182
                                   00000000G   EF   D4 00030    CLRL    RCOUNT                                 ; 1183
                                         04    A8   D4 00036    CLRL    DOING_LAST                             ; 1184
                                50       02    AB   32 00039    CVTWL   CMDBLK+6, R0                           ; 1186
                                03             50   B1 0003D    CMPW    R0, #3                                 ; 1189
                                               0D   12 00040    BNEQ    1$                                     ;
                50      08      AB       0C    AB   C1 00042    ADDL3   CMDBLK+16, CMDBLK+12, R0                ; 1190
                                68       18 BB40   9E 00048    MOVAB   @CMDBLK+28[R0], PAGE_WIDTH             ;
```

```
                                    14  11 0004D                BRB     3$
                              01    50  B1 0004F 1$:            CMPW    R0, #1                                  ;1192
                                    0B  12 00052                BNEQ    2$
                              50    08  AB  D0 00054            MOVL    CMDBLK+12, R0                           ;1193
                              68    0C BB40 3E 00058            MOVAW   @CMDBLK+16[R0], PAGE_WIDTH
                                    04  11 0005D                BRB     3$
                              68    08  AB  D0 0005F 2$:        MOVL    CMDBLK+12, PAGE_WIDTH                   ;1196
                              56    44  AB  9E 00063 3$:        MOVAB   CMDBLK+72, DSC_PTR                      ;1206
                              53        66  3C 00067            MOVZWL  (DSC_PTR), CMD_LEN                      ;1207
                              52    04  A6  D0 0006A            MOVL    4(DSC_PTR), CMD_PTR                     ;1208
                    62        53        3C  3A 0006E 4$:        LOCC    #60, CMD_LEN, (CMD_PTR)                 ;1210
                                    02  12 00072                BNEQ    5$
                                    51  D4 00074                CLRL    R1
                                    51  D5 00076 5$:            TSTL    PTR
                                    06  13 00078                BEQL    6$
                              61    5B  8F  90 0007A            MOVB    #91, (PTR)                              ;1211
                                    EE  11 0007E                BRB     4$
                    62        53        3E  3A 00080 6$:        LOCC    #62, CMD_LEN, (CMD_PTR)                 ;1213
                                    02  12 00084                BNEQ    7$
                                    51  D4 00086                CLRL    R1
                                    51  D5 00088 7$:            TSTL    PTR
                                    06  13 0008A                BEQL    8$
                              61    5D  8F  90 0008C            MOVB    #93, (PTR)                              ;1214
                                    EE  11 00090                BRB     6$
                    62        53        3B  3A 00092 8$:        LOCC    #59, CMD_LEN, (CMD_PTR)                 ;1216
                                    02  12 00096                BNEQ    9$
                                    51  D4 00098                CLRL    R1
                                    51  D5 0009A 9$:            TSTL    PTR
                                    05  13 0009C                BEQL    10$
                              61        2E  90 0009E            MOVB    #46, (PTR)                              ;1217
                                    EF  11 000A1                BRB     8$
                              51    10  AB  D0 000A3 10$:       MOVL    CMDBLK+20, R1                           ;1228
                              50    14  AB  D0 000A7            MOVL    CMDBLK+24, R0                           ;1222
                                    0A  13 000AB                BEQL    11$
00000000G  EF                 51        50  C3 000AD            SUBL3   R0, R1, ALLOWD                          ;1228
                                    17  11 000B5                BRB     13$
                              01        6B  B1 000B7 11$:       CMPW    CMDBLK+4, #1                            ;1232
                                    0A  12 000BA                BNEQ    12$
                  00000000G   EF    FC  A1  9E 000BC            MOVAB   -4(R1), ALLOWD                          ;1234
                                    08  11 000C4                BRB     13$
                  00000000G   EF    EA  A1  9E 000C6 12$:       MOVAB   -22(R1), ALLOWD                         ;1236
                              50        6B  32 000CE 13$:       CVTWL   CMDBLK+4, R0                            ;1243
                              01        50  B1 000D1            CMPW    R0, #1                                  ;1246
                                    03  13 000D4                BEQL    14$
                                  01A9  31 000D6                BRW     16$
                        F8    AD 00000000G  EF  B0 000D9 14$:   MOVW    NDXVRL, $STR$STRING1                    ;1260
                        FA    AD            0E  90 000E1        MOVB    #14, $STR$STRING1+2
                        FB    AD            01  90 000E5        MOVB    #1, $STR$STRING1+3
                        FC    AD 00000000G  EF  D0 000E9        MOVL    NDXVRP, $STR$STRING1+4
                                    F8    AD  9F 000F1        PUSHAB  $STR$STRING1
                                    0C    A8  9F 000F4        PUSHAB  $STR$STRING0
                  00000000G   EF            02  FB 000F7        CALLS   #2, XST$JOIN
                              67            50  D0 000FE        MOVL    R0, IOB$+68
                        E8    A7            07  90 00101        MOVB    #7, IOB$+44
                                            59  DD 00105        PUSHL   R9
                                            7E  D4 00107        CLRL    -(SP)
                                    BC    A7  9F 00109        PUSHAB  IOB$
```

```
                         6A              03  FB 0010C       CALLS   #3, XPO$PUT
                                         56  DD 0010F       PUSHL   DSC_PTR                                  ;1264
                                 14      A8  9F 00111       PUSHAB  $STR$STRINGO
                 00000000G       EF      02  FB 00114       CALLS   #2, XST$JOIN
                                 67      50  D0 0011B       MOVL    R0, IOB$+68
                        E8       A7      07  90 0011E       MOVB    #7, IOB$+44
                                         59  DD 00122       PUSHL   R9
                                         7E  D4 00124       CLRL    -(SP)
                                 BC      A7  9F 00126       PUSHAB  IOB$
                                 6A      03  FB 00129       CALLS   #3, XPO$PUT
                                 67      1C  A8  9E 0012C   MOVAB   $IOB$OUTPUT, IOB$+68                      ;1265
                        E8       A7      07  90 00130       MOVB    #7, IOB$+44
                                         59  DD 00134       PUSHL   R9
                                         7E  D4 00136       CLRL    -(SP)
                                 BC      A7  9F 00138       PUSHAB  IOB$
                                 6A      03  FB 0013B       CALLS   #3, XPO$PUT
                                 67      24  A8  9E 0013E   MOVAB   $IOB$OUTPUT, IOB$+68                      ;1266
                        E8       A7      07  90 00142       MOVB    #7, IOB$+44
                                         59  DD 00146       PUSHL   R9
                                         7E  D4 00148       CLRL    -(SP)
                                 BC      A7  9F 0014A       PUSHAB  IOB$
                                 6A      03  FB 0014D       CALLS   #3, XPO$PUT
                                 67      2C  A8  9E 00150   MOVAB   $IOB$OUTPUT, IOB$+68                      ;1267
                        E8       A7      07  90 00154       MOVB    #7, IOB$+44
                                         59  DD 00158       PUSHL   R9
                                         7E  D4 0015A       CLRL    -(SP)
                                 BC      A7  9F 0015C       PUSHAB  IOB$
                                 6A      03  FB 0015F       CALLS   #3, XPO$PUT
                                 67      34  A8  9E 00162   MOVAB   $IOB$OUTPUT, IOB$+68                      ;1268
                        E8       A7      07  90 00166       MOVB    #7, IOB$+44
                                         59  DD 0016A       PUSHL   R9
                                         7E  D4 0016C       CLRL    -(SP)
                                 BC      A7  9F 0016E       PUSHAB  IOB$
                                 6A      03  FB 00171       CALLS   #3, XPO$PUT
                                 67      3C  A8  9E 00174   MOVAB   $IOB$OUTPUT, IOB$+68                      ;1269
                        E8       A7      07  90 00178       MOVB    #7, IOB$+44
                                         59  DD 0017C       PUSHL   R9
                                         7E  D4 0017E       CLRL    -(SP)
                                 BC      A7  9F 00180       PUSHAB  IOB$
                                 6A      03  FB 00183       CALLS   #3, XPO$PUT
                                 67      44  A8  9E 00186   MOVAB   $IOB$OUTPUT, IOB$+68                      ;1270
                        E8       A7      07  90 0018A       MOVB    #7, IOB$+44
                                         59  DD 0018E       PUSHL   R9
                                         7E  D4 00190       CLRL    -(SP)
                                 BC      A7  9F 00192       PUSHAB  IOB$
                                 6A      03  FB 00195       CALLS   #3, XPO$PUT
                                 67      4C  A8  9E 00198   MOVAB   $IOB$OUTPUT, IOB$+68                      ;1271
                        E8       A7      07  90 0019C       MOVB    #7, IOB$+44
                                         59  DD 001A0       PUSHL   R9
                                         7E  D4 001A2       CLRL    -(SP)
                                 BC      A7  9F 001A4       PUSHAB  IOB$
                                 6A      03  FB 001A7       CALLS   #3, XPO$PUT
                                 67      54  A8  9E 001AA   MOVAB   $IOB$OUTPUT, IOB$+68                      ;1272
                        E8       A7      07  90 001AE       MOVB    #7, IOB$+44
                                         59  DD 001B2       PUSHL   R9
                                         7E  D4 001B4       CLRL    -(SP)
                                 BC      A7  9F 001B6       PUSHAB  IOB$
```

```
                         6A           03  FB 0G1B9          CALLS   #3, XPO$PUT
                         67      5C   A8  9E 001BC          MOVAB   $IOB$OUTPUT, IOB$+68                1273
                      E8 A7           07  90 001C0          MOVB    #7, IOB$+44
                                      59  DD 001C4          PUSHL   R9
                                      7E  D4 001C6          CLRL    -(SP)
                             BC       A7  9F 001C8          PUSHAB  IOB$
                         6A           03  FB 001CB          CALLS   #3, XPO$PUT
                         67      64   A8  9E 001CE          MOVAB   $IOB$OUTPUT, IOB$+68                1274
                      E8 A7           07  90 001D2          MOVB    #7, IOB$+44
                                      59  DD 001D6          PUSHL   R9
                                      7E  D4 001D8          CLRL    -(SP)
                             BC       A7  9F 001DA          PUSHAB  IOB$
                         6A           03  FB 001DD          CALLS   #3, XPO$PUT
                                      7E  D4 0G1E0          CLRL    -(SP)                               1283
                                      68  DD 001E2          PUSHL   PAGE_WIDTH
                         7E      0903 8F  3C 001E4          MOVZWL  #2307, -(SP)
                00000000G EF          03  FB 001E9          CALLS   #3, XST$ASCII
                         52           50  D0 001F0          MOVL    R0, R2
                                      7E  D4 001F3          CLRL    -(SP)
                                      68  DD 001F5          PUSHL   PAGE_WIDTH
                         7E      0903 8F  3C 001F7          MOVZWL  #2307, -(SP)
                00000000G EF          03  FB 001FC          CALLS   #3, XST$ASCII
                                      50  DD 00203          PUSHL   R0
                             74       A8  9F 00205          PUSHAB  $STR$STRING2
                                      52  DD 00208          PUSHL   R2
                             6C       A8  9F 0020A          PUSHAB  $STR$STRING0
                00000000G EF          04  FB 0020D          CALLS   #4, XST$JOIN
                         67           50  D0 00214          MOVL    R0, IOB$+68
                      E8 A7           07  90 00217          MOVB    #7, IOB$+44
                                      59  DD 0021B          PUSHL   R9
                                      7E  D4 0021D          CLRL    -(SP)
                             BC       A7  9F 0021F          PUSHAB  IOB$
                         6A           03  FB 00222          CALLS   #3, XPO$PUT
                         67      7C   A8  9E 00225          MOVAB   $IOB$OUTPUT, IOB$+68                1285
                      E8 A7           07  90 00229          MOVB    #7, IOB$+44
                                      59  DD 0022D          PUSHL   R9
                                      7E  D4 0022F          CLRL    -(SP)
                             BC       A7  9F 00231          PUSHAB  IOB$
                         6A           03  FB 00234          CALLS   #3, XPO$PUT
                39    FD AB           04  E1 00237          BBC     #4, CMDBLK+1, 15$                   1287
                         67      0084 C8  9E 0023C          MOVAB   $IOB$OUTPUT, IOB$+68                1293
                      E8 A7           07  90 00241          MOVB    #7, IOB$+44
                                      59  DD 00245          PUSHL   R9
                                      7E  D4 00247          CLRL    -(SP)
                             BC       A7  9F 00249          PUSHAB  IOB$
                         6A           03  FB 0024C          CALLS   #3, XPO$PUT
                         67      008C C8  9E 0024F          MOVAB   $IOB$OUTPUT, IOB$+68                1294
                      E8 A7           07  90 00254          MOVB    #7, IOB$+44
                                      59  DD 00258          PUSHL   R9
                                      7E  D4 0025A          CLRL    -(SP)
                             BC       A7  9F 0025C          PUSHAB  IOB$
                         6A           03  FB 0025F          CALLS   #3, XPO$PUT
                         67      0094 C8  9E 00262          MOVAB   $IOB$OUTPUT, IOB$+68                1295
                      E8 A7           07  90 00267          MOVB    #7, IOB$+44
                                      59  DD 0026B          PUSHL   R9
                                      7E  D4 0026D          CLRL    -(SP)
                             BC       A7  9F 0026F          PUSHAB  IOB$
```

```
                                    6A              03  FB 00272          CALLS   #3, XPO$PUT
                                    04        02    AB  B1 00275 15$:     CMPW    CMDBLK+6, #4                    ;  1298
                                                    1D  12 00279          BNEQ    18$
                                    67      009C    C8  9E 0027B          MOVAB   $IOB$OUTPUT, IOB$+68            ;
                                                    72  11 00280          BRB     19$
                                    02              50  B1 00282 16$:     CMPW    R0, #2                          ;  1301
                                                    0E  19 00285          BLSS    17$
                                    03              50  B1 00287          CMPW    R0, #3                          ;
                                                    09  14 0028A          BGTR    17$
                          00000000G EF              00  FB 0028C          CALLS   #0, TMSINI                      ;  1305
                                                    6D  11 00293          BRB     20$
                                    04              50  B1 00295 17$:     CMPW    R0, #4                          ;  1307
                                                    68  12 00298 18$:     BNEQ    20$
                              F8    AD 00000000G    EF  B0 0029A          MOVW    NDXVRL, $STR$STRING1            ;  1312
                              FA    AD              0E  90 002A2          MOVB    #14, $STR$STRING1+2             ;
                              FB    AD              01  90 002A6          MOVB    #1, $STR$STRING1+3              ;
                              FC    AD 00000000G    EF  D0 002AA          MOVL    NDXVRP, $STR$STRING1+4          ;
                                             F8     AD  9F 002B2          PUSHAB  $STR$STRING1                    ;
                                           00A4     C8  9F 002B5          PUSHAB  $STR$STRING0                    ;
                          00000000G EF              02  FB 002B9          CALLS   #2, XST$JOIN                    ;
                                    67              50  D0 002C0          MOVL    R0, IOB$+68                     ;
                              E8    A7              07  90 002C3          MOVB    #7, IOB$+44                     ;
                                                    59  DD 002C7          PUSHL   R9                              ;
                                                    7E  D4 002C9          CLRL    -(SP)                           ;
                                             BC     A7  9F 002CB          PUSHAB  IOB$                            ;
                                    6A              03  FB 002CE          CALLS   #3, XPO$PUT                     ;
                                                    56  DD 002D1          PUSHL   DSC_PTR                         ;  1313
                                           00AC     C8  9F 002D3          PUSHAB  $STR$STRING0                    ;
                          00000000G EF              02  FB 002D7          CALLS   #2, XST$JOIN                    ;
                                    67              50  D0 002DE          MOVL    R0, IOB$+68                     ;
                              E8    A7              07  90 002E1          MOVB    #7, IOB$+44                     ;
                                                    59  DD 002E5          PUSHL   R9                              ;
                                                    7E  D4 002E7          CLRL    -(SP)                           ;
                                             BC     A7  9F 002E9          PUSHAB  IOB$                            ;
                                    6A              03  FB 002EC          CALLS   #3, XPO$PUT                     ;
                                    67      00B4    C8  9E 002EF          MOVAB   $IOB$OUTPUT, IOB$+68            ;  1314
                              E8    A7              07  90 002F4 19$:     MOVB    #7, IOB$+44                     ;
                                                    59  DD 002F8          PUSHL   R9                              ;
                                                    7E  D4 002FA          CLRL    -(SP)                           ;
                                             BC     A7  9F 002FC          PUSHAB  IOB$                            ;
                                    6A              03  FB 002FF          CALLS   #3, XPO$PUT                     ;
                                    56        34    AB  9E 00302 20$:     MOVAB   CMDBLK+56, DSC_PTR              ;  1319
                        03          FC      AB        02  E0 00306          BBS     #2, CMDBLK, 22$                 ;  1321
                                                  00A2  31 0030B 21$:     BRW     27$                             ;
                                                    66  B5 0030E 22$:     TSTW    (DSC_PTR)                       ;
                                                    F9  13 00310          BEQL    21$                             ;
  00F4    8F                00                  6E        00  2C 00312          MOVC5   #0, (SP), #0, #244, IOB$        ;  1330
                                                        6E    00319
                                    6E 0301003D     8F  D0 0031A          MOVL    #50397245, IOB$                 ;
                              1E    AE      020E    8F  B0 00321          MOVW    #526, IOB$RESULTANT+2           ;
                              04    AE              56  D0 00327          MOVL    DSC_PTR, IOB$+4                 ;  1333
                              2C    AE              01  90 0032B          MOVB    #1, IOB$+44                     ;
                                           00000000G  EF  9F 0032F          PUSHAB  OPEN_ERROR                      ;
                                                    7E  D4 00335          CLRL    -(SP)                           ;
                                             08     AE  9F 00337          PUSHAB  IOB$                            ;
                          00000000G EF              03  FB 0033A          CALLS   #3, XPO$OPEN                    ;
                              2C    AE              06  90 00341 23$:     MOVB    #6, IOB$+44                     ;  1335
```

```
                                    59  DD 00345          PUSHL   R9
                                    7E  D4 00347          CLRL    -(SP)
                                08  AE  9F 00349          PUSHAB  IOB$
                  00000000G     EF    03  FB 0034C          CALLS   #3, XPO$GET
                  00208001      8F    50  D1 00353          CMPL    R0, #2129921
                                      40  12 0035A          BNEQ    26$
                                50    6B  32 0035C          CVTWL   CMDBLK+4, R0                    1338
                                02    50  B1 0035F          CMPW    R0, #2                          1341
                                      1A  19 00362          BLSS    24$
                                03    50  B1 00364          CMPW    R0, #3
                                      15  14 00367          BGTR    24$
                                      7E  D4 00369          CLRL    -(SP)                           1342
                                BC    A7  9F 0036B          PUSHAB  OUTIOB
                                40    AE  DD 0036E          PUSHL   REQUIRE_IOB+56
                                7E    40  AE 3C 00371       MOVZWL  REQUIRE_IOB+52, -(SP)
                  00000000G     EF    04  FB 00375          CALLS   #4, TMSPUT
                                      C3  11 0037C          BRB     23$
                                01    50  B1 0037E 24$:     CMPW    R0, #1                          1344
                                      05  13 00381          BEQL    25$
                                04    50  B1 00383          CMPW    R0, #4
                                      B9  12 00386          BNEQ    23$
                                67    34  AE 9E 00388 25$:  MOVAB   $IOB$OUTPUT, IOB$+68           1345
                        E8      A7    07  90 0038C          MOVB    #7, IOB$+44
                                      59  DD 00390          PUSHL   R9
                                      7E  D4 00392          CLRL    -(SP)
                                BC    A7  9F 00394          PUSHAB  IOB$
                                6A    03  FB 00397          CALLS   #3, XPO$PUT
                                      A5  11 0039A          BRB     23$                             1335
                        2C      AE    02  90 0039C 26$:     MOVB    #2, IOB$+44                     1351
                                      59  DD 003A0          PUSHL   R9
                                      7E  D4 003A2          CLRL    -(SP)
                                08    AE  9F 003A4          PUSHAB  IOB$
                  00000000G     EF    03  FB 003A7          CALLS   #3, XPO$CLOSE
                                      2B  11 003AE          BRB     28$                             1321
                                01    6B  B1 003B0 27$:     CMPW    CMDBLK+4, #1                    1354
                                      26  12 003B3          BNEQ    28$
                                67    00BC  C8 9E 003B5     MOVAB   $IOB$OUTPUT, IOB$+68            1361
                        E8      A7    07  90 003BA          MOVB    #7, IOB$+44
                                      59  DD 003BE          PUSHL   R9
                                      7E  D4 003C0          CLRL    -(SP)
                                BC    A7  9F 003C2          PUSHAB  IOB$
                                6A    03  FB 003C5          CALLS   #3, XPO$PUT
                                67    00C4  C8 9E 003C8     MOVAB   $IOB$OUTPUT, IOB$+68            1362
                        E8      A7    07  90 003CD          MOVB    #7, IOB$+44
                                      59  DD 003D1          PUSHL   R9
                                      7E  D4 003D3          CLRL    -(SP)
                                BC    A7  9F 003D5          PUSHAB  IOB$
                                6A    03  FB 003D8          CALLS   #3, XPO$PUT
                                02    04  AB B1 003DB 28$:  CMPW    CMDBLK+8, #2                    1365
                                      09  13 003DF          BEQL    29$
                                      7E  D4 003E1          CLRL    -(SP)                           1372
                  00000000V     EF    01  FB 003E3          CALLS   #1, EXAM_BUCKET
                                52    01  D0 003EA 29$:     MOVL    #1, I                           1377
                                      52  DD 003ED 30$:     PUSHL   I                               1378
                  00000000V     EF    01  FB 003EF          CALLS   #1, EXAM_BUCKET
            F3                  52    1A  F3 003F6          AOBLEQ  #26, I, 30$
                                02    04  AB B1 003FA       CMPW    CMDBLK+8, #2                    1380
```

```
                                 09  12 003FE          BNEQ    31$                                     ;
                                 7E  D4 00400          CLRL    -(SP)                                   ; 1385
                                 01  FB 00402          CALLS   #1, EXAM_BUCKET                         ;
                                 00  FB 00409 31$:     CALLS   #0, LASTPG                              ; 1390
                                     04 00410          RET                                             ; 1392

; Routine Size:  1041 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  617          1393  1 %SBTTL 'EXAM_BUCKET -- Print contents of a bucket'
;  618          1394  1 ROUTINE EXAM_BUCKET (BKT) : NOVALUE =
;  619          1395  1 !++
;  620          1396  1 !
;  621          1397  1 ! FUNCTIONAL DESCRIPTION:
;  622          1398  1 !
;  623          1399  1 !       This routine is called to examine the contents of a bucket.
;  624          1400  1 !
;  625          1401  1 !       It generates a guide heading if one was requested via the
;  626          1402  1 !       /GUIDE_HEADING qualifier.
;  627          1403  1 !
;  628          1404  1 ! FORMAL PARAMETERS:
;  629          1405  1 !
;  630          1406  1 !       BKT - bucket number to examine
;  631          1407  1 !
;  632          1408  1 ! IMPLICIT INPUTS:
;  633          1409  1 !
;  634          1410  1 !       CMDBLK - command line information block
;  635          1411  1 !
;  636          1412  1 ! IMPLICIT OUTPUTS:
;  637          1413  1 !
;  638          1414  1 !       INDLVL  - subindex level is set to zero
;  639          1415  1 !       LSTPTR - current entry pointer is zeroed
;  640          1416  1 !       LINE_TYPE- is set if a guide heading is generated
;  641          1417  1 !                  and at the end of the bucket
;  642          1418  1 !
;  643          1419  1 ! ROUTINE VALUE:
;  644          1420  1 ! COMPLETION CODES:
;  645          1421  1 !
;  646          1422  1 !       None
;  647          1423  1 !
;  648          1424  1 ! SIDE EFFECTS:
;  649          1425  1 !
;  650          1426  1 !       None
;  651          1427  1 !--
;  652          1428  2     BEGIN
;  653          1429  2
;  654          1430  2     LOCAL
;  655          1431  2         LAST_SUB_BUCKET,
;  656          1432  2         TEXT_GENERATED;
;  657          1433  2
;  658          1434  2     TEXT_GENERATED = FALSE;
;  659          1435  2
;  660          1436  2     IF .BKT EQL 0
;  661          1437  2     THEN
;  662          1438  2         !
;  663          1439  2         ! Examining the nonalphabetic bucket (bucket 0).
;  664          1440  2         !
;  665          1441  2         ! The only sub-bucket used is zero because nonalphabetics do
;  666          1442  2         ! not sort correctly using the 'squared bucket' approach.
;  667          1443  2         !
;  668          1444  2         LAST_SUB_BUCKET = 0
;  669          1445  2     ELSE
;  670          1446  2         !
;  671          1447  2         ! Not examining the nonalphabetic bucket. Use sub-buckets 0-26.
;  672          1448  2         !
;  673          1449  2         LAST_SUB_BUCKET = 26;
```

```
;  674     1450  2
;  675     1451  2      INCR I FROM 0 TO .LAST_SUB_BUCKET DO
;  676     1452  3          BEGIN
;  677     1453  3          LSTPTR=0;                                   ! Set starting point for listing at head
;  678     1454  3          INDLVL=0;                                   ! Set starting indent level
;  679     1455  3
;  680     1456  3          IF GENTRY (.BKT, .I, FALSE)
;  681     1457  3          THEN
;  682     1458  4              BEGIN
;  683     1459  4              !
;  684     1460  4              ! Non empty bucket
;  685     1461  4              !
;  686     1462  5              IF .CMDBLK [NDX$V_GUIDE] AND (NOT .TEXT_GENERATED)
;  687     1463  4              THEN
;  688     1464  5                  BEGIN
;  689     1465  5                  !
;  690     1466  5                  ! Build guide heading
;  691     1467  5                  !
;  692     1468  5                  LINE_TYPE = GUIDE;
;  693     1469  5
;  694     1470  5                  IF .CMDBLK [NDX$H_FORMAT] EQL DSR
;  695     1471  5                  THEN
;  696     1472  6                      BEGIN
;  697     1473  6                      !
;  698     1474  6                      ! RUNOFF output
;  699     1475  6                      !
;  700     1476  6                      BIND
;  701     1477  6                          BUCKET_NAMES = UPLIT (
;  702     1478  6                              CH$PTR (UPLIT ('   ')),
;  703     1479  6                              CH$PTR (UPLIT ('-A-')),
;  704     1480  6                              CH$PTR (UPLIT ('-B-')),
;  705     1481  6                              CH$PTR (UPLIT ('-C-')),
;  706     1482  6                              CH$PTR (UPLIT ('-D-')),
;  707     1483  6                              CH$PTR (UPLIT ('-E-')),
;  708     1484  6                              CH$PTR (UPLIT ('-F-')),
;  709     1485  6                              CH$PTR (UPLIT ('-G-')),
;  710     1486  6                              CH$PTR (UPLIT ('-H-')),
;  711     1487  6                              CH$PTR (UPLIT ('-I-')),
;  712     1488  6                              CH$PTR (UPLIT ('-J-')),
;  713     1489  6                              CH$PTR (UPLIT ('-K-')),
;  714     1490  6                              CH$PTR (UPLIT ('-L-')),
;  715     1491  6                              CH$PTR (UPLIT ('-M-')),
;  716     1492  6                              CH$PTR (UPLIT ('-N-')),
;  717     1493  6                              CH$PTR (UPLIT ('-O-')),
;  718     1494  6                              CH$PTR (UPLIT ('-P-')),
;  719     1495  6                              CH$PTR (UPLIT ('-Q-')),
;  720     1496  6                              CH$PTR (UPLIT ('-R-')),
;  721     1497  6                              CH$PTR (UPLIT ('-S-')),
;  722     1498  6                              CH$PTR (UPLIT ('-T-')),
;  723     1499  6                              CH$PTR (UPLIT ('-U-')),
;  724     1500  6                              CH$PTR (UPLIT ('-V-')),
;  725     1501  6                              CH$PTR (UPLIT ('-W-')),
;  726     1502  6                              CH$PTR (UPLIT ('-X-')),
;  727     1503  6                              CH$PTR (UPLIT ('-Y-')),
;  728     1504  6                              CH$PTR (UPLIT ('-Z-'))
;  729     1505  6                      ) : VECTOR;
;  730     1506  6
```

```
;  731     1507  6                    LOCAL
;  732     1508  6                        INDENT;
;  733     1509  6
;  734     1510  6                    INDENT = .CMDBLK [NDX$G_COLUMN_WID] / 3;     ! Number of leading blanks
;  735     1511  6                    INDENT = ((.INDENT + 1) / 2) * 2;            ! Round to next even number
;  736     1512  6
;  737     1513  6                    INDENT_LINE (.INDENT);
;  738     1514  6                    FILL_LINE (3, .BUCKET_NAMES [.BKT], .INDENT, .CMDBLK [NDX$G_COLUMN_WID], TRUE, FALSE, FA
;  739     1515  6                    INS_LINE (.INT_LINE_LENGTH, CH$PTR (RNO_LINE), .EXT_LINE_LENGTH);
;  740     1516  6                    END
;  741     1517  5                ELSE
;  742     1518  6                    BEGIN
;  743     1519  6                    !
;  744     1520  6                    ! TMS11 or TEX output
;  745     1521  6                    !
;  746     1522  6                    BIND
;  747     1523  6                        BUCKET_NAMES = UPLIT (
;  748     1524  6                            CH$PTR (UPLIT (' ')),
;  749     1525  6                            CH$PTR (UPLIT ('A')),
;  750     1526  6                            CH$PTR (UPLIT ('B')),
;  751     1527  6                            CH$PTR (UPLIT ('C')),
;  752     1528  6                            CH$PTR (UPLIT ('D')),
;  753     1529  6                            CH$PTR (UPLIT ('E')),
;  754     1530  6                            CH$PTR (UPLIT ('F')),
;  755     1531  6                            CH$PTR (UPLIT ('G')),
;  756     1532  6                            CH$PTR (UPLIT ('H')),
;  757     1533  6                            CH$PTR (UPLIT ('I')),
;  758     1534  6                            CH$PTR (UPLIT ('J')),
;  759     1535  6                            CH$PTR (UPLIT ('K')),
;  760     1536  6                            CH$PTR (UPLIT ('L')),
;  761     1537  6                            CH$PTR (UPLIT ('M')),
;  762     1538  6                            CH$PTR (UPLIT ('N')),
;  763     1539  6                            CH$PTR (UPLIT ('O')),
;  764     1540  6                            CH$PTR (UPLIT ('P')),
;  765     1541  6                            CH$PTR (UPLIT ('Q')),
;  766     1542  6                            CH$PTR (UPLIT ('R')),
;  767     1543  6                            CH$PTR (UPLIT ('S')),
;  768     1544  6                            CH$PTR (UPLIT ('T')),
;  769     1545  6                            CH$PTR (UPLIT ('U')),
;  770     1546  6                            CH$PTR (UPLIT ('V')),
;  771     1547  6                            CH$PTR (UPLIT ('W')),
;  772     1548  6                            CH$PTR (UPLIT ('X')),
;  773     1549  6                            CH$PTR (UPLIT ('Y')),
;  774     1550  6                            CH$PTR (UPLIT ('Z'))
;  775     1551  6                        ) : VECTOR;
;  776     1552  6
;  777     1553  6                    INS_LINE (1, .BUCKET_NAMES [.BKT], 1);
;  778     1554  5                    END;
;  779     1555  5
;  780     1556  5                LINE_TYPE = GUIDE_FILL;
;  781     1557  5                INS_LINE (4, .BLANKS, 4);
;  782     1558  4                END;
;  783     1559  4
;  784     1560  4            !
;  785     1561  4            ! Now process rest of bucket
;  786     1562  4            !
;  787     1563  4            DO FORMAT_ENTRY () UNTIL NOT GENTRY (.BKT, .I, FALSE);
```

```
;  788          1564  4
;  789          1565  4              TEXT_GENERATED = TRUE;
;  790          1566  3              END;
;  791          1567  3
;  792          1568  2          END;
;  793          1569  2
;  794          1570  2      IF .TEXT_GENERATED
;  795          1571  2      THEN
;  796          1572  3          BEGIN
;  797          1573  3          LINE_TYPE = BKT_E;
;  798          1574  3          INS_LINE (4, .BLANKS, 4);       ! Skip a line
;  799          1575  2          END;
;  800          1576  2
;  801          1577  1      END;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                 0020B          .BLKB   1
                                          00  20  20  20 0020C P.ABM:  .ASCII  \    \<0>
                                          00  2D  41  2D 00210 P.ABN:  .ASCII  \-A-\<0>
                                          00  2D  42  2D 00214 P.ABO:  .ASCII  \-B-\<0>
                                          00  2D  43  2D 00218 P.ABP:  .ASCII  \-C-\<0>
                                          00  2D  44  2D 0021C P.ABQ:  .ASCII  \-D-\<0>
                                          00  2D  45  2D 00220 P.ABR:  .ASCII  \-E-\<0>
                                          00  2D  46  2D 00224 P.ABS:  .ASCII  \-F-\<0>
                                          00  2D  47  2D 00228 P.ABT:  .ASCII  \-G-\<0>
                                          00  2D  48  2D 0022C P.ABU:  .ASCII  \-H-\<0>
                                          00  2D  49  2D 00230 P.ABV:  .ASCII  \-I-\<0>
                                          00  2D  4A  2D 00234 P.ABW:  .ASCII  \-J-\<0>
                                          00  2D  4B  2D 00238 P.ABX:  .ASCII  \-K-\<0>
                                          00  2D  4C  2D 0023C P.ABY:  .ASCII  \-L-\<0>
                                          00  2D  4D  2D 00240 P.ABZ:  .ASCII  \-M-\<0>
                                          00  2D  4E  2D 00244 P.ACA:  .ASCII  \-N-\<0>
                                          00  2D  4F  2D 00248 P.ACB:  .ASCII  \-O-\<0>
                                          00  2D  50  2D 0024C P.ACC:  .ASCII  \-P-\<0>
                                          00  2D  51  2D 00250 P.ACD:  .ASCII  \-Q-\<0>
                                          00  2D  52  2D 00254 P.ACE:  .ASCII  \-R-\<0>
                                          00  2D  53  2D 00258 P.ACF:  .ASCII  \-S-\<0>
                                          00  2D  54  2D 0025C P.ACG:  .ASCII  \-T-\<0>
                                          00  2D  55  2D 00260 P.ACH:  .ASCII  \-U-\<0>
                                          00  2D  56  2D 00264 P.ACI:  .ASCII  \-V-\<0>
                                          00  2D  57  2D 00268 P.ACJ:  .ASCII  \-W-\<0>
                                          00  2D  58  2D 0026C P.ACK:  .ASCII  \-X-\<0>
                                          00  2D  59  2D 00270 P.ACL:  .ASCII  \-Y-\<0>
                                          00  2D  5A  2D 00274 P.ACM:  .ASCII  \-Z-\<0>
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00278 P.ABL:  .ADDRESS P.ABM, P.ABN, P.ABO, P.ABP, P.ABQ, -
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00290                  P.ABR, P.ABS, P.ABT, P.ABU, P.ABV, P.ABW, -
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 002A8                  P.ABX, P.ABY, P.ABZ, P.ACA, P.ACB, P.ACC, -
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 002C0                  P.ACD, P.ACE, P.ACF, P.ACG, P.ACH, P.ACI, -
                              00000000' 00000000' 00000000' 002D8                  P.ACJ, P.ACK, P.ACL, P.ACM
                                          00  00  00  20 002E4 P.ACO:  .ASCII  \ \<0><0><0>
                                          00  00  00  41 002E8 P.ACP:  .ASCII  \A\<0><0><0>
                                          00  00  00  42 002EC P.ACQ:  .ASCII  \B\<0><0><0>
                                          00  00  00  43 002F0 P.ACR:  .ASCII  \C\<0><0><0>
                                          00  00  00  44 002F4 P.ACS:  .ASCII  \D\<0><0><0>
```

```
                                                00  00  00  45  002F8 P.ACT:   .ASCII  \E\<0><0><0>                                                ;
                                                00  00  00  46  002FC P.ACU:   .ASCII  \F\<0><0><0>                                                ;
                                                00  00  00  47  00300 P.ACV:   .ASCII  \G\<0><0><0>                                                ;
                                                00  00  00  48  00304 P.ACW:   .ASCII  \H\<0><0><0>                                                ;
                                                00  00  00  49  00308 P.ACX:   .ASCII  \I\<0><0><0>                                                ;
                                                00  00  00  4A  0030C P.ACY:   .ASCII  \J\<0><0><0>                                                ;
                                                00  00  00  4B  00310 P.ACZ:   .ASCII  \K\<0><0><0>                                                ;
                                                00  00  00  4C  00314 P.ADA:   .ASCII  \L\<0><0><0>                                                ;
                                                00  00  00  4D  00318 P.ADB:   .ASCII  \M\<0><0><0>                                                ;
                                                00  00  00  4E  0031C P.ADC:   .ASCII  \N\<0><0><0>                                                ;
                                                00  00  00  4F  00320 P.ADD:   .ASCII  \O\<0><0><0>                                                ;
                                                00  00  00  50  00324 P.ADE:   .ASCII  \P\<0><0><0>                                                ;
                                                00  00  00  51  00328 P.ADF:   .ASCII  \Q\<0><0><0>                                                ;
                                                00  00  00  52  0032C P.ADG:   .ASCII  \R\<0><0><0>                                                ;
                                                00  00  00  53  00330 P.ADH:   .ASCII  \S\<0><0><0>                                                ;
                                                00  00  00  54  00334 P.ADI:   .ASCII  \T\<0><0><0>                                                ;
                                                00  00  00  55  00338 P.ADJ:   .ASCII  \U\<0><0><0>                                                ;
                                                00  00  00  56  0033C P.ADK:   .ASCII  \V\<0><0><0>                                                ;
                                                00  00  00  57  00340 P.ADL:   .ASCII  \W\<0><0><0>                                                ;
                                                00  00  00  58  00344 P.ADM:   .ASCII  \X\<0><0><0>                                                ;
                                                00  00  00  59  00348 P.ADN:   .ASCII  \Y\<0><0><0>                                                ;
                                                00  00  00  5A  0034C P.ADO:   .ASCII  \Z\<0><0><0>                                                ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00350 P.ACN:   .ADDRESS P.ACO, P.ACP, P.ACQ, P.ACR, P.ACS, -                           ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00368                    P.ACT, P.ACU, P.ACV, P.ACW, P.ACX, P.ACY, -                ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00380                    P.ACZ, P.ADA, P.ADB, P.ADC, P.ADD, P.ADE, -                ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00398                    P.ADF, P.ADG, P.ADH, P.ADI, P.ADJ, P.ADK, -                ;
                              00000000' 00000000' 00000000' 003B0                    P.ADL, P.ADM, P.ADN, P.ADO                                 ;

                                                                BUCKET_NAMES=       P.ABL
                                                                BUCKET_NAMES=       P.ACN


                                                                .PSECT  $CODE$,NOWRT,2

                                                OFFC 00000 EXAM_BUCKET:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11                                ; 1394
5B 000000000G   EF  9E 00002                                    MOVAB   GENTRY, R11                                                         ;
5A 00000000V    EF  9E 00009                                    MOVAB   INS_LINE, R10                                                       ;
59 000000000G   EF  9E 00010                                    MOVAB   CMDBLK+12, R9                                                       ;
58 00000000'    EF  9E 00017                                    MOVAB   LINE_TYPE, R8                                                       ;
                56  D4 0001E                                    CLRL    TEXT_GENERATED                                                      ; 1434
54          04  AC  D0 00020                                    MOVL    BKT, R4                                                             ; 1436
                04  12 00024                                    BNEQ    1$                                                                  ;
                57  D4 00026                                    CLRL    LAST_SUB_BUCKET                                                     ; 1444
                03  11 00028                                    BRB     2$                                                                  ;
57              1A  D0 0002A 1$:                                MOVL    #26, LAST_SUB_BUCKET                                                ; 1449
55              01  CE 0002D 2$:                                MNEGL   #1, I                                                               ; 1451
              009B  31 00030 3$:                                BRW     8$                                                                  ;
    000000000G  EF  D4 00033 4$:                                CLRL    LSTPTR                                                              ; 1453
    000000000G  EF  D4 00039                                    CLRL    INDLVL                                                              ; 1454
                7E  D4 0003F                                    CLRL    -(SP)                                                               ; 1456
                30  BB 00041                                    PUSHR   #^M<R4,R5>                                                          ;
6B              03  FB 00043                                    CALLS   #3, GENTRY                                                          ;
E7              50  E9 00046                                    BLBC    R0, 3$                                                              ;
            F4  A9  95 00049                                    TSTB    CMDBLK                                                              ; 1462
                6C  18 0004C                                    BGEQ    7$                                                                  ;
69              56  E8 C004E                                    BLBS    TEXT_GENERATED, 7$                                                  ;
```

```
                        68              03  D0 00051          MOVL    #3, LINE_TYPE                           ; 1468
                53      54              02  78 00054          ASHL    #2, R4, R3                              ; 1514
                        01        F8    A9  B; 00058          CMPW    CMDBLK+4, #1                            ; 1470
                                        3E  12 0005C          BNEQ    5$
                52      69              03  C7 0005E          DIVL3   #3, CMDBLK+12, INDENT                   ; 1510
                        50        01    A2  9E 00062          MOVAB   1(R2), R0                               ; 1511
                        50              02  C6 00066          DIVL2   #2, R0
                52      50              01  78 00069          ASHL    #1, R0, INDENT
                                        52  DD 0006D          PUSHL   INDENT                                  ; 1513
           00000000V    EF              01  FB 0006F          CALLS   #1, INDENT_LINE
                                        7E  7C 00076          CLRQ    -(SP)                                   ; 1514
                                        01  DD 00078          PUSHL   #1
                                        69  DD 0007A          PUSHL   CMDBLK+12
                                        52  DD 0007C          PUSHL   INDENT
                          00000000'EF   43  9F 0007E          PUSHAB  BUCKET_NAMES[R3]
                                        9E  DD 00085          PUSHL   @(SP)+
                                        03  DD 00087          PUSHL   #3
           00000000V    EF              07  FB 00089          CALLS   #7, FILL_LINE
                              F8        A8  DD 00090          PUSHL   EXT_LINE_LENGTH                         ; 1515
                            FB44        C8  9F 00093          PUSHAB  RNO_LINE
                              F4        A8  DD 00097          PUSHL   INT_LINE_LENGTH
                                        0D  11 0009A          BRB     6$
                                        01  DD 0009C 5$:      PUSHL   #1                                      ; 1553
                          00000000'EF   43  9F 0009E          PUSHAB  BUCKET_NAMES[R3]
                                        9E  DD 000A5          PUSHL   @(SP)+
                                        01  DD 000A7          PUSHL   #1
                        6A              03  FB 000A9 6$:      CALLS   #3, INS_LINE
                        68              04  D0 000AC          MOVL    #4, LINE_TYPE                           ; 1554
                                        04  DD 000AF          PUSHL   #4                                      ; 1557
                              FB3C      C8  DD 000B1          PUSHL   BLANKS
                                        04  DD 000B5          PUSHL   #4
                        6A              03  FB 000B7          CALLS   #3, INS_LINE
           00000000V    EF              00  FB 000BA 7$:      CALLS   #0, FORMAT_ENTRY                        ; 1563
                                        7E  D4 000C1          CLRL    -(SP)
                                        30  BB 000C3          PUSHR   #^M<R4,R5>
                        6B              03  FB 000C5          CALLS   #3, GENTRY
                        EF              50  E8 000C8          BLBS    R0, 7$
                        56              01  D0 000CB          MOVL    #1, TEXT_GENERATED                      ; 1565
        F; >F     55    01              57  F1 000CE 8$:      ACBL    LAST_SUB_BUCKET, #1, I, 4$              ; 1451
                        0E              56  E9 000D4          BLBC    TEXT_GENERATED, 9$                      ; 1570
                        68              01  D0 000D7          MOVL    #1, LINE_TYPE                           ; 1573
                                        04  DD 000DA          PUSHL   #4                                      ; 1574
                              FB3C      C8  DD 000DC          PUSHL   BLANKS
                                        04  DD 000E0          PUSHL   #4
                        6A              03  FB 000E2          CALLS   #3, INS_LINE
                                        04  000E5 9$:         RET                                             ; 1577

; Routine Size:  230 bytes,    Routine Base:  $CODE$ + 0411

;  802          1578  1
```

```
;  804          1579  1 %SBTTL 'FORMAT_ENTRY -- collect & print index entry'
;  805          1580  1 ROUTINE FORMAT_ENTRY : NOVALUE =
;  806          1581  1
;  807          1582  1 !++
;  808          1583  1 ! FUNCTIONAL DESCRIPTION:
;  809          1584  1 !
;  810          1585  1 !       Collect and print the entire index line, including
;  811          1586  1 !       name and all line number references.
;  812          1587  1 !
;  813          1588  1 !       No line is generated if doing a master index and the
;  814          1589  1 !       subindex level is greater than the deepest level
;  815          1590  1 !       allowed in the master index.
;  816          1591  1 !
;  817          1592  1 ! FORMAL PARAMETERS:
;  818          1593  1 !
;  819          1594  1 !       None
;  820          1595  1 !
;  821          1596  1 ! IMPLICIT INPUTS:
;  822          1597  1 !
;  823          1598  1 !       LSTPTR          - address of list item for index entry
;  824          1599  1 !       INDLVL          - subindex level
;  825          1600  1 !       LSTSTK          - subindex list stack
;  826          1601  1 !       LINE_TYPE       - current line type
;  827          1602  1 !       CMDBLK          - command line information block
;  828          1603  1 !       CHRSIZ          - TMS character size vector
;  829          1604  1 !
;  830          1605  1 ! IMPLICIT OUTPUTS:
;  831          1606  1 !
;  832          1607  1 !       LINE_TYPE       - current line type
;  833          1608  1 !       LINE_POINTER    - pointer to next character position in output line
;  834          1609  1 !       EXT_LINE_LENGTH - external line length
;  835          1610  1 !       INT_LINE_LENGTH - internal line length
;  836          1611  1 !       TMS_LINE_LENGTH - length of line in TMS relative units
;  837          1612  1 !
;  838          1613  1 ! ROUTINE VALUE:
;  839          1614  1 ! COMPLETION CODES:
;  840          1615  1 !
;  841          1616  1 !       None
;  842          1617  1 !
;  843          1618  1 ! SIDE EFFECTS:
;  844          1619  1 !
;  845          1620  1 !       None
;  846          1621  1 !
;  847          1622  1 !--
;  848          1623  1
;  849          1624  2     BEGIN
;  850          1625  2
;  851          1626  2     LOCAL
;  852          1627  2         INDENT,
;  853          1628  2         STR_VEC : REF VECTOR,
;  854          1629  2         CHAR_COUNT,
;  855          1630  2         TEXT_PTR;
;  856          1631  2
;  857          1632  2     IF .INDLVL GTR .CMDBLK [NDX$H_LEVEL] THEN RETURN;
;  858          1633  2
;  859          1634  2     !
;  860          1635  2     ! Get pointer to text and length of text
```

```
;  861          1636  2      !
;  862          1637  2      STR_VEC = .LSTPTR [XE$A_TEXT];
;  863          1638  2      CHAR_COUNT = .STR_VEC [0];                          ! Length is 1st fullword of string
;  864          1639  2      TEXT_PTR = CH$PTR (STR_VEC [1]);
;  865          1640  2
;  866          1641  2      IF .LINE_TYPE NEQ CONT_HEAD THEN LINE_TYPE = (IF .INDLVL EQL 0 THEN ENTRY_B ELSE SUB_B);
;  867          1642  2
;  868          1643  2      !
;  869          1644  2      ! Indent the new line
;  870          1645  2      !
;  871          1646  2      INDENT = 2 * .INDLVL;
;  872          1647  2      INDENT_LINE (.INDENT);
;  873          1648  2
;  874          1649  2      !
;  875          1650  2      ! Now format the entry
;  876          1651  2      !
;  877          1652  2      FILL_LINE (.CHAR_COUNT, .TEXT_PTR, .INDENT, .CMDBLK [NDX$G_COLUMN_WID], FALSE, FALSE, FALSE);
;  878          1653  2
;  879          1654  2      IF .LINE_TYPE EQL CONT_HEAD
;  880          1655  2      THEN
;  881          1656  3          BEGIN
;  882          1657  3          !
;  883          1658  3          ! Doing a continuation heading
;  884          1659  3          !
;  885          1660  4          IF (.LSTSTK [.INDLVL + 1] EQL 0)
;  886          1661  4          OR (.INDLVL EQL .CMDBLK [NDX$H_LEVEL])
;  887          1662  3          THEN
;  888          1663  3              !
;  889          1664  3              ! Append ' (Cont.)' to line
;  890          1665  3              !
;  891          1666  3              FILL_LINE (7, CH$PTR (UPLIT ('(Cont.)')), .INDENT, .CMDBLK [NDX$G_COLUMN_WID], FALSE, FALSE, TRU
;  892          1667  3          END
;  893          1668  2      ELSE
;  894          1669  3          BEGIN
;  895          1670  3          !
;  896          1671  3          ! Doing a normal entry
;  897          1672  3          !
;  898          1673  3          IF .LSTPTR [XE$A_REF] NEQ 0
;  899          1674  3          THEN
;  900          1675  4              BEGIN
;  901          1676  4              !
;  902          1677  4              ! Have page references.
;  903          1678  4              !
;  904          1679  4              IF .CMDBLK [NDX$V_PAGES]                    ! Generating page references
;  905          1680  4              OR .CMDBLK [NDX$V_MASTER]                   ! or master index
;  906          1681  4              THEN
;  907          1682  5                  BEGIN
;  908          1683  5                  !
;  909          1684  5                  ! Insert a null following the entry so that we can detect the
;  910          1685  5                  ! start of the page references for building telltale headings
;  911          1686  5                  ! or continuation headings on the last page.
;  912          1687  5                  !
;  913          1688  5                  CH$WCHAR_A (0, LINE_POINTER);
;  914          1689  5                  EXT_LINE_LENGTH = .EXT_LINE_LENGTH + 1;
;  915          1690  5                  INT_LINE_LENGTH = .INT_LINE_LENGTH + 1;
;  916          1691  5
;  917          1692  5                  TMS_LINE_LENGTH = .TMS_LINE_LENGTH
```

```
;  918     1693  5                     + (IF .CMDBLK [NDX$H_LAYOUT] EQL SEPARATE THEN .CHRSIZ [%C' '] ELSE .CHRSIZ [%C',']);
;  919     1694  5
;  920     1695  5                 INS_PAGE_NO (.LSTPTR [XE$A_REF]); ! Insert page numbers
;  921     1696  4                 END;
;  922     1697  4             END
;  923     1698  3         ELSE
;  924     1699  4             BEGIN
;  925     1700  4             !
;  926     1701  4             ! No page references
;  927     1702  4             !
;  928     1703  4             IF  .CMDBLK [NDX$V_MASTER]
;  929     1704  5             AND (.INDLVL EQL .CMDBLK [NDX$H_LEVEL])
;  930     1705  5             AND (.LSTPTR [XE$A_SUBX] NEQ 0)
;  931     1706  4             THEN
;  932     1707  5                 BEGIN
;  933     1708  5                 !
;  934     1709  5                 ! Doing master indexing, at deepest level specified,
;  935     1710  5                 ! and not at the bottom of a .Y - generate book references
;  936     1711  5                 !
;  937     1712  5                 LOCAL
;  938     1713  5                     M_INDENT,
;  939     1714  5                     M_EXT,
;  940     1715  5                     NEW_LINE,
;  941     1716  5                     XMPTR : REF $XM_BLOCK;
;  942     1717  5
;  943     1718  5                 NEW_LINE = FALSE;
;  944     1719  5                 XMPTR = .LSTPTR [XE$A_BOOK_LIST];
;  945     1720  5
;  946     1721  5                 IF .CMDBLK [NDX$H_LAYOUT] EQL SEPARATE
;  947     1722  5                 THEN
;  948     1723  6                     BEGIN
;  949     1724  6                     !
;  950     1725  6                     ! Set indent and maximum external length for separate format
;  951     1726  6                     !
;  952     1727  6                     M_INDENT = .CMDBLK [NDX$G_COLUMN_WID] + .CMDBLK [NDX$G_GUTTER_WID];
;  953     1728  6                     M_EXT = .PAGE_WIDTH;
;  954     1729  6                     END
;  955     1730  5                 ELSE
;  956     1731  6                     BEGIN
;  957     1732  6                     !
;  958     1733  6                     ! Set indent and maximum external length for other formats.
;  959     1734  6                     !
;  960     1735  6                     M_INDENT = .INDENT;
;  961     1736  6                     M_EXT = .CMDBLK [NDX$G_COLUMN_WID],
;  962     1737  5                     END;
;  963     1738  5
;  964     1739  5                 !
;  965     1740  5                 ! Insert a null following the entry so that we can detect the
;  966     1741  5                 ! start of the page references for building telltale headings
;  967     1742  5                 ! or continuation headings on the last page.
;  968     1743  5                 !
;  969     1744  5                 CH$WCHAR_A (0, LINE_POINTER);
;  970     1745  5                 EXT_LINE_LENGTH = .EXT_LINE_LENGTH + 1;
;  971     1746  5                 INT_LINE_LENGTH = .INT_LINE_LENGTH + 1;
;  972     1747  5
;  973     1748  5                 TMS_LINE_LENGTH = .TMS_LINE_LENGTH
;  974     1749  5                     + (IF .CMDBLK [NDX$H_LAYOUT] EQL SEPARATE THEN .CHRSIZ [%C' '] ELSE .CHRSIZ [%C',']);
```

```
;  975          1750  5
;  976          1751  5                  WHILE .XMPTR NEQ 0 DO
;  977          1752  6                      BEGIN
;  978          1753  6                      BOOK_REF (.XMPTR [XM$A_BOOK], .M_INDENT, .M_EXT, .NEW_LINE, FALSE);
;  979          1754  6                      IF .CMDBLK [NDX$H_LAYOUT] EQL SEPARATE THEN NEW_LINE = TRUE;
;  980          1755  6
;  981          1756  6                      IF .XMPTR [XM$A_LINK] NEQ 0
;  982          1757  6                      THEN
;  983          1758  6                          XMPTR = .XMPTR [XM$A_LINK]
;  984          1759  6                      ELSE
;  985          1760  6                          XMPTR = 0;
;  986          1761  5                      END;
;  987          1762  4                  END;
;  988          1763  3              END;
;  989          1764  2          END;
;  990          1765  2
;  991          1766  2      IF .LINE_TYPE NEQ CONT_HEAD THEN LINE_TYPE = (IF .INDLVL EQL 0 THEN ENTRY_E ELSE SUB_E);
;  992          1767  2
;  993          1768  2      INS_LINE (.INT_LINE_LENGTH, CH$PTR (RNO_LINE), .EXT_LINE_LENGTH);
;  994          1769  2
;  995          1770  1      END;                            !End of FORMAT_ENTRY


                                                            .PSECT  $PLIT$,NOWRT,NOEXE,2
                    00  29  2E  74  6E  6F  43  28  003BC P.ADP:  .ASCII  \(Cont.)\<0>                                    ;


                                                            .PSECT  $CODE$,NOWRT,2

                                                OFFC 00000 FORMAT_ENTRY:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11                 ; 1580
                                5B 00000000G    EF   9E 00002          MOVAB   CHRSIZ, R11
                                5A 00000000V    EF   9E 00009          MOVAB   FILL_LINE, R10
                                59 00000000G    EF   9E 00010          MOVAB   LSTPTR, R9
                                58 00000000G    EF   9E 00017          MOVAB   INDLVL, R8
                                57 00000000G    EF   9E 0001E          MOVAB   CMDBLK+12, R7
                                56 00000000'    EF   9E 00025          MOVAB   LINE_TYPE, R6
                                51              68   D0 0002C          MOVL    INDLVL, R1                              ; 1632
                51      FE  A7  10              00   EC 0002F          CMPV    #0, #16, CMDBLK+10, R1
                                                01   18 00035          BGEQ    1$
                                                     04 00037          RET
                                50              69   D0 00038 1$:      MOVL    LSTPTR, R0                              ; 1637
                                50          10  A0   D0 0003B          MOVL    16(R0), STR_VEC
                                54              60   D0 0003F          MOVL    (STR_VEC), CHAR_COUNT                   ; 1638
                                52          04  A0   9E 00042          MOVAB   4(R0), TEXT_PTR                         ; 1639
                                08              66   D1 00046          CMPL    LINE_TYPE, #11                          ; 1641
                                                0F   13 00049          BEQL    4$
                                                51   D5 0004B          TSTL    R1
                                                05   12 0004D          BNEQ    2$
                                50              05   D0 0004F          MOVL    #5, R0
                                                03   11 00052          BRB     3$
                                50              08   D0 00054 2$:      MOVL    #8, R0
                                66              50   D0 00057 3$:      MOVL    R0, LINE_TYPE
                            53                  51   01 78 0005A 4$:   ASHL    #1, R1, INDENT                          ; 1646
```

```
                                       53  DD 0005E          PUSHL   INDENT                                  ; 1647
                    00000000V  EF      01  FB 00060          CALLS   #1, INDENT_LINE                         ;
                                       7E  7C 00067          CLRQ    -(SP)                                   ; 1652
                                       7E  D4 00069          CLRL    -(SP)                                   ;
                                       67  DD 0006B          PUSHL   CMDBLK+12                               ;
                                       0C  BB 0006D          PUSHR   #^M<R2,R3>                              ;
                                       54  DD 0006F          PUSHL   CHAR_COUNT                              ;
                               6A      07  FB 00071          CALLS   #7, FILL_LINE                           ;
                               0B      66  D1 00074          CMPL    LINE_TYPE, #11                          ; 1654
                                       29  12 00077          BNEQ    6$                                      ;
                               50      68  D0 00079          MOVL    INDLVL, R0                              ; 1660
                                  00000000GEF40  D5 0007C    TSTL    LSTSTK+4[R0]                            ;
                                       08  13 00083          BEQL    5$                                      ;
     68       FE   A7          10      00  EC 00085          CMPV    #0, #16, CMDBLK+10, INDLVL              ; 1661
                                       66  12 0008B          BNEQ    12$                                     ;
                                       01  DD 0008D  5$:     PUSHL   #1                                      ; 1666
                                       7E  7C 0008F          CLRQ    -(SP)                                   ;
                                       67  DD 00091          PUSHL   CMDBLK+12                               ;
                                       53  DD 00093          PUSHL   INDENT                                  ;
                        00000000'  EF  9F 00095              PUSHAB  P.ADP                                   ;
                                       07  DD 0009B          PUSHL   #7                                      ;
                               6A      07  FB 0009D          CALLS   #7, FILL_LINE                           ;
                                       43  11 000A0          BRB     10$                                     ; 1656
                               51      69  D0 000A2  6$:     MOVL    LSTPTR, R1                              ; 1673
                                 0C    A1  D5 000A5          TSTL    12(R1)                                  ;
                                       3E  13 000A8          BEQL    11$                                     ;
               05       F4     A7      03  E0 000AA          BBS     #3, CMDBLK, 7$                          ; 1679
               34       F5     A7      02  E1 000AF          BBC     #2, CMDBLK+1, 11$                       ; 1680
                               FB40    D6  94 000B4  7$:     CLRB    @LINE_POINTER                           ; 1688
                               FB40    C6  D6 000B8          INCL    LINE_POINTER                            ;
                                 F8    A6  D6 000BC          INCL    EXT_LINE_LENGTH                         ; 1689
                                 F4    A6  D6 000BF          INCL    INT_LINE_LENGTH                         ; 1690
                               50      6B  D0 000C2          MOVL    CHRSIZ, R0                              ; 1693
                               03  FA  A7  B1 000C5          CMPW    CMDBLK+6, #3                            ;
                                       07  12 000C9          BNEQ    8$                                      ;
                               50  0080    C0  D0 000CB      MOVL    128(R0), R0                             ;
                                       05  11 000D0          BRB     9$                                      ;
                               50  00B0    C0  D0 000D2  8$: MOVL    176(R0), R0                             ;
                        FC     A6      50  C0 000D7  9$:     ADDL2   R0, TMS_LINE_LENGTH                     ;
                                 0C    A1  DD 000DB          PUSHL   12(R1)                                  ; 1695
                    00000000V  EF      01  FB 000DE          CALLS   #1, INS_PAGE_NO                         ;
                                     0081  31 000E5  10$:    BRW     20$                                     ; 1673
               7C       F5     A7      02  E1 000E8  11$:    BBC     #2, CMDBLK+1, 20$                       ; 1703
     68       FE   A7          10      00  EC 000ED          CMPV    #0, #16, CMDBLK+10, INDLVL              ; 1704
                                       74  12 000F3  12$:    BNEQ    20$                                     ;
                                 08    A1  D5 000F5          TSTL    8(R1)                                   ; 1705
                                       6F  13 000F8          BEQL    20$                                     ;
                                       54  D4 000FA          CLRL    NEW_LINE                                ; 1718
                               52  1C  A1  D0 000FC          MOVL    28(R1), XMPTR                           ; 1719
                                       51  D4 00100          CLRL    R1                                      ; 1721
                               03  FA  A7  B1 00102          CMPW    CMDBLK+6, #3                            ;
                                       0D  12 00106          BNEQ    13$                                     ;
                                       51  D6 00108          INCL    R1                                      ;
               55              67  04  A7  C1 0010A          ADDL3   CMDBLK+16, CMDBLK+12, M_INDENT          ; 1727
                               53  04  A6  D0 0010F          MOVL    PAGE_WIDTH, M_EXT                       ; 1728
                                       06  11 00113          BRB     14$                                     ; 1721
                               55      53  D0 00115  13$:    MOVL    INDENT, M_INDENT                        ; 1735
```

```
                          53              67  D0 00118          MOVL    CMDBLK+12, M_EXT                ; 1736
                                    FB40  D6  94 0011B 14$:     CLRB    @LINE_POINTER                   ; 1744
                                    FB40  C6  D6 0011F          INCL    LINE_POINTER                    ;
                                      F8  A6  D6 00123          INCL    EXT_LINE_LENGTH                 ; 1745
                                      F4  A6  D6 00126          INCL    INT_LINE_LENGTH                 ; 1746
                          50              6B  D0 00129          MOVL    CHRSIZ, R0                      ; 1749
                          07              51  E9 0012C          BLBC    R1, 15$                         ;
                          50        0080  C0  D0 0012F          MOVL    128(R0), R0                     ;
                                          05  11 00134          BRB     16$                             ;
                          50        00B0  C0  D0 00136 15$:     MOVL    176(R0), R0                     ;
                       FC A6              50  C0 0013B 16$:     ADDL2   R0, TMS_LINE_LENGTH             ;
                                          52  D5 0013F 17$:     TSTL    XMPTR                           ; 1751
                                          26  13 00141          BEQL    20$                             ;
                                          7E  D4 00143          CLRL    -(SP)                           ; 1753
                                          18  BB 00145          PUSHR   #^M<R3,R4>                      ;
                                          55  DD 00147          PUSHL   M_INDENT                        ;
                                        04 A2  DD 00149          PUSHL   4(XMPTR)                        ;
                00000000V EF              05  FB 0014C          CALLS   #5, BOOK_REF                    ;
                          03        FA A7   B1 00153          CMPW    CMDBLK+6, #3                    ; 1754
                                          03  12 00157          BNEQ    18$                             ;
                          54              01  D0 00159          MOVL    #1, NEW_LINE                    ;
                                          62  D5 0015C 18$:     TSTL    (XMPTR)                         ; 1756
                                          05  13 0015E          BEQL    19$                             ;
                          52              62  D0 00160          MOVL    (XMPTR), XMPTR                  ; 1758
                                          DA  11 00163          BRB     17$                             ;
                                          52  D4 00165 19$:     CLRL    XMPTR                           ; 1760
                                          D6  11 00167          BRB     17$                             ; 1751
                          0B              66  D1 00169 20$:     CMPL    LINE_TYPE, #11                  ; 1766
                                          0F  13 0016C          BEQL    23$                             ;
                                          68  D5 0016E          TSTL    INDLVL                          ;
                                          05  12 00170          BNEQ    21$                             ;
                          50              07  D0 00172          MOVL    #7, R0                          ;
                                          03  11 00175          BRB     22$                             ;
                          50              0A  D0 00177 21$:     MOVL    #10, R0                         ;
                          66              50  D0 0017A 22$:     MOVL    R0, LINE_TYPE                   ;
                                        F8 A6  DD 0017D 23$:     PUSHL   EXT_LINE_LENGTH                 ; 1768
                                    FB44  C6  9F 00180          PUSHAB  RNO_LINE                        ;
                                      F4  A6  DD 00184          PUSHL   INT_LINE_LENGTH                 ;
                             00000000V  EF    03  FB 00187          CALLS   #3, INS_LINE                    ;
                                              04  0018E          RET                                     ; 1770

; Routine Size:  399 bytes,    Routine Base:  $CODE$ + 04F7
```

```
;  997     1771  1 %SBTTL 'INS_PAGE_NO -- generate page # list'
;  998     1772  1 ROUTINE INS_PAGE_NO (LIST_HEAD) : NOVALUE =
;  999     1773  1
; 1000     1774  1 !++
; 1001     1775  1 ! FUNCTIONAL DESCRIPTION:
; 1002     1776  1 !
; 1003     1777  1 !       Generate the page number list for the index item
; 1004     1778  1 !
; 1005     1779  1 ! FORMAL PARAMETERS:
; 1006     1780  1 !
; 1007     1781  1 !       LIST_HEAD - address of page number list
; 1008     1782  1 !
; 1009     1783  1 ! IMPLICIT INPUTS:
; 1010     1784  1 !
; 1011     1785  1 !       CMDBLK          - command line information block
; 1012     1786  1 !       INDLVL          - subindex level
; 1013     1787  1 !       LSTSTK          - subindex list stack
; 1014     1788  1 !       PAGE_WIDTH      - size of output page
; 1015     1789  1 !       CHRSIZ          - TMS character size vector
; 1016     1790  1 !
; 1017     1791  1 ! IMPLICIT OUTPUTS:
; 1018     1792  1 !
; 1019     1793  1 !       LINE_POINTER    - pointer to next position in output line
; 1020     1794  1 !       EXT_LINE_LENGTH - external line length
; 1021     1795  1 !       INT_LINE_LENGTH - internal line length
; 1022     1796  1 !       TMS_LINE_LENGTH - TMS line length in relative units
; 1023     1797  1 !
; 1024     1798  1 ! ROUTINE VALUE:
; 1025     1799  1 ! COMPLETION CODES:
; 1026     1800  1 !
; 1027     1801  1 !       None
; 1028     1802  1 !
; 1029     1803  1 ! SIDE EFFECTS:
; 1030     1804  1 !
; 1031     1805  1 !       None
; 1032     1806  1 !
; 1033     1807  1 !--
; 1034     1808  1
; 1035     1809  2     BEGIN
; 1036     1810  2
; 1037     1811  2     LOCAL
; 1038     1812  2         INDENT,
; 1039     1813  2         MAX_EXT_LENGTH,
; 1040     1814  2         REF_GENERATED,
; 1041     1815  2         MORE,
; 1042     1816  2         NEW_LINE,
; 1043     1817  2         CUR_BOOK,
; 1044     1818  2         XXPTR : REF $XX_BLOCK;
; 1045     1819  2
; 1046     1820  2     XXPTR = .LIST_HEAD;
; 1047     1821  2
; 1048     1822  2     NEW_LINE = FALSE;
; 1049     1823  2     CUR_BOOK = 0;
; 1050     1824  2
; 1051     1825  2     IF .CMDBLK [NDX$H_LAYOUT] EQL SEPARATE
; 1052     1826  2     THEN
; 1053     1827  3         BEGIN
```

```
;  1054     1828  3          !
;  1055     1829  3          ! Doing an index with separate references
;  1056     1830  3          !
;  1057     1831  3          INDENT = .CMDBLK [NDX$G_COLUMN_WID] + .CMDBLK [NDX$G_GUTTER_WID];
;  1058     1832  3          MAX_EXT_LENGTH = .PAGE_WIDTH;
;  1059     1833  3          END
;  1060     1834  2      ELSE
;  1061     1835  3          BEGIN
;  1062     1836  3          INDENT = 2 * .INDLVL;
;  1063     1837  3          MAX_EXT_LENGTH = .CMDBLK [NDX$G_COLUMN_WID];
;  1064     1838  2          END;
;  1065     1839  2
;  1066     1840  2      DO
;  1067     1841  3          BEGIN
;  1068     1842  3          !
;  1069     1843  3          ! Process all page references
;  1070     1844  3          !
;  1071     1845  3          REF_GENERATED = FALSE;
;  1072     1846  3
;  1073     1847  3          IF .CUR_BOOK NEQ .XXPTR [XX$A_BOOK]
;  1074     1848  3          THEN
;  1075     1849  4              BEGIN
;  1076     1850  4              !
;  1077     1851  4              ! Have encountered a new book name
;  1078     1852  4              !
;  1079     1853  4              ! NOTE: If you want to insert a comma after the book reference,
;  1080     1854  4              ! use .CMDBLK [NDX$V_PAGES] as the last argument to BOOK_REF
;  1081     1855  4              !
;  1082     1856  4              CUR_BOOK = .XXPTR [XX$A_BOOK];
;  1083     1857  4              BOOK_REF (.CUR_BOOK, .INDENT, .MAX_EXT_LENGTH, .NEW_LINE, FALSE);
;  1084     1858  4
;  1085     1859  4              IF .CMDBLK [NDX$H_LAYOUT] EQL SEPARATE THEN NEW_LINE = TRUE;
;  1086     1860  4              REF_GENERATED = TRUE;
;  1087     1861  3              END;
;  1088     1862  3
;  1089     1863  3          IF .CMDBLK [NDX$V_PAGES]
;  1090     1864  3          THEN
;  1091     1865  4              BEGIN
;  1092     1866  4              !
;  1093     1867  4              ! Doing page references. Generate one.
;  1094     1868  4              !
;  1095     1869  4              SELECTONE TRUE OF
;  1096     1870  4              SET
;  1097     1871  4
;  1098     1872  4              [.XXPTR [XX$V_END]]:
;  1099     1873  4                  !
;  1100     1874  4                  ! END of page range.
;  1101     1875  4                  ! These are picked up by BEGIN's so no processing is required
;  1102     1876  4                  !
;  1103     1877  4                  ;
;  1104     1878  4
;  1105     1879  4              [.XXPTR [XX$V_BEGIN]]:
;  1106     1880  5                  BEGIN
;  1107     1881  5                  !
;  1108     1882  5                  ! BEGIN page range.
;  1109     1883  5                  !
;  1110     1884  5                  LOCAL
```

```
;  1111          1885  5                       SKIPPED_ENTRIES,
;  1112          1886  5                       XXEND : REF $XX_BLOCK;
;  1113          1887  5
;  1114          1888  5                   SKIPPED_ENTRIES = 0;
;  1115          1889  5                   PAGE_REF (.XXPTR, .INDENT, .MAX_EXT_LENGTH);    ! Write initial page
;  1116          1890  5                   REF_GENERATED = TRUE;
;  1117          1891  5
;  1118          1892  5                   !
;  1119          1893  5                   ! Find end of range and output
;  1120          1894  5                   !
;  1121          1895  5                   XXEND = .XXPTR;
;  1122          1896  5                   WHILE TRUE DO
;  1123          1897  6                       BEGIN
;  1124          1898  6
;  1125          1899  6                       IF .XXEND [XX$A_LINK] EQL 0
;  1126          1900  6                       THEN
;  1127          1901  7                           BEGIN
;  1128          1902  7                           !
;  1129          1903  7                           ! No corresponding END.
;  1130          1904  7                           ! Issue error message and exit loop.
;  1131          1905  7                           !
;  1132        L 1906  7 %IF %BLISS (BLISS32)
;  1133          1907  7 %THEN                                           ! Signal errors in BLISS32
;  1134          1908  7
;  1135          1909  7                           SIGNAL (INDEX$_NOEND);
;  1136          1910  7
;  1137        U 1911  7 %ELSE                                           ! Use $XPO_PUT_MSG otherwise
;  1138        U 1912  7
;  1139        U 1913  7                           $XPO_PUT_MSG (SEVERITY = WARNING,
;  1140        U 1914  7                               STRING = '.XPLUS (BEGIN) has no corresponding .XPLUS (END)');
;  1141        U 1915  7
;  1142          1916  7 %FI
;  1143          1917  7
;  1144          1918  7                           ENTMSG ();
;  1145          1919  7
;  1146          1920  7                           EXITLOOP;
;  1147          1921  6                           END;
;  1148          1922  6
;  1149          1923  6                   !
;  1150          1924  6                   ! Point to next entry
;  1151          1925  6                   !
;  1152          1926  6                   XXEND = .XXEND [XX$A_LINK];
;  1153          1927  6
;  1154          1928  6                   IF .XXEND [XX$V_END]
;  1155          1929  7                   AND (.XXEND [XX$A_BOOK] EQL .XXPTR [XX$A_BOOK])
;  1156          1930  6                   THEN
;  1157          1931  7                       BEGIN
;  1158          1932  7                       !
;  1159          1933  7                       ! Entry is an end range entry and the book identifiers
;  1160          1934  7                       ! are the same; found the corresponding END
;  1161          1935  7                       !
;  1162          1936  7                       ! If BEGIN and END are on same page and have identical
;  1163          1937  7                       ! display attributes, then don't generate a range.
;  1164          1938  7                       !
;  1165          1939  7                       XXPTR [XX$V_BEGIN] = FALSE;
;  1166          1940  7                       XXEND [XX$V_END] = FALSE;
;  1167          1941  7
```

```
;  1168          1942  7                          IF NOT SAMEPG (.XXPTR, .XXEND)
;  1169          1943  7                          THEN
;  1170          1944  8                              BEGIN
;  1171          1945  8                              !
;  1172          1946  8                              ! Not same page or display attributes not identical.
;  1173          1947  8                              ! Generate page reference.
;  1174          1948  8                              !
;  1175          1949  8                              ! Append ' to '
;  1176          1950  8                              !
;  1177          1951  8                              FILL_LINE (2, CH$PTR (UPLIT ('to')), .INDENT,
;  1178          1952  8                                  .MAX_EXT_LENGTH, FALSE, FALSE, TRUE);
;  1179          1953  8                              !
;  1180          1954  8                              ! Insert end of range number
;  1181          1955  8                              !
;  1182          1956  8                              PAGE_REF (.XXEND, .INDENT, .MAX_EXT_LENGTH);
;  1183          1957  8                              END;
;  1184          1958  7                          !
;  1185          1959  7                          ! Reset BEGIN and END attributes.
;  1186          1960  7                          !
;  1187          1961  7                          XXPTR [XX$V_BEGIN] = TRUE;
;  1188          1962  7                          XXEND [XX$V_END] = TRUE;
;  1189          1963  7
;  1190          1964  7                          EXITLOOP;                       ! Exit loop
;  1191          1965  7                          END
;  1192          1966  7                      ELSE
;  1193          1967  7                          SKIPPED_ENTRIES = .SKIPPED_ENTRIES + 1;
;  1194          1968  6                      END;
;  1195          1969  6
;  1196          1970  5                  IF .SKIPPED_ENTRIES NEQ 0
;  1197          1971  5                  THEN
;  1198          1972  5                      BEGIN
;  1199          1973  5                      !
;  1200          1974  6                      ! There were entries inside a page range that were skipped.
;  1201          1975  6                      ! Tell the user about it
;  1202          1976  6                      !
;  1203          1977  6
;  1204          1978  6
;  1205        L 1979  6 %IF %BLISS (BLISS32)
;  1206          1980  6 %THEN                                       ! Signal errors in BLISS32
;  1207          1981  6
;  1208          1982  6                      SIGNAL (INDEX$_SKIPPED, 1, .SKIPPED_ENTRIES);
;  1209          1983  6
;  1210        U 1984  6 %ELSE                                       ! Use $XPO_PUT_MSG otherwise
;  1211        U 1985  6
;  1212        U 1986  6                      $XPO_PUT_MSG (SEVERITY = WARNING,
;  1213        U 1987  6                          STRING = $STR_CONCAT ($STR_ASCII (.SKIPPED_ENTRIES),
;  1214        U 1988  6                              ' reference(s) inside page range - ignored'));
;  1215        U 1989  6
;  1216          1990  6 %FI
;  1217          1991  6
;  1218          1992  6                      ENTMSG ();                      ! Dump the entry too.
;  1219          1993  5                      END;
;  1220          1994  5
;  1221          1995  5                  !
;  1222          1996  5                  ! NOTE: to avoid skipping entries inside a page range, remove
;  1223          1997  5                  ! the following line.
;  1224          1998  5                  !
```

```
;  1225     1999  5                XXPTR = .XXEND;
;  1226     2000  4                END;
;  1227     2001  4
;  1228     2002  4            [OTHERWISE]:
;  1229     2003  5                BEGIN
;  1230     2004  5                !
;  1231     2005  5                ! Normal page ref
;  1232     2006  5                !
;  1233     2007  5                LOCAL
;  1234     2008  5                    XXPEEK : REF $XX_BLOCK;
;  1235     2009  5
;  1236     2010  5                PAGE_REF (.XXPTR, .INDENT, .MAX_EXT_LENGTH); ! Insert reference into line
;  1237     2011  5                REF_GENERATED = TRUE;
;  1238     2012  5
;  1239     2013  5                XXPEEK = .XXPTR [XX$A_LINK];    ! Point to next entry
;  1240     2014  5                WHILE .XXPEEK NEQ 0 DO
;  1241     2015  6                    BEGIN
;  1242     2016  6                    !
;  1243     2017  6                    ! Check for duplicate references
;  1244     2018  6                    !
;  1245     2019  6                    IF SAMEPG (.XXPTR, .XXPEEK)
;  1246     2020  6                    THEN
;  1247     2021  6                        !
;  1248     2022  6                        ! Duplicate reference - skip it
   1249     2023  6                        !
;  1250     2024  6                        XXPTR = .XXPEEK
;  1251     2025  6                    ELSE
;  1252     2026  6                        !
;  1253     2027  6                        ! Not same, terminate scan
;  1254     2028  6                        !
;  1255     2029  6                        EXITLOOP;
;  1256     2030  6
;  1257     2031  6                    XXPEEK = .XXPEEK [XX$A_LINK];
;  1258     2032  5                    END;
;  1259     2033  5
;  1260     2034  5            IF .CMDBLK [NDX$V_PAGE_MERGE]
;  1261     2035  5            THEN
;  1262     2036  6                BEGIN
;  1263     2037  6                !
;  1264     2038  6                ! User selected old style page merging
;  1265     2039  6                ! (adjacent pages form a page range)
;  1266     2040  6                !
;  1267     2041  6                LOCAL
;  1268     2042  6                    RANGE_SEEN;
;  1269     2043  6
;  1270     2044  6                RANGE_SEEN = FALSE;
;  1271     2045  6                XXPEEK = .XXPTR [XX$A_LINK];
;  1272     2046  6                WHILE .XXPEEK NEQ 0 DO
;  1273     2047  7                    BEGIN
;  1274     2048  7
;  1275     2049  7                    IF  PAGMRG (XTNPAG (.XXPTR [XX$H_PAGE]), XTNPAG (.XXPEEK [XX$H_PAGE]))
;  1276     2050  7                    AND SAME_ATTR (.XXPTR, .XXPEEK)
;  1277     2051  7                    THEN
;  1278     2052  8                        BEGIN
;  1279     2053  8                        !
;  1280     2054  8                        ! Found some adjacent pages with same display attributes
;  1281     2055  8                        !
```

```
; 1282          2056  8                             RANGE_SEEN = TRUE;
; 1283          2057  8                             XXPTR = .XXPEEK;
; 1284          2058  8                             END
; 1285          2059  7                         ELSE
; 1286          2060  7                             EXITLOOP;
; 1287          2061  7
; 1288          2062  7                         XXPEEK = .XXPEEK [XX$A_LINK];
; 1289          2063  6                         END;
; 1290          2064  6
; 1291          2065  6                     IF .RANGE_SEEN
; 1292          2066  6                     THEN
; 1293          2067  7                         BEGIN
; 1294          2068  7                         !
; 1295          2069  7                         ! We found a page range - write it out
; 1296          2070  7                         !
; 1297          2071  7                         FILL_LINE (2, CH$PTR (UPLIT ('to')), .INDENT, .MAX_EXT_LENGTH, FALSE, FALSE, TRUE);
; 1298          2072  7                         PAGE_REF (.XXPTR, .INDENT, .MAX_EXT_LENGTH);
; 1299          2073  6                         END;
; 1300          2074  5                     END;
; 1301          2075  4                 END;
; 1302          2076  4
; 1303          2077  4             TES;
; 1304          2078  3             END;
; 1305          2079  3
; 1306          2080  3             !
; 1307          2081  3             ! Check for more to do
; 1308          2082  3             !
; 1309          2083  3             MORE = FALSE;
; 1310          2084  3
; 1311          2085  3             XXPTR = .XXPTR [XX$A_LINK];
; 1312          2086  3             WHILE .XXPTR NEQ 0 DO
; 1313          2087  4                 BEGIN
; 1314          2088  4
; 1315          2089  4                 IF NOT .XXPTR [XX$V_END]
; 1316          2090  4                 THEN
; 1317          2091  5                     BEGIN
; 1318          2092  5                     !
; 1319          2093  5                     ! Not a END range
; 1320          2094  5                     !
; 1321          2095  5                     MORE = TRUE;
; 1322          2096  5                     EXITLOOP;
; 1323          2097  4                     END;
; 1324          2098  4
; 1325          2099  4                 XXPTR = .XXPTR [XX$A_LINK];
; 1326          2100  3                 END;
; 1327          2101  3
; 1328          2102  3             IF .MORE                                    ! More to do
; 1329          2103  3             AND .CMDBLK [NDX$V_PAGES]                   ! Generating page references
; 1330          2104  3             AND .REF_GENERATED                          ! Generated a reference
; 1331          2105  4             AND (
; 1332          2106  5                 (.CMDBLK [NDX$H_LAYOUT] NEQ SEPARATE)   ! Not an index with separate references
; 1333          2107  5                 OR (.CUR_BOOK EQL .XXPTR [XX$A_BOOK])   ! or next book is the same as the current one
; 1334          2108  4                 )
; 1335          2109  3             THEN
; 1336          2110  4                 BEGIN
; 1337          2111  4                 !
; 1338          2112  4                 ! Append a comma
```

```
;  1339     2113  4              !
;  1340     2114  4              CH$WCHAR_A (%C',', LINE_POINTER);
;  1341     2115  4              INT_LINE_LENGTH = .INT_LINE_LENGTH + 1;
;  1342     2116  4              EXT_LINE_LENGTH = .EXT_LINE_LENGTH + 1;
;  1343     2117  4              TMS_LINE_LENGTH = .TMS_LINE_LENGTH + .CHRSIZ [%C','];
;  1344     2118  3              END;
;  1345     2119  3
;  1346     2120  3          END
;  1347     2121  3
;  1348     2122  2      WHILE .MORE;                ! Loop if more to do
;  1349     2123  2
;  1350     2124  1      END;                        !End of INS_PAGE_NO


                                                    .PSECT  $PLIT$,NOWRT,NOEXE,2

                                00  00  6F  74  003C4 P.ADQ:  .ASCII  \to\<0><0>                                    ;
                                00  00  6F  74  003C8 P.ADR:  .ASCII  \to\<0><0>                                    ;


                                                    .PSECT  $CODE$,NOWRT,2

                                          0FFC 00000 INS_PAGE_NO:
                                                    .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11                    ; 1772
                                5B 00000000G    EF  9E 00002        MOVAB   CMDBLK+4, R11                           ;
                                53        04    AC  D0 00009        MOVL    LIST_HEAD, XXPTR                        ; 1820
                                                59  D4 0000D        CLRL    NEW_LINE                                ; 1822
                                                57  D4 0000F        CLRL    CUR_BOOK                                ; 1823
                                03        02    AB  B1 00011        CMPW    CMDBLK+6, #3                            ; 1825
                                                0F  12 00015        BNEQ    1$                                      ;
                          56        08    AB        0C  AB  C1 00017      ADDL3   CMDBLK+16, CMDBLK+12, INDENT     ; 1831
                                55 00000000'    EF  D0 0001D        MOVL    PAGE_WIDTH, MAX_EXT_LENGTH              ; 1832
                                                0C  11 00024        BRB     2$                                      ; 1825
                          56 00000000G      EF      01  78 00026 1$:  ASHL    #1, INDLVL, INDENT                      ; 1836
                                55        08    AB  D0 0002E        MOVL    CMDBLK+12, MAX_EXT_LENGTH               ; 1837
                                                58  D4 00032 2$:  CLRL    REF_GENERATED                           ; 1845
                                  0C    A3        57  D1 00034        CMPL    CUR_BOOK, 12(XXPTR)                     ; 1847
                                                21  13 00038        BEQL    4$                                      ;
                                57        0C    A3  D0 0003A        MOVL    12(XXPTR), CUR_BOOK                     ; 1856
                                                7E  D4 0003E        CLRL    -(SP)                                   ; 1857
                                        0220    8F  BB 00040        PUSHR   #^M<R5,R9>                              ;
                                                56  DD 00044        PUSHL   INDENT                                  ;
                                                57  DD 00046        PUSHL   CUR_BOOK                                ;
                             00000000V      EF      05  FB 00048        CALLS   #5, BOOK_REF                        ;
                                03        02    AB  B1 0004F        CMPW    CMDBLK+6, #3                            ; 1859
                                                03  12 00053        BNEQ    3$                                      ;
                                59              01  D0 00055        MOVL    #1, NEW_LINE                            ;
                                58              01  D0 00058 3$:  MOVL    #1, REF_GENERATED                       ; 1860
                          03        FC    AB        03  E0 0005B 4$:  BBS     #3, CMDBLK, 6$                          ; 1863
                                                31  00060 5$:  BRW     19$                                     ;
                          F8        0A    A3        03  E0 00063 6$:  BBS     #3, 10(XXPTR), 5$                       ; 1872
                          03        0A    A3        02  E0 00068        BBS     #2, 10(XXPTR), 7$                       ; 1879
                                                00A8  31 0006D        BRW     14$                                     ;
                                                54  D4 00070 7$:  CLRL    SKIPPED_ENTRIES                         ; 1888
                                                55  DD 00072        PUSHL   MAX_EXT_LENGTH                          ; 1889
```

```
                                 0048   8F  BB 00074          PUSHR   #^M<R3,R6>                                      ;
                     00000000V   EF     03  FB 00078          CALLS   #3, PAGE_REF                                    ;
                                 58     01  D0 0007F          MOVL    #1, REF_GENERATED                               ; 1890
                                 52     53  D0 00082          MOVL    XXPTR, XXEND                                    ; 1895
                                        62  D5 00085 8$:      TSTL    (XXEND)                                         ; 1899
                                        16  12 00087          BNEQ    9$                                              ;
                           00000000G    8F  DD 00089          PUSHL   #DSRINDEX$_NOEND                                ; 1909
                     00000000G   00     01  FB 0008F          CALLS   #1, LIB$SIGNAL                                  ;
                     00000000G   EF     00  FB 00096          CALLS   #0, ENTMSG                                      ; 1918
                                        57  11 0009D          BRB     12$                                             ; 1901
                                 52     62  D0 0009F 9$:      MOVL    (XXEND), XXEND                                  ; 1926
                           4B     0A     03  E1 000A2          BBC     #3, 10(XXEND), 11$                              ; 1928
                                  0C     A2  D1 000A7          CMPL    12(XXEND), 12(XXPTR)                            ; 1929
                                         44  12 000AC          BNEQ    11$                                             ;
                                  0A     A3  04 8A 000AE       BICB2   #4, 10(XXPTR)                                   ; 1939
                                  0A     A2  08 8A 000B2       BICB2   #8, 10(XXEND)                                   ; 1940
                                             52 DD 000B6        PUSHL   XXEND                                           ; 1942
                                             53 DD 000B8        PUSHL   XXPTR                                           ;
                     00000000V   EF         02 FB 000BA        CALLS   #2, SAMEPG                                      ;
                                 24         50 E8 000C1        BLBS    R0, 10$                                         ;
                                            01 DD 000C4        PUSHL   #1                                              ; 1951
                                            7E 7C 000C6        CLRQ    -(SP)                                           ;
                                            55 DD 000C8        PUSHL   MAX_EXT_LENGTH                                  ; 1952
                                            56 DD 000CA        PUSHL   INDENT                                          ; 1951
                           00000000'    EF  9F 000CC          PUSHAB  P.ADQ                                           ;
                                            02 DD 000D2        PUSHL   #2                                              ;
                     00000000V   EF         07 FB 000D4        CALLS   #7, FILL_LINE                                   ;
                                            55 DD 000DB        PUSHL   MAX_EXT_LENGTH                                  ; 1957
                                 0044       8F BB 000DD        PUSHR   #^M<R2,R6>                                      ;
                     00000000V   EF         03 FB 000E1        CALLS   #3, PAGE_REF                                    ;
                           0A    A3         04 88 000E8 10$:   BISB2   #4, 10(XXPTR)                                   ; 1963
                           0A    A2         08 88 000EC        BISB2   #8, 10(XXEND)                                   ; 1964
                                            04 11 000F0        BRB     12$                                             ; 1931
                                            54 D6 000F2 11$:   INCL    SKIPPED_ENTRIES                                 ; 1969
                                            8F 11 000F4        BRB     8$                                              ; 1896
                                            54 D5 000F6 12$:   TSTL    SKIPPED_ENTRIES                                 ; 1972
                                            18 13 000F8        BEQL    13$                                             ;
                                            54 DD 000FA        PUSHL   SKIPPED_ENTRIES                                 ; 1982
                                            01 DD 000FC        PUSHL   #1                                              ;
                                 00000000G  8F DD 000FE        PUSHL   #DSRINDEX$_SKIPPED                              ;
                     00000000G   00         03 FB 00104        CALLS   #3, LIB$SIGNAL                                  ;
                     00000000G   EF         00 FB 0010B        CALLS   #0, ENTMSG                                      ; 1992
                                 53         52 D0 00112 13$:   MOVL    XXEND, XXPTR                                    ; 1999
                                       009B 31 00115          BRW     19$                                             ; 1869
                                            55 DD 00118 14$:   PUSHL   MAX_EXT_LENGTH                                  ; 2010
                                       0048 8F BB 0011A        PUSHR   #^M<R3,R6>                                      ;
                     00000000V   EF         03 FB 0011E        CALLS   #3, PAGE_REF                                    ;
                                 58         01 D0 00125        MOVL    #1, REF_GENERATED                               ; 2011
                                 52         63 D0 00128        MOVL    (XXPTR), XXPEEK                                 ; 2013
                                            16 13 0012B 15$:   BEQL    16$                                             ; 2014
                                            52 DD 0012D        PUSHL   XXPEEK                                          ; 2019
                                            53 DD 0012F        PUSHL   XXPTR                                           ;
                     00000000V   EF         02 FB 00131        CALLS   #2, SAMEPG                                      ;
                                 08         50 E9 00138        BLBC    R0, 16$                                         ;
                                 53         52 D0 0013B        MOVL    XXPEEK, XXPTR                                   ; 2024
                                 52         62 D0 0013E        MOVL    (XXPEEK), XXPEEK                                ; 2031
                                            E8 11 00141        BRB     15$                                             ; 2014
```

```
                6B      FD   AB                 03  E1 00143 16$:    BBC     #3, CMDBLK+1, 19$                    ; 2034
                                                54  D4 00148         CLRL    RANGE_SEEN                           ; 2044
                             52                 63  D0 0014A         MOVL    (XXPTR), XXPEEK                      ; 2045
                                                3D  13 0014D 17$:    BEQL    18$                                  ; 2046
                             7E          08     A2  32 0014F         CVTWL   8(XXPEEK), -(SP)                     ; 2049
                00000000G    EF                 01  FB 00153         CALLS   #1, XTNPAG
                                                50  DD 0015A         PUSHL   R0
                             7E          08     A3  32 0015C         CVTWL   8(XXPTR), -(SP)
                00000000G    EF                 01  FB 00160         CALLS   #1, XTNPAG
                                                50  DD 00167         PUSHL   R0
                00000000G    EF                 02  FB 00169         CALLS   #2, PAGMRG
                             19                 50  E9 00170         BLBC    R0, 18$
                                                52  DD 00173         PUSHL   XXPEEK                               ; 2050
                                                53  DD 00175         PUSHL   XXPTR
                00000000V    EF                 02  FB 00177         CALLS   #2, SAME_ATTR
                             0B                 50  E9 0017E         BLBC    R0, 18$
                             54                 01  D0 00181         MOVL    #1, RANGE_SEEN                       ; 2056
                             53                 52  D0 00184         MOVL    XXPEEK, XXPTR                        ; 2057
                             52                 62  D0 00187         MOVL    (XXPEEK), XXPEEK                     ; 2062
                                                C1  11 0018A         BRB     17$                                  ; 2046
                             24                 54  E9 0018C 18$:    BLBC    RANGE_SEEN, 19$                      ; 2065
                                                01  DD 0018F         PUSHL   #1                                   ; 2071
                                                7E  7C 00191         CLRQ    -(SP)
                                                55  DD 00193         PUSHL   MAX_EXT_LENGTH
                                                56  DD 00195         PUSHL   INDENT
                                    00000000'   EF  9F 00197         PUSHAB  P.ADR
                                                02  DD 0019D         PUSHL   #2
                00000000V    EF                 07  FB 0019F         CALLS   #7, FILL_LINE
                                                55  DD 001A6         PUSHL   MAX_EXT_LENGTH                       ; 2072
                                          0048  8F  BB 001A8         PUSHR   #^M<R3,R6>
                00000000V    EF                 03  FB 001AC         CALLS   #3, PAGE_REF
                                                5A  D4 001B3 19$:    CLRL    MORE                                 ; 2083
                             53                 63  D0 001B5 20$:    MOVL    (XXPTR), XXPTR                       ; 2085
                                                08  13 001B8         BEQL    21$                                  ; 2086
                F6      0A   A3                 03  E0 001BA         BBS     #3, 10(XXPTR), 20$                   ; 2089
                             5A                 01  D0 001BF         MOVL    #1, MORE                             ; 2095
                             43                 5A  E9 001C2 21$:    BLBC    MORE, 24$                            ; 2102
                38      FC   AB                 03  E1 001C5         BBC     #3, CMDBLK, 23$                      ; 2103
                             35                 58  E9 001CA         BLBC    REF_GENERATED, 23$                   ; 2104
                             03          02     AB  B1 001CD         CMPW    CMDBLK+6, #3                         ; 2106
                                                06  12 001D1         BNEQ    22$
                         0C  A3                 57  D1 001D3         CMPL    CUR_BOOK, 12(XXPTR)                  ; 2107
                                                29  12 001D7         BNEQ    23$
                00000000'    FF                 2C  90 001D9 22$:    MOVB    #44, @LINE_POINTER                   ; 2114
                                    00000000'   EF  D6 001E0         INCL    LINE_POINTER
                                    00000000'   EF  D6 001E6         INCL    INT_LINE_LENGTH                      ; 2115
                                    00000000'   EF  D6 001EC         INCL    EXT_LINE_LENGTH                      ; 2116
                             50     00000000G   EF  D0 001F2         MOVL    CHRSIZ, R0                           ; 2117
                00000000'    EF          00B0   C0  C0 001F9         ADDL2   176(R0), TMS_LINE_LENGTH
                             03                 5A  E9 00202 23$:    BLBC    MORE, 24$                            ; 2122
                                           FE2A  31  00205         BRW     2$
                                                 04  00208 24$:    RET                                          ; 2124

; Routine Size:  521 bytes,    Routine Base:  $CODE$ + 0686
```

```
;  1352     2125  1 %SBTTL 'BOOK_REF -- Generate a book reference for master index'
;  1353     2126  1 ROUTINE BOOK_REF (BOOK, INDENT, MAX_EXT, NEW_LINE, COMMA) : NOVALUE =
;  1354     2127  1 !++
;  1355     2128  1 !
;  1356     2129  1 ! FUNCTIONAL DESCRIPTION:
;  1357     2130  1 !
;  1358     2131  1 !       This routine inserts a book reference into RNO_LINE
;  1359     2132  1 !
;  1360     2133  1 ! FORMAL PARAMETERS:
;  1361     2134  1 !
;  1362     2135  1 !       BOOK    - Address of book name string
;  1363     2136  1 !       INDENT  - Number of blanks to indent a new line
;  1364     2137  1 !       MAX_EXT - Maximum external line length
;  1365     2138  1 !       NEW_LINE- Put book reference on a new line
;  1366     2139  1 !       COMMA   - Append a comma to the book name
;  1367     2140  1 !
;  1368     2141  1 ! IMPLICIT INPUTS:
;  1369     2142  1 !
;  1370     2143  1 !       LINE_POINTER    - Pointer to RNO_LINE
;  1371     2144  1 !       EXT_LINE_LENGTH - External line length
;  1372     2145  1 !       INT_LINE_LENGTH - Internal line length
;  1373     2146  1 !       INDLVL          - Current index level
;  1374     2147  1 !       CMDBLK          - Command line information block
;  1375     2148  1 !       PAGE_WIDTH      - Output page width
;  1376     2149  1 !       TMSSTD          - Standard character size for TMS
;  1377     2150  1 !
;  1378     2151  1 ! IMPLICIT OUTPUTS:
;  1379     2152  1 !
;  1380     2153  1 !       LINE_POINTER    - pointer to next position in output line
;  1381     2154  1 !       EXT_LINE_LENGTH - external line length
;  1382     2155  1 !       INT_LINE_LENGTH - internal line length
;  1383     2156  1 !       TMS_LINE_LENGTH - length of line in TMS relative units
;  1384     2157  1 !
;  1385     2158  1 ! ROUTINE VALUE:
;  1386     2159  1 ! COMPLETION CODES:
;  1387     2160  1 !
;  1388     2161  1 !       None
;  1389     2162  1 !
;  1390     2163  1 ! SIDE EFFECTS:
;  1391     2164  1 !
;  1392     2165  1 !       None
;  1393     2166  1 !--
;  1394     2167  2     BEGIN
;  1395     2168  2     LOCAL
;  1396     2169  2         LEADING_BLANK,
;  1397     2170  2         STR_VEC : REF VECTOR,
;  1398     2171  2         PTR,
;  1399     2172  2         LEN;
;  1400     2173  2
;  1401     2174  2     LEADING_BLANK = TRUE;
;  1402     2175  2
;  1403     2176  2     IF .NEW_LINE
;  1404     2177  2     THEN
;  1405     2178  3         BEGIN
;  1406     2179  3         !
;  1407     2180  3         ! Flush current line before inserting reference
;  1408     2181  3         !
```

```
;  1409          2182  3          INS_LINE (.INT_LINE_LENGTH, CH$PTR (RNO_LINE), .EXT_LINE_LENGTH);
;  1410          2183  3          LINE_TYPE = (IF .INDLVL EQL 0 THEN ENTRY_W ELSE SUB_W);
;  1411          2184  3
;  1412          2185  3          INDENT_LINE (.INDENT);
;  1413          2186  3
;  1414          2187  3          LEADING_BLANK = FALSE;
;  1415          2188  3          END
;  1416          2189  2      ELSE
;  1417          2190  2          IF .CMDBLK [NDX$H_LAYOUT] EQL SEPARATE
;  1418          2191  2          THEN
;  1419          2192  3              BEGIN
;  1420          2193  3              !
;  1421          2194  3              ! Separate layout. Pad line to appropriate length.
;  1422          2195  3              ! NOTE: Separate layout is not allowed for TEX format.
;  1423          2196  3              !
;  1424          2197  3              LEN = .CMDBLK [NDX$G_COLUMN_WID] + .CMDBLK [NDX$G_GUTTER_WID] - .EXT_LINE_LENGTH;
;  1425          2198  3
;  1426          2199  3              IF .LEN GTR 0
;  1427          2200  3              THEN
;  1428          2201  4                  BEGIN
;  1429          2202  4                  !
;  1430          2203  4                  ! NOTE: if LEN LEQ 0 then line was already padded or overflowed.
;  1431          2204  4                  ! In any case we don't pad it cause we'll blow up with the
;  1432          2205  4                  ! negative count.
;  1433          2206  4                  !
;  1434          2207  4                  EXT_LINE_LENGTH = .EXT_LINE_LENGTH + .LEN;
;  1435          2208  4
;  1436          2209  5                  IF (.CMDBLK [NDX$H_FORMAT] EQL TMS11_A)
;  1437          2210  5                  OR (.CMDBLK [NDX$H_FORMAT] EQL TMS11_E)
;  1438          2211  4                  THEN
;  1439          2212  5                      BEGIN
;  1440          2213  5                      !
;  1441          2214  5                      ! TMS11 output (line is padded with a tabs)
;  1442          2215  5                      !
;  1443          2216  5                      CH$WCHAR_A (TAB, LINE_POINTER);
;  1444          2217  5                      CH$WCHAR_A (TAB, LINE_POINTER);
;  1445          2218  5                      INT_LINE_LENGTH = .INT_LINE_LENGTH + 2;
;  1446          2219  5                      TMS_LINE_LENGTH = (.CMDBLK [NDX$G_COLUMN_WID] + .CMDBLK [NDX$G_GUTTER_WID]) * TMSSTD;
;  1447          2220  5                      END
;  1448          2221  4                  ELSE
;  1449          2222  5                      BEGIN
;  1450          2223  5                      !
;  1451          2224  5                      ! RUNOFF output (pad line with spaces)
;  1452          2225  5                      !
;  1453          2226  5                      INT_LINE_LENGTH = .INT_LINE_LENGTH + .LEN;
;  1454          2227  5                      CH$FILL (%C' ', .LEN, .LINE_POINTER);
;  1455          2228  5                      LINE_POINTER = CH$PLUS (.LINE_POINTER, .LEN);
;  1456          2229  4                      END;
;  1457          2230  3                  END;
;  1458          2231  3
;  1459          2232  3              LEADING_BLANK = FALSE;
;  1460          2233  2              END;
;  1461          2234  2
;  1462          2235  2      !
;  1463          2236  2      ! Get pointer to book and length of book name
;  1464          2237  2      !
;  1465          2238  2      STR_VEC = .BOOK;
```

```
;  1466          2239  2     LEN = .STR_VEC [0];
;  1467          2240  2     PTR = CH$PTR (STR_VEC [1]);
;  1468          2241  2
;  1469          2242  2     !
;  1470          2243  2     ! Insert book name into line
;  1471          2244  2     !
;  1472          2245  2     FILL_LINE (.LEN, .PTR, .INDENT, .MAX_EXT, FALSE, TRUE, .LEADING_BLANK);
;  1473          2246  2
;  1474          2247  2     !
;  1475          2248  2     ! Append comma to end of reference if necessary
;  1476          2249  2     !
;  1477          2250  2     IF .COMMA
;  1478          2251  2     THEN
;  1479          2252  3         BEGIN
;  1480          2253  3         CH$WCHAR_A (%C',', LINE_POINTER);
;  1481          2254  3         INT_LINE_LENGTH = .INT_LINE_LENGTH + 1;
;  1482          2255  3         EXT_LINE_LENGTH = .EXT_LINE_LENGTH + 1;
;  1483          2256  3         TMS_LINE_LENGTH = .TMS_LINE_LENGTH + .CHRSIZ [%C','];
;  1484          2257  2         END;
;  1485          2258  2
;  1486          2259  1     END;


                                03FC 00000 BOOK_REF:
                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9                    ; 2126
                      59 00000000G   EF  9E 00002          MOVAB   CMDBLK+4, R9                    ;
                      58 00000000'   EF  9E 00009          MOVAB   LINE_POINTER, R8                ;
                      57             01  D0 00010          MOVL    #1, LEADING_BLANK               ; 2174
                      33       10    AC  E9 00013          BLBC    NEW_LINE, 3$                    ; 2176
                               04B8  C8  DD 00017          PUSHL   EXT_LINE_LENGTH                 ; 2182
                                 04  A8  9F 0001B          PUSHAB  RNO_LINE                        ;
                               04B4  C8  DD 0001E          PUSHL   INT_LINE_LENGTH                 ;
                00000000V      EF        03  FB 00021          CALLS   #3, INS_LINE                    ;
                          00000000G  EF  D5 00029          TSTL    INDLVL                          ; 2183
                                     05  12 0002F          BNEQ    1$                              ;
                      50             06  D0 00031          MOVL    #6, R0                          ;
                                     03  11 00034          BRB     2$                              ;
                      50             09  D0 00036  1$:     MOVL    #9, R0                          ;
                   04C0    C8        50  D0 00039  2$:     MOVL    R0, LINE_TYPE                   ;
                               08    AC  DD 0003E          PUSHL   INDENT                          ; 2185
                00000000V      EF        01  FB 00041          CALLS   #1, INDENT_LINE                 ;
                                     51  11 00048          BRB     6$                              ; 2187
                      03       02    A9  B1 0004A  3$:     CMPW    CMDBLK+6, #3                    ; 2190
                                     4D  12 0004E          BNEQ    7$                              ;
50       08    A9       0C    A9  C1 00050          ADDL3   CMDBLK+16, CMDBLK+12, R0                ; 2197
56             50     04B8    C8  C3 00056          SUBL3   EXT_LINE_LENGTH, R0, LEN                ;
                                     3D  15 0005C          BLEQ    6$                              ; 2199
                   04B8    C8        56  C0 0005E          ADDL2   LEN, EXT_LINE_LENGTH            ; 2207
                      02             69  B1 00063          CMPW    CMDBLK+4, #2                    ; 2209
                                     05  13 00066          BEQL    4$                              ;
                      03             69  B1 00068          CMPW    CMDBLK+4, #3                    ; 2210
                                     1F  12 0006B          BNEQ    5$                              ;
                   00    B8       00G  8F  90 0006D  4$:     MOVB    #TAB, @LINE_POINTER             ; 2216
                                     68  D6 00072          INCL    LINE_POINTER                    ;
```

```
                                00    B8        00G   8F   90 00074          MOVB    #TAB, @LINE_POINTER                     ; 2217
                                                      68   D6 00079          INCL    LINE_POINTER                            ;
                              04B4    C8              02   C0 0007B          ADDL2   #2, INT_LINE_LENGTH                     ; 2218
                      04BC    C8              50 00000000G  8F   C5 00080  MULL3   #TMSSTD, R0, TMS_LINE_LENGTH          ; 2219
                                                      0F   11 0008A          BRB     6$                                      ; 2209
                              04B4    C8              56   C0 0008C 5$:     ADDL2   LEN, INT_LINE_LENGTH                    ; 2226
              56              20      6E              00   2C 00091          MOVC5   #0, (SP), #32, LEN, @LINE_POINTER       ; 2227
                                              00      B8        00096          
                                      68              56   C0 00098          ADDL2   LEN, LINE_POINTER                       ; 2228
                                                      57   D4 0009B 6$:     CLRL    LEADING_BLANK                           ; 2232
                                      50      04      AC   D0 0009D 7$:     MOVL    BOOK, STR_VEC                           ; 2238
                                      56              80   D0 000A1          MOVL    (STR_VEC)+, LEN                         ; 2239
                                                      57   DD 000A4          PUSHL   LEADING_BLANK                           ; 2245
                                                      01   DD 000A6          PUSHL   #1                                      ;
                                                      7E   D4 000A8          CLRL    -(SP)                                   ;
                                      7E      08      AC   7D 000AA          MOVQ    INDENT, -(SP)                           ;
                                                      50   DD 000AE          PUSHL   PTR                                     ;
                                                      56   DD 000B0          PUSHL   LEN                                     ;
                          00000000V   EF              07   FB 000B2          CALLS   #7, FILL_LINE                           ;
                                      1C      14      AC   E9 000B9          BLBC    COMMA, 8$                               ; 2250
                                00    B8              2C   90 000BD          MOVB    #44, @LINE_POINTER                      ; 2253
                                                      68   D6 000C1          INCL    LINE_POINTER                            ;
                                              04B4    C8   D6 000C3          INCL    INT_LINE_LENGTH                         ; 2254
                                              04B8    C8   D6 000C7          INCL    EXT_LINE_LENGTH                         ; 2255
                                      50 00000000G    EF   D0 000CB          MOVL    CHRSIZ, R0                              ; 2256
                              04BC    C8      00B0    C0   C0 000D2          ADDL2   176(R0), TMS_LINE_LENGTH                ;
                                                      04   D9 000D9 8$:     RET                                             ; 2259

; Routine Size:  218 bytes,    Routine Base:  $CODE$ + 088F
```

```
;  1488          2260  1 %SBTTL 'PAGE_REF -- Format a page reference'
;  1489          2261  1 ROUTINE PAGE_REF (XX_ENTRY, INDENT, MAX_EXT) : NOVALUE =
;  1490          2262  1 !++
;  1491          2263  1 !
;  1492          2264  1 ! FUNCTIONAL DESCRIPTION:
;  1493          2265  1 !
;  1494          2266  1 !       This routine formats a page reference and inserts it into RNO_WORD
;  1495          2267  1 !
;  1496          2268  1 ! FORMAL PARAMETERS:
;  1497          2269  1 !
;  1498          2270  1 !       XX_ENTRY - Pointer to reference to be formatted
;  1499          2271  1 !       INDENT   - Number of blanks to indent new line if reference doesn't fit
;  1500          2272  1 !       MAX_EXT  - Maximum external length of line before wrapping
;  1501          2273  1 !
;  1502          2274  1 ! IMPLICIT INPUTS:
;  1503          2275  1 !
;  1504          2276  1 !       CMDBLK  - Command line information block
;  1505          2277  1 !
;  1506          2278  1 ! IMPLICIT OUTPUTS:
;  1507          2279  1 !
;  1508          2280  1 !       None
;  1509          2281  1 !
;  1510          2282  1 ! ROUTINE VALUE:
;  1511          2283  1 ! COMPLETION CODES:
;  1512          2284  1 !
;  1513          2285  1 !       None
;  1514          2286  1 !
;  1515          2287  1 ! SIDE EFFECTS:
;  1516          2288  1 !
;  1517          2289  1 !       None
;  1518          2290  1 !--
;  1519          2291  2     BEGIN
;  1520          2292  2     MAP
;  1521          2293  2         XX_ENTRY : REF $XX_BLOCK;
;  1522          2294  2
;  1523          2295  2     LOCAL
;  1524          2296  2         REFBUF    : VECTOR [CH$ALLOCATION (100)],
;  1525          2297  2         PTR,
;  1526          2298  2         LEN,
;  1527          2299  2         PAGE : REF PAGE_DEFINITION;
;  1528          2300  2
;  1529          2301  2     PAGE = XTNPAG (.XX_ENTRY [XX$H_PAGE]);
;  1530          2302  2
;  1531          2303  2     PTR = CH$PTR (REFBUF);
;  1532          2304  2
;  1533          2305  2     IF .CMDBLK [NDX$V_STANDARD_PAGE]
;  1534          2306  2     THEN
;  1535          2307  2         !
;  1536          2308  2         ! User wants traditional page numbers
;  1537          2309  2         !
;  1538          2310  2         LEN = PACPAG (.PAGE, PTR)
;  1539          2311  2     ELSE
;  1540          2312  2         !
;  1541          2313  2         ! User wants running pages
;  1542          2314  2         !
;  1543          2315  2         LEN = PACBAS (.PAGE [SCT_RUN_PAGE], PTR, 10);
;  1544          2316  2
```

```
;  1545          2317  2     IF .XX_ENTRY [XX$A_APPEND] NEQ 0
;  1546          2318  2     THEN
;  1547          2319  3         BEGIN
;  1548          2320  3         !
;  1549          2321  3         ! Have appended text
;  1550          2322  3         !
;  1551          2323  3         LOCAL
;  1552          2324  3             A_VEC : REF VECTOR,
;  1553          2325  3             A_PTR,
;  1554          2326  3             A_LEN;
;  1555          2327  3
;  1556          2328  3         A_VEC = .XX_ENTRY [XX$A_APPEND];
;  1557          2329  3         A_LEN = .A_VEC [0];
;  1558          2330  3         A_PTR = CH$PTR (A_VEC [1]);
;  1559          2331  3
;  1560          2332  3         CH$MOVE (.A_LEN, .A_PTR, .PTR);
;  1561          2333  3         LEN = .LEN + .A_LEN;
;  1562          2334  2         END;
;  1563          2335  2
;  1564          2336  2     !
;  1565          2337  2     ! Insert reference into line
;  1566          2338  2     !
;  1567          2339  2     FILL_LINE (.LEN, CH$PTR (REFBUF), .INDENT, .MAX_EXT, .XX_ENTRY [XX$V_BOLD], .XX_ENTRY [XX$V_UNDERLINE],
;  1568          2340  2
;  1569          2341  1     END;


                                        01FC 00000 PAGE_REF:
                                                            .WORD   Save R2,R3,R4,R5,R6,R7,R8       ; 2261
                                    5E      98  AE  9E 00002        MOVAB   -104(SP), SP
                                    56      04  AC  D0 00006        MOVL    XX_ENTRY, R6            ; 2301
                                    7E      08  A6  32 0000A        CVTWL   8(R6), -(SP)
                            00000000G       EF      01 FB 0000E     CALLS   #1, XTNPAG
                                    52          50  D0 00015        MOVL    R0, PAGE
                                    6E      04  AE  9E 00018        MOVAB   REFBUF, PTR             ; 2303
                        0D  00000000G       EF      05 E1 0001C     BBC     #5, CMDBLK, 1$          ; 2305
                                          4004  8F  BB 00024        PUSHR   #^M<R2,SP>              ; 2310
                            00000000G       EF      02 FB 00028     CALLS   #2, FACPAG
                                                    10 11 0002F     BRB     2$
                                                    0A DD 00031 1$: PUSHL   #10                     ; 2315
                                        04      AE  9F 00033        PUSHAB  PTR
                                    7E      0E  A2  3C 00036        MOVZWL  14(PAGE), -(SP)
                            00000000G       EF      03 FB 0003A     CALLS   #3, PACBAS
                                    58          50  D0 00041 2$:    MOVL    R0, LEN
                                        04      A6  D5 00044        TSTL    4(R6)                   ; 2317
                                                0F  13 00047        BEQL    3$
                                    50      04  A6  D0 00049        MOVL    4(R6), A_VEC            ; 2328
                                    57          80  D0 0004D        MOVL    (A_VEC)+, A_LEN         ; 2329
                        00  BE      60          57  28 00050        MOVC3   A_LEN, (A_PTR), @PTR    ; 2332
                                    58          57  C0 00055        ADDL2   A_LEN, LEN              ; 2333
                                                01  DD 00058 3$:    PUSHL   #1                      ; 2339
                7E      0A  A6      01          01  EF 0005A        EXTZV   #1, #1, 10(R6), -(SP)
                7E      0A  A6      01          00  EF 00060        EXTZV   #0, #1, 10(R6), -(SP)
                                    7E      08  AC  7D 00066        MOVQ    INDENT, -(SP)           ;
```

```
                                    18   AE  9F 0006A          PUSHAB  REFBUF                          ;
                                         58  DD 0006D          PUSHL   LEN                            ;
                         00000000V  EF       07  FB 0006F          CALLS   #7, FILL_LINE              ;
                                             04 00076          RET                                    ; 2341

; Routine Size:  119 bytes,    Routine Base:  $CODE$ + 0969
```

```
;  1571          2342  1 %SBTTL 'SAMEPG -- Compare page references'
;  1572          2343  1 ROUTINE SAMEPG (X1, X2) =
;  1573          2344  1 !++
;  1574          2345  1 !
;  1575          2346  1 ! FUNCTIONAL DESCRIPTION:
;  1576          2347  1 !
;  1577          2348  1 !       This routine compares two page references to see if they are equal.
;  1578          2349  1 !
;  1579          2350  1 ! FORMAL PARAMETERS:
;  1580          2351  1 !
;  1581          2352  1 !       X1 - Pointer to first page reference
;  1582          2353  1 !       X2 - Pointer to second page reference
;  1583          2354  1 !
;  1584          2355  1 ! IMPLICIT INPUTS:
;  1585          2356  1 !
;  1586          2357  1 !       None
;  1587          2358  1 !
;  1588          2359  1 ! IMPLICIT OUTPUTS:
;  1589          2360  1 !
;  1590          2361  1 !       None
;  1591          2362  1 !
;  1592          2363  1 ! ROUTINE VALUE:
;  1593          2364  1 ! COMPLETION CODES:
;  1594          2365  1 !
;  1595          2366  1 !       Returns TRUE if references are equal, FALSE otherwise
;  1596          2367  1 !
;  1597          2368  1 ! SIDE EFFECTS:
;  1598          2369  1 !
;  1599          2370  1 !       None
;  1600          2371  1 !--
;  1601          2372  2     BEGIN
;  1602          2373  2     MAP
;  1603          2374  2         X1 : REF $XX_BLOCK,
;  1604          2375  2         X2 : REF $XX_BLOCK;
;  1605          2376  2
;  1606          2377  2     IF PAGEQL (XTNPAG (.X1 [XX$H_PAGE]), XTNPAG (.X2 [XX$H_PAGE]), TRUE)
;  1607          2378  2     THEN
;  1608          2379  2         !
;  1609          2380  2         ! Pages are identical in the RUNOFF sense.
;  1610          2381  2         ! Compare reference attributes
;  1611          2382  2         !
;  1612          2383  2         RETURN SAME_ATTR (.X1, .X2)
;  1613          2384  2     ELSE
;  1614          2385  2         !
;  1615          2386  2         ! Pages not equal in RUNOFF sense
;  1616          2387  2         !
;  1617          2388  2         RETURN FALSE;
;  1618          2389  2
;  1619          2390  1     END;


                                  001C 00000 SAMEPG: .WORD   Save R2,R3,R4                                    ; 2343
                      54 00000000G  EF   9E 00002        MOVAB   XTNPAG, R4                                   ;
                                    01   DD 00009        PUSHL   #1                                           ; 2377
```

```
                                53        08    AC  D0 0000B          MOVL    X2, R3                                  ;
                                7E        08    A3  32 0000F          CVTWL   8(R3), -(SP)                            ;
                                64              01  FB 00013          CALLS   #1, XFNPAG                              ;
                                                50  DD 00016          PUSHL   R0                                      ;
                                52        04    AC  D0 00018          MOVL    X1, R2                                  ;
                                7E        08    A2  32 0001C          CVTWL   8(R2), -(SP)                            ;
                                64              01  FB 00020          CALLS   #1, XFNPAG                              ;
                                                50  DD 00023          PUSHL   R0                                      ;
                    00000000G   EF              03  FB 00025          CALLS   #3, PAGEQL                              ;
                                0A              50  E9 0002C          BLBC    R0, 1$                                  ;
                                                0C  BB 0002F          PUSHR   #^M<R2,R3>                              ; 2383
                    00000000V   EF              02  FB 00031          CALLS   #2, SAME_ATTR                           ;
                                                    04 00038          RET                                             ; 2388
                                                50  D4 00039 1$:      CLRL    R0                                      ;
                                                    04 0003B          RET                                             ; 2390

; Routine Size:  60 bytes,    Routine Base:  $CODE$ + 09E0
```

```
;  1621          2391  1 %SBTTL 'SAME_ATTR - Check page references for same attributes'
;  1622          2392  1 ROUTINE SAME_ATTR (X1, X2)  =
;  1623          2393  1 !++
;  1624          2394  1 !
;  1625          2395  1 ! FUNCTIONAL DESCRIPTION:
;  1626          2396  1 !
;  1627          2397  1 !       This routine compares the display attributes of two page references
;  1628          2398  1 !
;  1629          2399  1 ! FORMAL PARAMETERS:
;  1630          2400  1 !
;  1631          2401  1 !       X1      - Pointer to first page reference
;  1632          2402  1 !       X2      - Pointer to second page reference
;  1633          2403  1 !
;  1634          2404  1 ! IMPLICIT INPUTS:
;  1635          2405  1 !
;  1636          2406  1 !       None
;  1637          2407  1 !
;  1638          2408  1 ! IMPLICIT OUTPUTS:
;  1639          2409  1 !
;  1640          2410  1 !       None
;  1641          2411  1 !
;  1642          2412  1 ! ROUTINE VALUE:
;  1643          2413  1 ! COMPLETION CODES:
;  1644          2414  1 !
;  1645          2415  1 !       TRUE if references have the same display attributes, FALSE otherwise
;  1646          2416  1 !
;  1647          2417  1 ! SIDE EFFECTS:
;  1648          2418  1 !
;  1649          2419  1 !       None
;  1650          2420  1 !--
;  1651          2421  2     BEGIN
;  1652          2422  2     MAP
;  1653          2423  2         X1 : REF $XX_BLOCK,
;  1654          2424  2         X2 : REF $XX_BLOCK;                                                                    ;
;  1655          2425  2
;  1656          2426  2     SELECTONE TRUE OF
;  1657          2427  2     SET
;  1658          2428  2
;  1659          2429  2     [(.X1 [XX$A_APPEND] NEQ 0) AND (.X2 [XX$A_APPEND] NEQ 0)]:
;  1660          2430  3         BEGIN
;  1661          2431  3         !
;  1662          2432  3         ! Both have appended text. Compare strings
;  1663          2433  3         !
;  1664          2434  3         LOCAL
;  1665          2435  3             STR_VEC : REF VECTOR,
;  1666          2436  3             L1,
;  1667          2437  3             P1,
;  1668          2438  3             L2,
;  1669          2439  3             P2;
;  1670          2440  3
;  1671          2441  3         STR_VEC = .X1 [XX$A_APPEND];
;  1672          2442  3         L1 = .STR_VEC [0];
;  1673          2443  3         P1 = CH$PTR (STR_VEC [1]);
;  1674          2444  3
;  1675          2445  3         STR_VEC = .X2 [XX$A_APPEND];
;  1676          2446  3         L2 = .STR_VEC [0];
;  1677          2447  3         P2 = CH$PTR (STR_VEC [1]);
```

```
;  1678          2448  3
;  1679          2449  3          IF CH$NEQ (.L1, .P1, .L2, .P2) THEN RETURN FALSE;
;  1680          2450  2          END;
;  1681          2451  2
;  1682          2452  2      [(.X1 [XX$A_APPEND] EQL 0) AND (.X2 [XX$A_APPEND] EQL 0)]:
;  1683          2453  2          !
;  1684          2454  2          ! Neither have appended strings. Do nothing.
;  1685          2455  2          !
;  1686          2456  2          ;
;  1687          2457  2
;  1688          2458  2      [OTHERWISE]:
;  1689          2459  2          !
;  1690          2460  2          ! One has appended text and the other doesn't.
;  1691          2461  2          ! References are not identical
;  1692          2462  2          !
;  1693          2463  2          RETURN FALSE;
;  1694          2464  2
;  1695          2465  2      TES;
;  1696          2466  2
;  1697          2467  3      RETURN (
;  1698          2468  4                  (.X1 [XX$V_BOLD]      EQL .X2 [XX$V_BOLD])
;  1699          2469  4          AND     (.X1 [XX$V_UNDERLINE] EQL .X2 [XX$V_UNDERLINE])
;  1700          2470  4          AND     (.X1 [XX$V_BEGIN]     EQL .X2 [XX$V_BEGIN])
;  1701          2471  4          AND     (.X1 [XX$V_END]       EQL .X2 [XX$V_END])
;  1702          2472  4          AND     (.X1 [XX$A_BOOK]      EQL .X2 [XX$A_BOOK])
;  1703          2473  2          );
;  1704          2474  2
;  1705          2475  1      END;


                                007C 00000 SAME_ATTR:
                                           .WORD   Save R2,R3,R4,R5,R6                                    ;  2392
                        54  04  AC  D0 00002       MOVL    X1, R4                                         ;  2429
                        52  04  A4  D0 00006       MOVL    4(R4), R2                                      ;
                                53  D4 0000A       CLRL    R3                                             ;
                                52  D5 0000C       TSTL    R2                                             ;
                                02  13 0000E       BEQL    1$                                             ;
                                53  D6 00010       INCL    R3                                             ;
                        50  08  AC  D0 00012 1$:    MOVL    X2, R0                                         ;
                                51  D4 00016       CLRL    R1                                             ;
                            04  A0  D5 00018       TSTL    4(R0)                                          ;
                                02  13 0001B       BEQL    2$                                             ;
                                51  D6 0001D       INCL    R1                                             ;
                        55      53  D2 0001F 2$:    MCOML   R3, R5                                         ;
                        51      55  CA 00022       BICL2   R5, R1                                         ;
                        01      51  D1 00025       CMPL    R1, #1                                         ;
                                1D  12 00028       BNEQ    3$                                             ;
                        50      52  D0 0002A       MOVL    R2, STR_VEC                                    ;  2441
                        53      60  D0 0002D       MOVL    (STR_VEC), L1                                  ;  2442
                        51  04  A0  9E 00030       MOVAB   4(R0), P1                                      ;  2443
                        52  08  AC  D0 00034       MOVL    X2, R2                                         ;  2445
                        50  04  A2  D0 00038       MOVL    4(R2), STR_VEC                                 ;
                        52      80  D0 0003C       MOVL    (STR_VEC)+, L2                                 ;  2446
                52      00  61      53  2D 0003F   CMPC5   L1, (P1), #0, L2, (P2)                         ;  2449
```

```
                                                  60     00044
                                                  1E  11 00045          BRB     6$
                                                  53  D4 00047 3$:      CLRL    R3                                      2452
                                                  52  D5 00049          TSTL    R2
                                                  02  12 0004B          BNEQ    4$
                                                  53  D6 0004D          INCL    R3
                                  50      08      AC  D0 0004F 4$:      MOVL    X2, R0
                                                  51  D4 00053          CLRL    R1
                                          04      A0  D5 00055          TSTL    4(R0)
                                                  02  12 00058          BNEQ    5$
                                                  51  D6 0005A          INCL    R1
                                  55              53  D2 0005C 5$:      MCOML   R3, R5
                                  51              55  CA 0005F          BICL2   R5, R1
                                  01              51  D1 00062          CMPL    R1, #1
                                                  6A  12 00065 6$:      BNEQ    12$
                                  50      08      AC  D0 00067          MOVL    X2, R0                                  2468
                                                  51  D4 0006B          CLRL    R1
                          52      0A      A0      0A  A4  8D 0006D      XORB3   10(R4), 10(R0), R2
                          04              52  E8 00073          BLBS    R2, 7$
                                                  02  12 00076          BNEQ    7$
                                                  51  D6 00078          INCL    R1
                                                  52  D4 0007A 7$:      CLRL    R2                                      2469
                          53      0A      A0      0A  A4  8D 0007C      XORB3   10(R4), 10(R0), R3
                          04              53  01  E0 00082          BBS     #1, R3, 8$
                                                  02  12 00086          BNEQ    8$
                                                  52  D6 00088          INCL    R2
                                  55              51  D2 0008A 8$:      MCOML   R1, R5
                                  52              55  CA 0008D          BICL2   R5, R2
                                                  51  D4 00090          CLRL    R1                                      2470
                          53      0A      A0      0A  A4  8D 00092      XORB3   10(R4), 10(R0), R3
                          04              53  02  E0 00098          BBS     #2, R3, 9$
                                                  02  12 0009C          BNEQ    9$
                                                  51  D6 0009E          INCL    R1
                                  56              52  D2 000A0 9$:      MCOML   R2, R6
                                  51              56  CA 000A3          BICL2   R6, R1
                                                  52  D4 000A6          CLRL    R2                                      2471
                          53      0A      A0      0A  A4  8D 000A8      XORB3   10(R4), 10(R0), R3
                          04              53  03  E0 000AE          BBS     #3, R3, 10$
                                                  02  12 000B2          BNEQ    10$
                                                  52  D6 000B4          INCL    R2
                                  55              51  D2 000B6 10$:     MCOML   R1, R5
                                  52              55  CA 000B9          BICL2   R5, R2
                                                  51  D4 000BC          CLRL    R1                                      2472
                                  0C      A0      0C  A4  D1 000BE      CMPL    12(R4), 12(R0)
                                                  02  12 000C3          BNEQ    11$
                                                  51  D6 000C5          INCL    R1
                                  53              52  D2 000C7 11$:     MCOML   R2, R3
                                  51              53  CA 000CA          BICL2   R3, R1
                                  50              51  D0 000CD          MOVL    R1, R0                                  2467
                                                  04  000D0            RET
                                                  50  D4 000D1 12$:     CLRL    R0                                      2475
                                                  04  000D3            RET

; Routine Size:  212 bytes,    Routine Base:  $CODE$ + 0A1C
```

```
;  1707          2476  1 %SBTTL 'INS_LINE -- Insert line into page'
;  1708          2477  1 ROUTINE INS_LINE (INT_LEN, PTR, EXT_LEN) : NOVALUE =
;  1709          2478  1 !++
;  1710          2479  1 !
;  1711          2480  1 ! FUNCTIONAL DESCRIPTION:
;  1712          2481  1 !
;  1713          2482  1 !       This routine inserts a line into the print page
;  1714          2483  1 !
;  1715          2484  1 ! FORMAL PARAMETERS:
;  1716          2485  1 !
;  1717          2486  1 !       INT_LEN         - Internal line length
;  1718          2487  1 !       PTR             - CH$PTR to text
;  1719          2488  1 !       EXT_LEN         - External line length
;  1720          2489  1 !
;  1721          2490  1 ! IMPLICIT INPUTS:
;  1722          2491  1 !
;  1723          2492  1 !       ALLOWD          - Maximum number of lines allowed on page
;  1724          2493  1 !       LCOUNT          - Left column line count
;  1725          2494  1 !       RCOUNT          - Right column line count
;  1726          2495  1 !       CMDBLK          - Command line information block
;  1727          2496  1 !       DOING_LAST      - TRUE if LAST_LINE was called to process the last
;  1728          2497  1 !                         line in the column
;  1729          2498  1 !
;  1730          2499  1 ! IMPLICIT OUTPUTS:
;  1731          2500  1 !
;  1732          2501  1 !       LCOUNT          - Incremented if line inserted into left column
;  1733          2502  1 !       LLINES          - If line inserted into left column
;  1734          2503  1 !       LTYPE           - Type of line if inserted into left column
;  1735          2504  1 !
;  1736          2505  1 !       RCOUNT          - Incremented if line inserted into right column
;  1737          2506  1 !       RLINES          - If line inserted into right column
;  1738          2507  1 !       RTYPE           - Type of line if inserted into right column
;  1739          2508  1 !
;  1740          2509  1 ! ROUTINE VALUE:
;  1741          2510  1 ! COMPLETION CODES:
;  1742          2511  1 !
;  1743          2512  1 !       None
;  1744          2513  1 !
;  1745          2514  1 ! SIDE EFFECTS:
;  1746          2515  1 !
;  1747          2516  1 !       None
;  1748          2517  1 !--
;  1749          2518  2     BEGIN
;  1750          2519  2
;  1751          2520  2     SELECTONE .CMDBLK [NDX$H_LAYOUT] OF
;  1752          2521  2         SET
;  1753          2522  2
;  1754          2523  2         [TWO_COLUMN]:
;  1755          2524  2             !
;  1756          2525  2             ! Two column output
;  1757          2526  2             !
;  1758          2527  2             IF .LCOUNT EQL .ALLOWD
;  1759          2528  2             THEN
;  1760          2529  3                 BEGIN
;  1761          2530  3                 !
;  1762          2531  3                 ! Filling right column
;  1763          2532  3                 !
```

```
;  1764     2533  4                    IF (.RCOUNT EQL 0)
;  1765     2534  3                    THEN
;  1766     2535  4                        BEGIN
;  1767     2536  4                        !
;  1768     2537  4                        ! First line in columr
;  1769     2538  4                        !
;  1770     2539  4                        IF CH$EQL (1, .BLANKS, .INT_LEN, .PTR, %C' ') THEN RETURN;
;  1771     2540  4
;  1772     2541  4                        IF .CMDBLK [NDX$V_CONTINUATION]
;  1773     2542  4                        THEN
;  1774     2543  5                            BEGIN
;  1775     2544  5                            !
;  1776     2545  5                            ! Generate continuation heading
;  1777     2546  5                            !
;  1778     2547  5                            DO_CONTINUE (.INT_LEN, .PTR, .EXT_LEN);
;  1779     2548  5                            RETURN;
;  1780     2549  4                            END;
;  1781     2550  3                        END;
;  1782     2551  3
;  1783     2552  3                    IF (.RCOUNT EQL .ALLOWD - 1) AND NOT .DOING_LAST
;  1784     2553  3                    THEN
;  1785     2554  3                        !
;  1786     2555  3                        ! Insert last line in columr
;  1787     2556  3                        !
;  1788     2557  3                        LAST_LINE (.INT_LEN, .PTR, .EXT_LEN)
;  1789     2558  3                    ELSE
;  1790     2559  4                        BEGIN
;  1791     2560  4                        !
;  1792     2561  4                        ! Insert normal line
;  1793     2562  4                        !
;  1794     2563  4                        RCOUNT = .RCOUNT + 1;
;  1795     2564  4                        RTYPE [.RCOUNT] = .LINE_TYPE;
;  1796     2565  4                        PADLIN (.INT_LEN, .PTR, .EXT_LEN, RLINES [.RCOUNT, 0,0,0,0]);
;  1797     2566  4                        SAVE_TELLTALE_HEADING (RLINES [0, 0,0,0,0], RTYPE [0], .RCOUNT);
;  1798     2567  4
;  1799     2568  4                        IF .RCOUNT EQL .ALLOWD THEN PUTPAG (FALSE);
;  1800     2569  3                        END;
;  1801     2570  3                    END
;  1802     2571  2                ELSE
;  1803     2572  3                    BEGIN
;  1804     2573  3                    !
;  1805     2574  3                    ! Left column
;  1806     2575  3                    !
;  1807     2576  4                    IF (.LCOUNT EQL 0)
;  1808     2577  3                    THEN
;  1809     2578  4                        BEGIN
;  1810     2579  4                        !
;  1811     2580  4                        ! First line in left column
;  1812     2581  4                        !
;  1813     2582  4                        IF CH$EQL (1, .BLANKS, .INT_LEN, .PTR, %C' ') THEN RETURN;
;  1814     2583  4
;  1815     2584  4                        IF .CMDBLK [NDX$V_CONTINUATION]
;  1816     2585  4                        THEN
;  1817     2586  5                            BEGIN
;  1818     2587  5                            !
;  1819     2588  5                            ! Generate continuation heading
;  1820     2589  5                            !
```

```
;  1821     2590  5                        DO_CONTINUE (.INT_LEN, .PTR, .EXT_LEN);
;  1822     2591  5                        RETURN;
;  1823     2592  4                        END;
;  1824     2593  3                    END;
;  1825     2594  3
;  1826     2595  3                IF (.LCOUNT EQL .ALLOWD - 1) AND NOT .DOING_LAST
;  1827     2596  3                THEN
;  1828     2597  3                    !
;  1829     2598  3                    ! Insert last line in left column
;  1830     2599  3                    !
;  1831     2600  3                    LAST_LINE (.INT_LEN, .PTR, .EXT_LEN)
;  1832     2601  3                ELSE
;  1833     2602  4                    BEGIN
;  1834     2603  4                    !
;  1835     2604  4                    ! Insert a normal line
;  1836     2605  4                    !
;  1837     2606  4                    LCOUNT = .LCOUNT + 1;
;  1838     2607  4                    LTYPE [.LCOUNT] = .LINE_TYPE;
;  1839     2608  4                    PADLIN (.INT_LEN, .PTR, .EXT_LEN, LLINES [.LCOUNT, 0,0,0,0]);
;  1840     2609  4                    SAVE_TELLTALE_HEADING (LLINES [0, 0,0,0,0], LTYPE [0], .LCOUNT);
;  1841     2610  3                    END;
;  1842     2611  3
;  1843     2612  2                END;
;  1844     2613  2
;  1845     2614  2            [OTHERWISE]:
;  1846     2615  3                BEGIN
;  1847     2616  3                !
;  1848     2617  3                ! One column output
;  1849     2618  3                !
;  1850     2619  3
;  1851     2620  4                IF (.LCOUNT EQL 0)
;  1852     2621  3                THEN
;  1853     2622  4                    BEGIN
;  1854     2623  4                    !
;  1855     2624  4                    ! First line in column
;  1856     2625  4                    !
;  1857     2626  4                    IF CH$EQL (1, .BLANKS, .INT_LEN, .PTR, %C' ') THEN RETURN;
;  1858     2627  4
;  1859     2628  4                    IF .CMDBLK [NDX$V_CONTINUATION]
;  1860     2629  4                    THEN
;  1861     2630  5                        BEGIN
;  1862     2631  5                        !
;  1863     2632  5                        ! Generate continuation heading
;  1864     2633  5                        !
;  1865     2634  5                        DO_CONTINUE (.INT_LEN, .PTR, .EXT_LEN);
;  1866     2635  5                        RETURN;
;  1867     2636  4                        END;
;  1868     2637  3                    END;
;  1869     2638  3
;  1870     2639  3                IF (.LCOUNT EQL .ALLOWD - 1) AND NOT .DOING_LAST
;  1871     2640  3                THEN
;  1872     2641  3                    !
;  1873     2642  3                    ! Insert last line in column
;  1874     2643  3                    !
;  1875     2644  3                    LAST_LINE (.INT_LEN, .PTR, .EXT_LFN)
;  1876     2645  3                ELSE
;  1877     2646  4                    BEGIN
```

```
;  1878     2647  4                    !
;  1879     2648  4                    ! Insert a normal line
;  1880     2649  4                    !
;  1881     2650  4                    LCOUNT = .LCOUNT + 1;
;  1882     2651  4                    LTYPE [.LCOUNT] = .LINE_TYPE;
;  1883     2652  4                    PADLIN (.INT_LEN, .PTR, .EXT_LEN, LLINES [.LCOUNT, 0,0,0,0]);
;  1884     2653  4                    SAVE_TELLTALE_HEADING (LLINES [0, 0,0,0,0], LTYPE [0], .LCOUNT);
;  1885     2654  4
;  1886     2655  4                    IF .LCOUNT EQL .ALLOWD THEN PUTPAG (FALSE);
;  1887     2656  3                    END;
;  1888     2657  2                END;
;  1889     2658  2
;  1890     2659  2            TES;
;  1891     2660  1        END;


                                    0FFC 00000 INS_LINE:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 2477
                                5B 00000000V   EF   9E 00002    MOVAB   PADLIN, R11                         ;
                                5A 00000000G   EF   9E 00009    MOVAB   CMDBLK, R10                         ;
                                59 00000000G   EF   9E 00010    MOVAB   LLINES, R9                          ;
                                58 00000000G   EF   9E 00017    MOVAB   LTYPE, R8                           ;
                                57 00000000G   EF   9E 0001E    MOVAB   RCOUNT, R7                          ;
                                56 00000000G   EF   9E 00025    MOVAB   ALLOWD, R6                          ;
                                55 00000000'   EF   9E 0002C    MOVAB   BLANKS, R5                          ;
                                54 00000000G   EF   9E 00033    MOVAB   LCOUNT, R4                          ;
                                50        06   AA   32 0003A    CVTWL   CMDBLK+6, R0                        ; 2520
                                01           50   B1 0003E    CMPW    R0, #1                              ; 2523
                                             03   13 00041    BEQL    1$                                  ;
                                           00A8   31 00043    BRW     10$                                 ;
                                66           64   D1 00046 1$: CMPL    LCOUNT, ALLOWD                      ; 2527
                                             5C   12 00049    BNEQ    4$                                  ;
                                             67   D5 0004B    TSTL    RCOUNT                              ; 2533
                                             0F   12 0004D    BNEQ    2$                                  ;
04  AC            20        00   B5           01   2D 0004F    CMPC5   #1, @BLANKS, #32, INT_LEN, @PTR     ; 2539
                                       08    BC             00056                                       ;
                                             5A   13 00058    BEQL    5$                                  ;
                              58             6A   06 E0 0005A BBS     #6, CMDBLK, 6$                      ; 2541
                              50             66   01 C3 0005E 2$: SUBL3   #1, ALLOWD, R0                  ; 2552
                                             50   67 D1 00062    CMPL    RCOUNT, R0                          ;
                                             05   12 00065    BNEQ    3$                                  ;
                                          57      04CC   C5   E9 00067    BLBC    DOING_LAST, 8$              ;
                                             67   D6 0006C 3$: INCL    RCOUNT                              ; 2563
                                          50     67   D0 0006E    MOVL    RCOUNT, R0                          ; 2564
                              00000000GEF40      04C4   C5   D0 00071    MOVL    LINE_TYPE, RTYPE[R0]        ;
                                 00000000GEF40   7F 0007B    PUSHAQ  RLINES[R0]                          ; 2565
                                          7E      08   AC   7D 00082    MOVQ    PTR, -(SP)                  ;
                                             04   AC   DD 00086    PUSHL   INT_LEN                         ;
                                          6B         04   FB 00089    CALLS   #4, PADLIN                  ;
                                             67   DD 0008C    PUSHL   RCOUNT                              ; 2566
                                 00000000G   EF   9F 0008E    PUSHAB  RTYPE                               ;
                                 00000000G   EF   9F 00094    PUSHAB  RLINES                              ;
                              00000000V      EF   03 FB 0009A    CALLS   #3, SAVE_TELLTALE_HEADING           ;
                                          66     67   D1 000A1    CMPL    RCOUNT, ALLOWD                      ; 2568
```

```
                                                00AE  31 000A4          BRW     16$
                                                  64  D5 000A7 4$:      TSTL    LCOUNT                                          ; 2576
                                                  0F  12 000A9          BNEQ    7$
      04   AC            20        00   B5         01  2D 000AB          CMPC5   #1, @BLANKS, #32, INT_LEN, @PTR                 ; 2582
                                                08    BC   000B2
                                                  45  13 000B4 5$:      BEQL    11$
                        47          6A              06  E0 000B6 6$:      BBS     #6, CMDBLK, 12$                                 ; 2584
                        50          66              01  C3 000BA 7$:      SUBL3   #1, ALLOWD, R0                                  ; 2595
                                    50              64  D1 000BE          CMPL    LCOUNT, R0
                                                    05  12 000C1          BNEQ    9$
                                    56      04CC    C5  E9 000C3 8$:      BLBC    DOING_LAST, 14$
                                                    64  D6 000C8 9$:      INCL    LCOUNT                                          ; 2606
                                    50              64  D0 000CA          MOVL    LCOUNT, R0                                      ; 2607
                                  6840      04C4    C5  D0 000CD          MOVL    LINE_TYPE, LTYPE[R0]
                                                  6940  7F 000D3          PUSHAQ  LLINES[R0]                                      ; 2608
                                    7E        08    AC  7D 000D6          MOVQ    PTR, -(SP)
                                              04    AC  DD 000DA          PUSHL   INT_LEN
                                    6B              04  FB 000DD          CALLS   #4, PADLIN
                                                    64  DD 000E0          PUSHL   LCOUNT                                          ; 2609
                                                    58  DD 000E2          PUSHL   R8
                                                    59  DD 000E4          PUSHL   R9
                 00000000V          EF              03  FB 000E6          CALLS   #3, SAVE_TELLTALE_HEADING
                                                        04 000ED          RET                                                     ; 2527
                                                    64  D5 000EE 10$:     TSTL    LCOUNT                                          ; 2620
                                                    1E  12 000F0          BNEQ    13$
      04   AC            20        00   B5         01  2D 000F2          CMPC5   #1, @BLANKS, #32, INT_LEN, @PTR                 ; 2626
                                                08    BC   000F9
                                                  63  13 000FB 11$:     BEQL    17$
                        0F          6A              06  E1 000FD          BBC     #6, CMDBLK, 13$                                 ; 2628
                                    7E        08    AC  7D 00101 12$:     MOVQ    PTR, -(SP)                                      ; 2634
                                              04    AC  DD 00105          PUSHL   INT_LEN
                 00000000V          EF              03  FB 00108          CALLS   #3, DO_CONTINUE
                                                        04 0010F          RET
                        50          66              01  C3 00110 13$:     SUBL3   #1, ALLOWD, R0                                  ; 2639
                                    50              64  D1 00114          CMPL    LCOUNT, R0
                                                    14  12 00117          BNEQ    15$
                                    0F      04CC    C5  E8 00119          BLBS    DOING_LAST, 15$
                                    7E        08    AC  7D 0011E 14$:     MOVQ    PTR, -(SP)                                      ; 2644
                                              04    AC  DD 00122          PUSHL   INT_LEN
                 00000000V          EF              03  FB 00125          CALLS   #3, LAST_LINE
                                                        04 0012C          RET
                                                    64  D6 0012D 15$:     INCL    LCOUNT                                          ; 2650
                                    50              64  D0 0012F          MOVL    LCOUNT, R0                                      ; 2651
                                  6840      04C4    C5  D0 00132          MOVL    LINE_TYPE, LTYPE[R0]
                                                  6940  7F 00138          PUSHAQ  LLINES[R0]                                      ; 2652
                                    7E        08    AC  7D 0013B          MOVQ    PTR, -(SP)
                                              04    AC  DD 0013F          PUSHL   INT_LEN
                                    6B              04  FB 00142          CALLS   #4, PADLIN
                                                    64  DD 00145          PUSHL   LCOUNT                                          ; 2653
                                                    58  DD 00147          PUSHL   R8
                                                    59  DD 00149          PUSHL   R9
                 00000000V          EF              03  FB 0014B          CALLS   #3, SAVE_TELLTALE_HEADING
                                    66              64  D1 00152          CMPL    LCOUNT, ALLOWD                                  ; 2655
                                                    09  12 00155 16$:     BNEQ    17$
                                                    7E  D4 00157          CLRL    -(SP)
                 00000000G          EF              01  FB 00159          CALLS   #1, PUTPAG
                                                        04 00160 17$:     RET                                                     ; 2660
```

; Routine Size: 353 bytes, Routine Base: $CODE$ + 0AF0

```
;1893   2661  1 %SBTTL 'SAVE_TELLTALE_HEADING -- Save telltale heading string'
;1894   2662  1 ROUTINE SAVE_TELLTALE_HEADING (
;1895   2663  1     LINES : REF BLOCKVECTOR [, STR$K_D_BLN],
;1896   2664  1     TYPES : REF VECTOR,
;1897   2665  1     L_NUM
;1898   2666  1     ) : NOVALUE =
;1899   2667  1 !++
;1900   2668  1 !
;1901   2669  1 ! FUNCTIONAL DESCRIPTION:
;1902   2670  1 !
;1903   2671  1 !       This routine is called by INS_LINE to save the current line as a
;1904   2672  1 !       possible telltale heading.
;1905   2673  1 !
;1906   2674  1 !       The line is saved only if it is the first line of a continuation
;1907   2675  1 !       heading, the first line of a multi-line top level entry or a
;1908   2676  1 !       top-level entry which is only one line long.
;1909   2677  1 !
;1910   2678  1 !       If the line is saved, it is always saved as the right telltale string
;1911   2679  1 !       which represents the last top level entry on the page.
;1912   2680  1 !
;1913   2681  1 !       The line is also saved as the left telltale if the right telltale
;1914   2682  1 !       string is null which occurs after a right side page has been generated
;1915   2683  1 !       and if the line is the first on the page.
;1916   2684  1 !
;1917   2685  1 ! FORMAL PARAMETERS:
;1918   2686  1 !
;1919   2687  1 !       LINES           - Pointer to the blockvector of string descriptors
;1920   2688  1 !                         for the current column.
;1921   2689  1 !       TYPES           - Pointer to the vector of line types for the
;1922   2690  1 !                         current column.
;1923   2691  1 !       L_NUM           - Number of the line being generated.
;1924   2692  1 !
;1925   2693  1 ! IMPLICIT INPUTS:
;1926   2694  1 !
;1927   2695  1 !       RLINES [0, ...] - The right telltale string
;1928   2696  1 !
;1929   2697  1 ! IMPLICIT OUTPUTS:
;1930   2698  1 !
;1931   2699  1 !       RLINES [0, ...] - The right telltale string is replaced if the line
;1932   2700  1 !                         is saved.
;1933   2701  1 !       RTYPES [0]      - The right telltale line type is replaced if the
;1934   2702  1 !                         line is saved.
;1935   2703  1 !       LLINES [0, ...] - The left telltale string is replaced if a right hand
;1936   2704  1 !                         page was just generated.
;1937   2705  1 !       LTYPES [0, ...] - The left telltale line type is replaced if the
;1938   2706  1 !                         line is saved as a left telltale.
;1939   2707  1 !
;1940   2708  1 ! ROUTINE VALUE:
;1941   2709  1 ! COMPLETION CODES:
;1942   2710  1 !
;1943   2711  1 !       None
;1944   2712  1 !
;1945   2713  1 ! SIDE EFFECTS:
;1946   2714  1 !
;1947   2715  1 !       None
;1948   2716  1 !--
;1949   2717  1
```

```
; 1950          2718  2      BEGIN
; 1951          2719  2
; 1952          2720  2      LOCAL
; 1953          2721  2          LINE_NUMBER,
; 1954          2722  2          SAVE_LINE;
; 1955          2723  2
; 1956          2724  2      LINE_NUMBER = .L_NUM;
; 1957          2725  2      SAVE_LINE = FALSE;
; 1958          2726  2
; 1959          2727  2      SELECTONE .TYPES [.LINE_NUMBER] OF
; 1960          2728  2          SET
; 1961          2729  2
; 1962          2730  2          [CONT_HEAD]:
; 1963          2731  3              BEGIN
; 1964          2732  3              !
; 1965          2733  3              ! A continuation heading.
; 1966          2734  3              !
; 1967          2735  3              IF .LINE_NUMBER EQL 1
; 1968          2736  3              THEN
; 1969          2737  4                  BEGIN
; 1970          2738  4                  !
; 1971          2739  4                  ! Only save the first line in a continuation heading
; 1972          2740  4                  !
; 1973          2741  4                  SAVE_LINE = TRUE;
; 1974          2742  3                  END;
; 1975          2743  3                                                                                  ;
; 1976          2744  2              END;
; 1977          2745  2
; 1978          2746  2          [GUIDE]:
; 1979          2747  3              BEGIN
; 1980          2748  3              !
; 1981          2749  3              ! A guide heading.
; 1982          2750  3              !
; 1983          2751  3              BIND
; 1984          2752  3                  RHEAD = RLINES [0, 0,0,0,0] : $STR_DESCRIPTOR ();
; 1985          2753  3
; 1986          2754  4              IF  (.LINE_NUMBER EQL 1)
; 1987          2755  4              AND (.RHEAD [STR$H_LENGTH] EQL 0)
; 1988          2756  3              THEN
; 1989          2757  4                  BEGIN
; 1990          2758  4                  !
; 1991          2759  4                  ! This guide heading is the first line on a left hand page.
; 1992          2760  4                  ! Set flag which indicates that the first top-level entry
; 1993          2761  4                  ! on the page should be saved as the left telltale.
; 1994          2762  4                  !
; 1995          2763  4                  SAVE_ENTRY = TRUE;
; 1996          2764  3                  END;
; 1997          2765  3
; 1998          2766  2              END;
; 1999          2767  2
; 2000          2768  2          [ENTRY_E]:
; 2001          2769  3              BEGIN
; 2002          2770  3              !
; 2003          2771  3              ! The last line in a top level entry.
; 2004          2772  3              !
; 2005          2773  3              DECR I FROM .LINE_NUMBER - 1 TO 1 DO
; 2006          2774  4                  BEGIN
```

```
; 2007          2775  4                   !
; 2008          2776  4                   ! Find the first line in the entry.
; 2009          2777  4                   !
; 2010          2778  4                   SELECTONE .TYPES [.I] OF
; 2011          2779  4                       SET
; 2012          2780  4
; 2013          2781  4                       [ENTRY_B]:
; 2014          2782  5                           BEGIN
; 2015          2783  5                           !
; 2016          2784  5                           ! Found the first line in a multi-line entry.
; 2017          2785  5                           ! Save its line number and exit the loop.
; 2018          2786  5                           !
; 2019          2787  5                           LINE_NUMBER = .I;
; 2020          2788  5                           EXITLOOP;
; 2021          2789  4                           END;
; 2022          2790  4
; 2023          2791  4                       [ENTRY_W]:
; 2024          2792  4                           !
; 2025          2793  4                           ! Intermediate line of a multi-line entry.
; 2026          2794  4                           ! Continue search by doing nothing.
; 2027          2795  4                           !
; 2028          2796  4                           ;
; 2029          2797  4
; 2030          2798  4                       [OTHERWISE]:
; 2031          2799  4                           !
; 2032          2800  4                           ! Entry was not a multi-line entry. Exit the loop.
; 2033          2801  4                           !
; 2034          2802  4                           EXITLOOP;
; 2035          2803  4
; 2036          2804  4                       TES;
; 2037          2805  4
; 2038          2806  3                   END;
; 2039          2807  3
; 2040          2808  3               SAVE_LINE = TRUE;
; 2041          2809  2                   END;
; 2042          2810  2
; 2043          2811  2           [OTHERWISE]:
; 2044          2812  2                   !
; 2045          2813  2                   ! Line should not be saved as a candidate for a telltale heading.
; 2046          2814  2                   !
; 2047          2815  2                   ;
; 2048          2816  2
; 2049          2817  2           TES;
; 2050          2818  2
; 2051          2819  2       IF .SAVE_LINE
; 2052          2820  2       THEN
; 2053          2821  3           BEGIN
; 2054          2822  3           !
; 2055          2823  3           ! We found a line which could be used as a telltale head.
; 2056          2824  3           !
; 2057          2825  3           BIND
; 2058          2826  3               RHEAD = RLINES [0, 0,0,0,0] : $STR_DESCRIPTOR ();
; 2059          2827  3
; 2060          2828  4           IF  (.RHEAD [STR$H_LENGTH] EQL 0)
; 2061          2829  4           AND (
; 2062          2830  4               (.LINE_NUMBER EQL 1) OR .SAVE_ENTRY
; 2063          2831  4               )
```

```
;  2064          2832  3          THEN
;  2065          2833  4              BEGIN
;  2066          2834  4              !
;  2067          2835  4              ! The right telltale is null only after a right hand page has been
;  2068          2836  4              ! generated. In this case, the line should be saved as the left
;  2069          2837  4              ! telltale only if this is the first line on the page or if the
;  2070          2838  4              ! first line on the page was a guide heading.
;  2071          2839  4              !
;  2072          2840  4              SAVE_ENTRY = FALSE;
;  2073          2841  4
;  2074        P 2842  4              $STR_COPY (STRING = LINES [.LINE_NUMBER, 0,0,0,0],
;  2075          2843  4                  TARGET = LLINES [0, 0,0,0,0]);
;  2076          2844  4
;  2077          2845  4              LTYPE [0] = .TYPES [.LINE_NUMBER];
;  2078          2846  3              END;
;  2079          2847  3
;  2080          2848  3          !
;  2081          2849  3          ! Save the line as the right telltale.
;  2082          2850  3          !
;  2083        P 2851  3          $STR_COPY (STRING = LINES [.LINE_NUMBER, 0,0,0,0],
;  2084          2852  3              TARGET = RLINES [0, 0,0,0,0]);
;  2085          2853  3
;  2086          2854  3          RTYPE [0] = .TYPES [.LINE_NUMBER];
;  2087          2855  2          END;
;  2088          2856  2
;  2089          2857  1      END;


                                                                .EXTRN  XST$COPY, STR$FAILURE

                                                00FC 00000 SAVE_TELLTALE_HEADING:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7          ; 2662
57 00000000G   EF   9E 00002                    MOVAB   XST$COPY, R7                    ;
56 00000000G   EF   9E 00009                    MOVAB   STR$FAILURE, R6                 ;
55 00000000G   EF   9E 00010                    MOVAB   RHEAD, R5                       ;
54 00000000'   EF   9E 00017                    MOVAB   SAVE_ENTRY, R4                  ;
52         0C  AC   D0 0001E                    MOVL    L_NUM, LINE_NUMBER              ; 2724
               53   D4 00022                    CLRL    SAVE_LINE                       ; 2725
50         08 BC42  D0 00024                    MOVL    @TYPES[LINE_NUMBER], R0         ; 2727
0B              50   D1 00029                    CMPL    R0, #11                         ; 2730
                07   12 0002C                    BNEQ    1$                              ;
01              52   D1 0002E                    CMPL    LINE_NUMBER, #1                 ; 2735
                39   12 00031                    BNEQ    7$                              ;
                34   11 00033                    BRB     6$                              ; 2741
03              50   D1 00035 1$:                CMPL    R0, #3                          ; 2746
                0E   12 00038                    BNEQ    2$                              ;
01              52   D1 0003A                    CMPL    LINE_NUMBER, #1                 ; 2754
                2D   12 0003D                    BNEQ    7$                              ;
                65   B5 0003F                    TSTW    RHEAD                           ; 2755
                29   12 00041                    BNEQ    7$                              ;
64              01   D0 00043                    MOVL    #1, SAVE_ENTRY                  ; 2763
                24   11 00046                    BRB     7$                              ; 2727
07              50   D1 00048 2$:                CMPL    R0, #7                          ; 2768
                1F   12 0004B                    BNEQ    7$                              ;
50              52   D0 0004D                    MOVL    LINE_NUMBER, I                  ; 2773
                14   11 00050                    BRB     5$                              ;
```

```
                                 51        08 BC40  D0 00052 3$:      MOVL    @TYPES[I], R1                           ; 2778
                                 05              51  D1 00057        CMPL    R1, #5                                  ; 2781
                                                 05  12 0005A        BNEQ    4$
                                 52              50  D0 0005C        MOVL    I, LINE_NUMBER                          ; 2787
                                                 08  11 0005F        BRB     6$                                      ; 2782
                                 06              51  D1 00061 4$:     CMPL    R1, #6                                  ; 2791
                                                 03  12 00064        BNEQ    6$
                                 E9              50  F5 00066 5$:     SOBGTR  I, 3$                                   ; 2773
                                 53              01  D0 00069 6$:     MOVL    #1, SAVE_LINE                           ; 2808
                                 42              53  E9 0006C 7$:     BLBC    SAVE_LINE, 10$                          ; 2819
                                                 65  B5 0006F        TSTW    RHEAD                                   ; 2828
                                                 26  12 00071        BNEQ    9$
                                 01              52  D1 00073        CMPL    LINE_NUMBER, #1                         ; 2830
                                                 03  13 00076        BEQL    8$
                                 1E              64  E9 00078        BLBC    SAVE_ENTRY, 9$
                                                 64  D4 0007B 8$:     CLRL    SAVE_ENTRY                              ; 2840
                                                 56  DD 0007D        PUSHL   R6                                      ; 2843
                                                 7E  D4 0007F        CLRL    -(SP)
                                    00000000G    EF  9F 00081        PUSHAB  $STR$TARGET
                                           04 BC42  7F 00087        PUSHAQ  @LINES[LINE_NUMBER]
                                                 7E  D4 0008B        CLRL    -(SP)
                                 67              05  FB 0008D        CALLS   #5, XST$COPY
                    00000000G    EF        08 BC42  D0 00090        MOVL    @TYPES[LINE_NUMBER], LTYPE              ; 2845
                                                 56  DD 00099 9$:     PUSHL   R6                                      ; 2852
                                                 7E  D4 0009B        CLRL    -(SP)
                                                 55  DD 0009D        PUSHL   R5
                                           04 BC42  7F 0009F        PUSHAQ  @LINES[LINE_NUMBER]
                                                 7E  D4 000A3        CLRL    -(SP)
                                 67              05  FB 000A5        CALLS   #5, XST$COPY
                    00000000G    EF        08 BC42  D0 000A8        MOVL    @TYPES[LINE_NUMBER], RTYPE              ; 2854
                                                 04 000B1 10$:     RET                                             ; 2857

; Routine Size:  178 bytes,    Routine Base:  $CODE$ + 0C51
```

```
;  2091          2858  1 %SBTTL 'LAST_LINE -- Process last line in column'
;  2092          2859  1 ROUTINE LAST_LINE (I_LEN, PTR, E_LEN) : NOVALUE =
;  2093          2860  1 !++
;  2094          2861  1 !
;  2095          2862  1 ! FUNCTIONAL DESCRIPTION:
;  2096          2863  1 !
;  2097          2864  1 !       This routine processes the last line in the column.
;  2098          2865  1 !
;  2099          2866  1 !       If the current line is the beginning or middle of a subindex entry,
;  2100          2867  1 !       any preceeding lines which are also part of that subindex entry
;  2101          2868  1 !       will be moved to the next column. If the line preceeding the subindex
;  2102          2869  1 !       entry is the end of a top level entry, the top level entry will also
;  2103          2870  1 !       be moved. If the line preceeding the top level entry is part of a
;  2104          2871  1 !       guide head, the guide head will also be moved.
;  2105          2872  1 !
;  2106          2873  1 !       If the line is part of a top level entry, the entry will be moved if
;  2107          2874  1 !       either it is not the last line in the entry or if the entry has
;  2108          2875  1 !       subindex entries. If the entry is to be moved and the line preceeding
;  2109          2876  1 !       the heading is part of a guide heading, the guide heading is also moved.
;  2110          2877  1 !
;  2111          2878  1 !       If the line is part of a guide heading, the guide heading is moved
;  2112          2879  1 !       to the next column.
;  2113          2880  1 !
;  2114          2881  1 !       Lines are moved from the current column to the next by determining
;  2115          2882  1 !       the number of lines to be moved and by establishing a pointer to
;  2116          2883  1 !       the first line to be moved. The lines are then copied to the temporary
;  2117          2884  1 !       column TLINES and the rest of the current column are filled with
;  2118          2885  1 !       blank FILL lines. The copied lines are then inserted into the next
;  2119          2886  1 !       output column.
;  2120          2887  1 !
;  2121          2888  1 ! FORMAL PARAMETERS:
;  2122          2889  1 !
;  2123          2890  1 !       I_LEN   - Internal line length
;  2124          2891  1 !       PTR     - CH$PTR to line
;  2125          2892  1 !       E_LEN   - External length
;  2126          2893  1 !
;  2127          2894  1 ! IMPLICIT INPUTS:
;  2128          2895  1 !
;  2129          2896  1 !       ALLOWD          - number of lines allowed in column
;  2130          2897  1 !       RTYPE           - line types if doing right column
;  2131          2898  1 !       RLINES          - right column lines if doing right column
;  2132          2899  1 !       RCOUNT          - number of lines in column if doing right column
;  2133          2900  1 !       LTYPE           - line types if doing left column
;  2134          2901  1 !       LLINES          - left column lines if doing left column
;  2135          2902  1 !       LCOUNT          - number of lines in left column
;  2136          2903  1 !       LINE_TYPE       - type of current line
;  2137          2904  1 !       CMDBLK          - command line information block
;  2138          2905  1 !
;  2139          2906  1 ! IMPLICIT OUTPUTS:
;  2140          2907  1 !
;  2141          2908  1 !       None
;  2142          2909  1 !
;  2143          2910  1 ! ROUTINE VALUE:
;  2144          2911  1 ! COMPLETION CODES:
;  2145          2912  1 !
;  2146          2913  1 !       None
;  2147          2914  1 !
```

```
;  2148          2915  1 ! SIDE EFFECTS:
;  2149          2916  1 !
;  2150          2917  1 !       During the execution of this routine, DOING_LAST is set to TRUE
;  2151          2918  1 !       to prevent INS_LINE from calling this routine again which would
;  2152          2919  1 !       cause an infinite loop of calls.
;  2153          2920  1 !
;  2154          2921  1 !       TLINES AND TTYPE are used during the execution of this routine.
;  2155          2922  1 !       Any other routine which may be called during the execution of
;  2156          2923  1 !       this routine must use some other area to store lines and line types.
;  2157          2924  1 !--
;  2158          2925  2     BEGIN
;  2159          2926  2
;  2160          2927  2     LOCAL
;  2161          2928  2         TYPE,
;  2162          2929  2         COUNT_PTR,
;  2163          2930  2         TYPE_PTR        : REF VECTOR,
;  2164          2931  2         LINES_PTR       : REF BLOCKVECTOR [, STR$K_D_BLN],
;  2165          2932  2         COPY_LINES,
;  2166          2933  2         FILL_LINES,
;  2167          2934  2         S               : REF $STR_DESCRIPTOR ();
;  2168          2935  2
;  2169          2936  2     DOING_LAST = TRUE;                          ! Tell INS_LINE we're processing last line
;  2170          2937  2
;  2171          2938  2     COPY_LINES = 0;                             ! Number of lines to copy to next column
;  2172          2939  2     FILL_LINES = 0;                             ! Number of fill lines to insert in this column
;  2173          2940  2     TYPE = .LINE_TYPE;                          ! Save current line type
;  2174          2941  2
;  2175          2942  2     IF .LCOUNT EQL .ALLOWD
;  2176          2943  2     THEN
;  2177          2944  3         BEGIN                                   ! Filling right column
;  2178          2945  3         COUNT_PTR = RCOUNT;                     ! Set up pointers to column variables
;  2179          2946  3         TYPE_PTR = RTYPE [0];
;  2180          2947  3         LINES_PTR = RLINES [0, 0,0,0,0];
;  2181          2948  3         END
;  2182          2949  2     ELSE
;  2183          2950  3         BEGIN                                   ! Filling left column
;  2184          2951  3         COUNT_PTR = LCOUNT;                     ! Set up pointers...
;  2185          2952  3         TYPE_PTR = LTYPE [0];
;  2186          2953  3         LINES_PTR = LLINES [0, 0,0,0,0];
;  2187          2954  2         END;
;  2188          2955  2
;  2189          2956  2     CASE .TYPE FROM BKT_E TO CONT_HEAD OF
;  2190          2957  2     SET
;  2191          2958  2
;  2192          2959  2     [SUB_B, SUB_W]:
;  2193          2960  3         BEGIN
;  2194          2961  3         !
;  2195          2962  3         ! Beginning or middle of subindex entry
;  2196          2963  3         !
;  2197          2964  3         DECR I FROM ..COUNT_PTR TO 1 DO
;  2198          2965  4             IF (.TYPE_PTR [.I] NEQ SUB_W) AND (.TYPE_PTR [.I] NEQ SUB_B)
;  2199          2966  3             THEN
;  2200          2967  3                 EXITLOOP
;  2201          2968  3             ELSE
;  2202          2969  4                 BEGIN
;  2203          2970  4                 COPY_LINES = .COPY_LINES + 1;   ! Copy this line to next column
;  2204          2971  4                 .COUNT_PTR = ..COUNT_PTR - 1;   ! One less line in column
```

```
;  2205     2972  3                    END;
;  2206     2973  3
;  2207     2974  4          IF (.TYPE_PTR [..COUNT_PTR] EQL ENTRY_E)
;  2208     2975  3          AND .CMDBLK [NDX$V_CONTINUATION]
;  2209     2976  3          THEN
;  2210     2977  4              BEGIN
;  2211     2978  4              !
;  2212     2979  4              ! Previous line is the end of an entry and we are
;  2213     2980  4              ! doing continuation headings. Move the entry too.
;  2214     2981  4              !
;  2215     2982  4              COPY_LINES = .COPY_LINES + 1;       ! Copy this line to next column
;  2216     2983  4              .COUNT_PTR = ..COUNT_PTR - 1;       ! One less line in column
;  2217     2984  4
;  2218     2985  4              DECR I FROM ..COUNT_PTR TO 1 DO
;  2219     2986  5                  IF (.TYPE_PTR [.I] NEQ ENTRY_W) AND (.TYPE_PTR [.I] NEQ ENTRY_B)
;  2220     2987  4                  THEN
;  2221     2988  4                      EXITLOOP
;  2222     2989  4                  ELSE
;  2223     2990  5                      BEGIN
;  2224     2991  5                      COPY_LINES = .COPY_LINES + 1;
;  2225     2992  5                      .COUNT_PTR = ..COUNT_PTR - 1;
;  2226     2993  4                      END;
;  2227     2994  4
;  2228     2995  4              IF .TYPE_PTR [..COUNT_PTR] EQL GUIDE_FILL
;  2229     2996  4              THEN
;  2230     2997  5                  BEGIN                           ! Move the guide heading too
;  2231     2998  5                  COPY_LINES = .COPY_LINES + 2;
;  2232     2999  5                  .COUNT_PTR = ..COUNT_PTR - 2;
;  2233     3000  4                  END;
;  2234     3001  3              END;
;  2235     3002  3
;  2236     3003  3          FILL_LINES = .COPY_LINES + 1;
;  2237     3004  2          END;
;  2238     3005  2
;  2239     3006  2      [ENTRY_B TO ENTRY_E]:
;  2240     3007  3          BEGIN
;  2241     3008  3          !
;  2242     3009  3          ! An index entry
;  2243     3010  3          !
;  2244     3011  4          IF (.TYPE NEQ ENTRY_E)
;  2245     3012  4          OR (.CMDBLK [NDX$V_CONTINUATION] AND (.LSTPTR [XE$A_SUBX] NEQ 0))
;  2246     3013  3          THEN
;  2247     3014  4              BEGIN
;  2248     3015  4              !
;  2249     3016  4              ! Line is not the end of an entry
;  2250     3017  4                      OR
;  2251     3018  4              ! Doing continuation headings and have subindex entries.
;  2252     3019  4              !
;  2253     3020  4              ! Move it to next column.
;  2254     3021  4              !
;  2255     3022  4              DECR I FROM ..COUNT_PTR TO 1 DO
;  2256     3023  5                  IF (.TYPE_PTR [.I] NEQ ENTRY_W) AND (.TYPE_PTR [.I] NEQ ENTRY_B)
;  2257     3024  4                  THEN
;  2258     3025  4                      EXITLOOP
;  2259     3026  4                  ELSE
;  2260     3027  5                      BEGIN
;  2261     3028  5                      COPY_LINES = .COPY_LINES + 1;
```

```
; 2262          3029  5                       .COUNT_PTR = ..COUNT_PTR - 1;
; 2263          3030  4                       END;
; 2264          3031  4
; 2265          3032  4                 IF .TYPE_PTR [..COUNT_PTR] EQL GUIDE_FILL
; 2266          3033  4                 THEN
; 2267          3034  5                     BEGIN                               ! Move the guide heading too
; 2268          3035  5                     COPY_LINES = .COPY_LINES + 2;
; 2269          3036  5                     .COUNT_PTR = ..COUNT_PTR - 2;
; 2270          3037  4                     END;
; 2271          3038  4
; 2272          3039  4                 FILL_LINES = .COPY_LINES + 1;
; 2273          3040  3                 END;
; 2274          3041  2             END;
; 2275          3042  2
; 2276          3043  2         [GUIDE_FILL]:
; 2277          3044  3             BEGIN
; 2278          3045  3             !
; 2279          3046  3             ! Line following guide heading.
; 2280          3047  3             !
; 2281          3048  3             COPY_LINES = 1;
; 2282          3049  3             .COUNT_PTR = ..COUNT_PTR - 1;
; 2283          3050  3             FILL_LINES = 2;
; 2284          3051  2             END;
; 2285          3052  2
; 2286          3053  2         [GUIDE]:
; 2287          3054  2             !
; 2288          3055  2             ! Guide heading.
; 2289          3056  2             ! Start guide heading in next column.
; 2290          3057  2             !
; 2291          3058  2             FILL_LINES = 1;
; 2292          3059  2
; 2293          3060  2         [INRANGE]:
; 2294          3061  2             !
; 2295          3062  2             ! Line is one of: BKT_E, FILL, SUB_E or CONT_HEAD. Just insert it.
; 2296          3063  2             !
; 2297          3064  2             ;
; 2298          3065  2
; 2299          3066  2         TES;
; 2300          3067  2
; 2301          3068  2         IF .COPY_LINES NEQ 0
; 2302          3069  2         THEN
; 2303          3070  2             !
; 2304          3071  2             ! Copy lines to temp column
; 2305          3072  2             !
; 2306          3073  2             INCR I FROM 1 TO .COPY_LINES DO
; 2307          3074  3                 BEGIN
; 2308        P 3075  3                 $STR_COPY (STRING = LINES_PTR [..COUNT_PTR + .I, 0,0,0,0],
; 2309          3076  3                     TARGET = TLINES [.I, 0,0,0,0]);
; 2310          3077  3                 TTYPE [.I] = .TYPE_PTR [..COUNT_PTR + .I];
; 2311          3078  2                 END;
; 2312          3079  2
; 2313          3080  2             !
; 2314          3081  2             ! Copy line we started with
; 2315          3082  2             !
; 2316          3083  2             COPY_LINES = .COPY_LINES + 1;
; 2317          3084  2             TTYPE [.COPY_LINES] = .TYPE;
; 2318          3085  2             PADLIN (.I_LEN, .PTR, .E_LEN, TLINES [.COPY_LINES, 0,0,0,0]);
```

```
;  2319          3086  2
;  2320          3087  2      IF .FILL_LINES NEQ 0
;  2321          3088  2      THEN
;  2322          3089  3          BEGIN
;  2323          3090  3          !
;  2324          3091  3          ! Insert FILL lines to fill out rest of column
;  2325          3092  3          !
;  2326          3093  3          LINE_TYPE = FILL;
;  2327          3094  3
;  2328          3095  3          INCR I FROM 1 TO .FILL_LINES DO INS_LINE (4, .BLANKS, 4);
;  2329          3096  2          END;
;  2330          3097  2
;  2331          3098  2      IF .COPY_LINES NEQ 0
;  2332          3099  2      THEN
;  2333          3100  2          !
;  2334          3101  2          ! Put copied lines in new column
;  2335          3102  2          !
;  2336          3103  2          INCR I FROM 1 TO .COPY_LINES DO
;  2337          3104  3              BEGIN
;  2338          3105  3              S = TLINES [.I, 0,0,0,0];
;  2339          3106  3              LINE_TYPE = .TTYPE [.I];
;  2340          3107  3              INS_LINE (.S [STR$H_LENGTH], .S [STR$A_POINTER], 0);
;  2341          3108  2              END;
;  2342          3109  2
;  2343          3110  2      DOING_LAST = FALSE;                         ! Not doing last line any more
;  2344          3111  2
;  2345          3112  1      END;


                                        0FFC 00000 LAST_LINE:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 2859
                                    5B 00000000G   EF   9E 00002          MOVAB   TLINES, R11                   ;
                                    5A 00000000'   EF   9E 00009          MOVAB   LINE_TYPE, R10                ;
                                08  AA             01   D0 00010          MOVL    #1, DOING_LAST                ; 2936
                                                  55   D4 00014          CLRL    COPY_LINES                    ; 2938
                                                  57   D4 00016          CLRL    FILL_LINES                    ; 2939
                                    59            6A   D0 00018          MOVL    LINE_TYPE, TYPE               ; 2940
                        00000000G   EF 00000000G  EF   D1 0001B          CMPL    LCOUNT, ALLOWD                ; 2942
                                                  17   12 00026          BNEQ    1$                            ;
                                    52 00000000G  EF   9E 00028          MOVAB   RCOUNT, COUNT_PTR             ; 2945
                                    56 00000000G  EF   9E 0002F          MOVAB   RTYPE, TYPE_PTR               ; 2946
                                    58 00000000G  EF   9E 00036          MOVAB   RLINES, LINES_PTR             ; 2947
                                                  15   11 0003D          BRB     2$                            ; 2942
                                    52 00000000G  EF   9E 0003F 1$:      MOVAB   LCOUNT, COUNT_PTR             ; 2951
                                    56 00000000G  EF   9E 00046          MOVAB   LTYPE, TYPE_PTR               ; 2952
                                    58 00000000G  EF   9E 0004D          MOVAB   LLINES, LINES_PTR             ; 2953
                                0A              01   59   CF 00054 2$:      CASEL   TYPE, #1, #10                 ; 2956
                00AF            00B9            00BC           00BC     00058 3$:      .WORD   23$-3$,-              ;
                0016            0068            0068           0068     00060          23$-3$,-              ;
                                00BC            00BC           0016     00068          22$-3$,-              ;
                                                                                      21$-3$,-              ;
                                                                                      13$-3$,-              ;
                                                                                      13$-3$,-              ;
                                                                                      13$-3$,-              ;
```

```
                                                                4$-3$,-
                                                                4$-3$,-
                                                                23$-3$,-
                                                                23$-3$
                50              62          01  C1 0006E 4$:    ADDL3   #1, (COUNT_PTR), I                 ; 2965
                                            10  11 00072        BRB     7$
                                09        6640  D1 00074 5$:    CMPL    (TYPE_PTR)[I], #9
                                            06  13 00078        BEQL    6$
                                08        6640  D1 0007A        CMPL    (TYPE_PTR)[I], #8
                                            07  12 0007E        BNEQ    8$
                                            55  D6 00080 6$:    INCL    COPY_LINES                        ; 2970
                                            62  D7 00082        DECL    (COUNT_PTR)                       ; 2971
                                ED          50  F5 00084 7$:    SOBGTR  I, 5$                             ; 2965
                                50          62  D0 00087 8$:    MOVL    (COUNT_PTR), R0                   ; 2974
                                07        6640  D1 0008A        CMPL    (TYPE_PTR)[R0], #7
                                            71  12 0008E        BNEQ    20$
    69 00000000G                EF          06  E1 00090        BBC     #6, CMDBLK, 20$                   ; 2975
                                            55  D6 00098        INCL    COPY_LINES                        ; 2982
                                            62  D7 0009A        DECL    (COUNT_PTR)                       ; 2983
                50              62          01  C1 0009C        ADDL3   #1, (COUNT_PTR), I                ; 2985
                                            10  11 000A0        BRB     11$
                                06        6640  D1 000A2 9$:    CMPL    (TYPE_PTR)[I], #6                 ; 2986
                                            06  13 000A6        BEQL    10$
                                05        6640  D1 000A8        CMPL    (TYPE_PTR)[I], #5
                                            07  12 000AC        BNEQ    12$
                                            55  D6 000AE 10$:    INCL    COPY_LINES                        ; 2991
                                            62  D7 000B0        DECL    (COUNT_PTR)                       ; 2992
                                ED          50  F5 000B2 11$:    SOBGTR  I, 9$                             ; 2986
                                50          62  D0 000B5 12$:    MOVL    (COUNT_PTR), R0                   ; 2995
                                04        6640  D1 000B8        CMPL    (TYPE_PTR)[R0], #4
                                            3D  13 000BC        BEQL    19$
                                            41  11 000BE        BRB     20$                               ; 3003
                                07            59  D1 000C0 13$:    CMPL    TYPE, #7                      ; 3011
                                              14  12 000C3        BNEQ    14$
    47 00000000G                EF            06  E1 000C5        BBC     #6, CMDBLK, 23$               ; 3012
                                50 00000000G  EF  D0 000CD        MOVL    LSTPTR, R0
                                         08   A0  D5 000D4        TSTL    8(R0)
                                              3B  13 000D7        BEQL    23$
                50              62            C1  C1 000D9 14$:    ADDL3   #1, (COUNT_PTR), I            ; 3023
                                              10  11 000DD        BRB     17$
                                06          6640  D1 000DF 15$:    CMPL    (TYPE_PTR)[I], #6
                                              06  13 000E3        BEQL    16$
                                05          6640  D1 000E5        CMPL    (TYPE_PTR)[I], #5
                                              07  12 000E9        BNEQ    18$
                                              55  D6 000EB 16$:    INCL    COPY_LINES                    ; 3028
                                              62  D7 000ED        DECL    (COUNT_PTR)                   ; 3029
                                ED            50  F5 000EF 17$:    SOBGTR  I, 15$                        ; 3023
                                50            62  D0 000F2 18$:    MOVL    (COUNT_PTR), R0               ; 3032
                                04          6640  D1 000F5        CMPL    (TYPE_PTR)[R0], #4
                                              06  12 000F9        BNEQ    20$
                                55            02  C0 000FB 19$:    ADDL2   #2, COPY_LINES                ; 3035
                                62            02  C2 000FE        SUBL2   #2, (COUNT_PTR)               ; 3036
                                57          01    A5 9E 00101 20$:    MOVAB   1(R5), FILE_LINES         ; 3039
                                              0D  11 00105        BRB     23$                           ; 2956
                                55            01  D0 00107 21$:    MOVL    #1, COPY_LINES                ; 3048
                                              62  D7 0010A        DECL    (COUNT_PTR)                   ; 3049
                                57            02  D0 0010C        MOVL    #2, FILE_LINES                ; 3050
```

```
                                    03  11 0010F             BRB     23$                                     ; 2956
                              57    01  D0 00111 22$:       MOVL    #1, FILL_LINES                          ; 3058
                                    55  D5 00114 23$:       TSTL    COPY_LINES                              ; 3068
                                    2C  13 00116             BEQL    26$                                     ;
                                    53  D4 00118             CLRL    I                                       ; 3077
                                    24  11 0011A             BRB     25$                                     ;
                        54    62    53  C1 0011C 24$:       ADDL3   I, (COUNT_PTR), R4                      ; 3076
                                00000000G   EF  9F 00120     PUSHAB  STR$FAILURE                             ;
                                            7E  D4 00126     CLRL    -(SP)                                   ;
                                        6B43    7F 00128     PUSHAQ  TLINES[I]                               ;
                                        6844    7F 0012B     PUSHAQ  (LINES_PTR)[R4]                         ;
                                            7E  D4 0012E     CLRL    -(SP)                                   ;
                          00000000G     EF      05  FB 00130 CALLS   #5, XST$COPY                            ;
                          00000000GEF43         6644  D0 00137 MOVL    (TYPE_PTR)[R4], TTYPE[I]                ; 3077
                        D8    53              55  F3 00140 25$:       AOBLEQ  COPY_LINES, I, 24$              ; 3073
                                              55  D6 00144 26$:       INCL    COPY_LINES                      ; 3083
                          00000000GEF45         59  D0 00146 MOVL    TYPE, TTYPE[COPY_LINES]                 ; 3084
                                        6B45    7F 0014E     PUSHAQ  TLINES[COPY_LINES]                      ; 3085
                                    7E      08  AC  7D 00151 MOVQ    PTR, -(SP)                              ;
                                        04  AC  DD 00155     PUSHL   I_LEN                                   ;
                          00000000V     EF      04  FB 00158 CALLS   #4, PADLIN                              ;
                                              57  D5 0015F     TSTL    FILL_LINES                              ; 3087
                                              18  13 00161     BEQL    29$                                     ;
                                    6A        02  D0 00163     MOVL    #2, LINE_TYPE                           ; 3093
                                              53  D4 00166     CLRL    I                                       ; 3095
                                              0D  11 00168     BRB     28$                                     ;
                                              04  DD 0016A 27$:       PUSHL   #4                              ;
                                        FB3C      CA  DD 0016C PUSHL   BLANKS                                  ;
                                              04  DD 00170     PUSHL   #4                                      ;
                                FC76      CF      03  FB 00172 CALLS   #3, INS_LINE                            ;
                            EF        53          57  F3 00177 28$:       AOBLEQ  FILL_LINES, I, 27$          ;
                                              55  D5 0017B 29$:       TSTL    COPY_LINES                      ; 3098
                                              21  13 0017D     BEQL    32$                                     ;
                                              53  D4 0017F     CLRL    I                                       ; 3103
                                              19  11 00181     BRB     31$                                     ;
                                    52        6B43  7E 00183 30$:       MOVAQ   TLINES[I], S                    ; 3105
                                    6A  00000000GEF43   D0 00187 MOVL    TTYPE[I], LINE_TYPE                 ; 3106
                                                7E  D4 0018F     CLRL    -(SP)                               ; 3107
                                            04  A2  DD 00191     PUSHL   4(S)                                ;
                                    7E          62  3C 00194     MOVZWL  (S), -(SP)                          ;
                                FC51      CF      03  FB 00197 CALLS   #3, INS_LINE                            ;
                            E3        53          55  F3 0019C 31$:       AOBLEQ  COPY_LINES, I, 30$          ; 3103
                                            08  AA  D4 001A0 32$:       CLRL    DOING_LAST                  ; 3110
                                                04 001A3             RET                                     ; 3112

; Routine Size:  420 bytes,    Routine Base:  $CODE$ + 0D03
```

```
;  2347     3113  1 %SBTTL 'DO_CONTINUE -- Generate continuation heading if necessary'
;  2348     3114  1 ROUTINE DO_CONTINUE (I_LEN, PTR, E_LEN) : NOVALUE =
;  2349     3115  1 !++
;  2350     3116  1 !
;  2351     3117  1 ! FUNCTIONAL DESCRIPTION:
;  2352     3118  1 !
;  2353     3119  1 !       This routine is called to generate a continuation heading if necessary
;  2354     3120  1 !
;  2355     3121  1 !       The input line is copied to local storage since it may reside in
;  2356     3122  1 !       RNO_LINE which is used to build the continuation heading. Note
;  2357     3123  1 !       that the input line cannot be copied to TLINES since this routine
;  2358     3124  1 !       may be called during the execution of LAST_LINE which uses TLINES
;  2359     3125  1 !       to temporarily store one to many lines of text.
;  2360     3126  1 !
;  2361     3127  1 !       The continuation heading is generated by setting LINE_TYPE to
;  2362     3128  1 !       CONT_HEAD and by scanning the subindex list stack from bottom
;  2363     3129  1 !       to top, calling FORMAT_ENTRY for each level.
;  2364     3130  1 !
;  2365     3131  1 ! FORMAL PARAMETERS:
;  2366     3132  1 !
;  2367     3133  1 !       I_LEN   - Internal length of line being processed
;  2368     3134  1 !       PTR     - CH$PTR to line
;  2369     3135  1 !       E_LEN   - External length of line being processed
;  2370     3136  1 !
;  2371     3137  1 ! IMPLICIT INPUTS:
;  2372     3138  1 !
;  2373     3139  1 !       CMDBLK  - Command line information block
;  2374     3140  1 !       LSTSTK  - Subindex list stack
;  2375     3141  1 !       INDLVL  - Subindex level
;  2376     3142  1 !       LINE_TYPE - Type of current line
;  2377     3143  1 !
;  2378     3144  1 ! IMPLICIT OUTPUTS:
;  2379     3145  1 !
;  2380     3146  1 !       None
;  2381     3147  1 !
;  2382     3148  1 ! ROUTINE VALUE:
;  2383     3149  1 ! COMPLETION CODES:
;  2384     3150  1 !
;  2385     3151  1 !       None
;  2386     3152  1 !
;  2387     3153  1 ! SIDE EFFECTS:
;  2388     3154  1 !
;  2389     3155  1 !       Sets CMDBLK [NDX$V_CONTINUATION] to false while processing
;  2390     3156  1 !       the input line. This prevents INS_LINE from calling the routine
;  2391     3157  1 !       again which would cause an infinite loop.
;  2392     3158  1 !
;  2393     3159  1 !       Sets LINE_TYPE to CONT_HEAD so that FORMAT_ENTRY will generate
;  2394     3160  1 !       a continuation heading instead of normal index entries.
;  2395     3161  1 !--
;  2396     3162  2     BEGIN
;  2397     3163  2     LOCAL
;  2398     3164  2         LINE : VECTOR [CH$ALLOCATION(1200)],
;  2399     3165  2         INT,
;  2400     3166  2         EXT,
;  2401     3167  2         TYP;
;  2402     3168  2
;  2403     3169  2     CMDBLK [NDX$V_CONTINUATION] = FALSE;        ! Prevent infinite loop of calls
```

```
;  2404          3170  2
;  2405          3171  2      !
;  2406          3172  2      ! Copy input parameters and input line
;  2407          3173  2      !
;  2408          3174  2      INT = .I_LEN;
;  2409          3175  2      EXT = .E_LEN;
;  2410          3176  2      TYP = .LINE_TYPE;
;  2411          3177  2      CH$MOVE (.INT, .PTR, CH$PTR (LINE));
;  2412          3178  2      CLR_RNO_LINE ();
;  2413          3179  2
;  2414          3180  3      IF (.TYP EQL SUB_B)                         ! First line of subindex entry
;  2415          3181  3      OR (.TYP EQL SUB_E)                         ! Last line of subindex entry
;  2416          3182  2      THEN                                        ! NOTE: can't get SUB_W as first
;  2417          3183  3          BEGIN                                   ! line in column.
;  2418          3184  3          LOCAL
;  2419          3185  3              PTR,
;  2420          3186  3              SAV_INDLVL;
;  2421          3187  3
;  2422          3188  3          PTR = .LSTPTR;                          ! Save current entry pointer
;  2423          3189  3          SAV_INDLVL = .INDLVL;                   ! Save current index level
;  2424          3190  3          LSTSTK [.INDLVL] = 0;                   ! Make sure top of stack is zero
;  2425          3191  3
;  2426          3192  3          INCR I FROM 0 TO .SAV_INDLVL - 1 DO
;  2427          3193  4              BEGIN                               ! Generate continuation heading
;  2428          3194  4              INDLVL = .I;
;  2429          3195  4              LSTPTR = .LSTSTK [.I];
;  2430          3196  4              LINE_TYPE = CONT_HEAD;
;  2431          3197  4              FORMAT_ENTRY ();
;  2432          3198  3              END;
;  2433          3199  3
;  2434          3200  3          INDLVL = .SAV_INDLVL;                   ! Restore current index level
;  2435          3201  3          LSTPTR = .PTR;                          ! and entry pointer
;  2436          3202  2          END;
;  2437          3203  2
;  2438          3204  2      LINE_TYPE = .TYP;
;  2439          3205  2      INS_LINE (.INT, CH$PTR (LINE), .EXT);       ! Insert original line
;  2440          3206  2      CMDBLK [NDX$V_CONTINUATION] = TRUE;         ! Reset continuation switch
;  2441          3207  1      END;


                                        0FFC 00000 DO_CONTINUE:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 3114
                                    5B 00000000G   EF   9E 00002   MOVAB   INDLVL, R11                         ;
                                    5A 00000000G   EF   9E 00009   MOVAB   LSTPTR, R10                         ;
                                    59 00000000'   EF   9E 00010   MOVAB   LINE_TYPE, R9                       ;
                                    5E      FB50   CE   9E 00017   MOVAB   -1200(SP), SP                      ;
                          00000000G EF        40   8F   8A 0001C   BICB2   #64, CMDBLK                        ; 3169
                                    57        04   AC   D0 00024   MOVL    I_LEN, INT                        ; 3174
                                    58        0C   AC   D0 00028   MOVL    E_LEN, EXT                        ; 3175
                                    56             69   D0 0002C   MOVL    LINE_TYPE, TYP                    ; 3176
                                6E      08   BC        57   28 0002F   MOVC3   INT, @PTR, LINE                ; 3177
                                      FB40   C9   FB44   C9   9E 00034   MOVAB   RNO_LINE, LINE_POINTER        ; 3178
                                             F4   A9   7C 0003B   CLRQ    INT_LINE_LENGTH                   ;
                                             FC   A9   D4 0003E   CLRL    TMS_LINE_LENGTH                   ;
```

```
                                 08              56  D1 00041          CMPL    TYP, #8                                 ; 3180
                                                 05  13 00044          BEQL    1$                                      ;
                                 0A              56  D1 00046          CMPL    TYP, #10                                ; 3181
                                                 32  12 00049          BNEQ    4$                                      ;
                                 54              6A  D0 0004B 1$:      MOVL    LSTPTR, PTR                             ; 3188
                                 50              6B  D0 0004E          MOVL    INDLVL, R0                              ; 3189
                                 53              50  D0 00051          MOVL    R0, SAV_INDLVL                          ;
                                    00000000GEF  40  D4 00054          CLRL    LSTSTK[R0]                              ; 3190
                                 52              01  CE 00058          MNEGL   #1, I                                   ; 3192
                                                 13  11 0005E          BRB     3$                                      ;
                                 6B              52  D0 00060 2$:      MOVL    I, INDLVL                               ; 3194
                                 6A 00000000GEF  42  D0 00063          MOVL    LSTSTK[I], LSTPTR                       ; 3195
                                 69              0B  D0 0006B          MOVL    #11, LINE_TYPE                          ; 3196
                              F5DD   CF          00  FB 0006E          CALLS   #0, FORMAT_ENTRY                        ; 3197
                           E9        52          53  F2 00073 3$:      AOBLSS  SAV_INDLVL, I, 2$                       ; 3192
                                 6B              53  D0 00077          MOVL    SAV_INDLVL, INDLVL                      ; 3200
                                 6A              54  D0 0007A          MOVL    PTR, LSTPTR                             ; 3201
                                 69              56  D0 0007D 4$:      MOVL    TYP, LINE_TYPE                          ; 3204
                                                 58  DD 00080          PUSHL   EXT                                     ; 3205
                                          04     AE  9F 00082          PUSHAB  LINE                                    ;
                                                 57  DD 00085          PUSHL   INT                                     ;
                              FBBD   CF          03  FB 00087          CALLS   #3, INS_LINE                            ;
                        00000000G    EF     40  8F  88 0008C          BISB2   #64, CMDBLK                             ; 3206
                                                 04  00094          RET                                             ; 3207

; Routine Size:  149 bytes,    Routine Base:  $CODE$ + 0EA7
```

```
;2443   3208  1 %SBTTL 'FILL_LINE -- Format an output line'
;2444   3209  1 ROUTINE FILL_LINE (I_LEN, I_PTR, INDENT, MAX_EXT, BOLD, UND, L_BLANK) : NOVALUE =
;2445   3210  1 !++
;2446   3211  1 !
;2447   3212  1 ! FUNCTIONAL DESCRIPTION:
;2448   3213  1 !
;2449   3214  1 !       This routine copies and formats the string specified by
;2450   3215  1 !       I_LEN and I_PTR into RNO_LINE. If the external length
;2451   3216  1 !       of the line exceeds MAX_EXT, the line is forced out.
;2452   3217  1 !
;2453   3218  1 ! FORMAL PARAMETERS:
;2454   3219  1 !
;2455   3220  1 !       I_LEN   - Length of input string
;2456   3221  1 !       I_PTR   - Pointer to input string
;2457   3222  1 !       INDENT  - Number of spaces to indent line
;2458   3223  1 !       MAX_EXT - Maximum external line length allowed.
;2459   3224  1 !       BOLD    - TRUE if string is to be bolded
;2460   3225  1 !       UND     - TRUE if string is to be underlined
;2461   3226  1 !       L_BLANK - TRUE if a blank should be inserted before the string
;2462   3227  1 !
;2463   3228  1 ! IMPLICIT INPUTS:
;2464   3229  1 !
;2465   3230  1 !       LINE_POINTER    - pointer to next character in output line
;2466   3231  1 !       EXT_LINE_LENGTH - external line length
;2467   3232  1 !       INT_LINE_LENGTH - internal line length
;2468   3233  1 !       CMDBLK          - command line information block
;2469   3234  1 !       TMSSTD          - standard character size for TMS
;2470   3235  1 !       TMS_LINE_LENGTH - length of line in TMS relative units
;2471   3236  1 !       CHRSIZ          - TMS character size vector
;2472   3237  1 !
;2473   3238  1 ! IMPLICIT OUTPUTS:
;2474   3239  1 !
;2475   3240  1 !       LINE_POINTER    - pointer to next character in output line
;2476   3241  1 !       EXT_LINE_LENGTH - external line length
;2477   3242  1 !       INT_LINE_LENGTH - internal line length
;2478   3243  1 !       TMS_LINE_LENGTH - length of line in TMS relative units
;2479   3244  1 !
;2480   3245  1 ! ROUTINE VALUE:
;2481   3246  1 ! COMPLETION CODES:
;2482   3247  1 !
;2483   3248  1 !       None
;2484   3249  1 !
;2485   3250  1 ! SIDE EFFECTS:
;2486   3251  1 !
;2487   3252  1 !       None
;2488   3253  1 !--
;2489   3254  2     BEGIN
;2490   3255  2     LOCAL
;2491   3256  2         WORD_POINTER,
;2492   3257  2         RNO_WORD : VECTOR [CH$ALLOCATION(1200)],
;2493   3258  2         INT_WORD_LENGTH,
;2494   3259  2         EXT_WORD_LENGTH,
;2495   3260  2         TMS_WORD_LENGTH,
;2496   3261  2         MAX_LEN,
;2497   3262  2         HYPHENATE,
;2498   3263  2         HYPH,
;2499   3264  2         INSERT_BLANK,
```

```
; 2500          3265  2          CHAR_COUNT,
; 2501          3266  2          TEXT_PTR;
; 2502          3267  2
; 2503          3268  2      IF .CMDBLK [NDX$H_FORMAT] EQL DSR
; 2504          3269  2      THEN
; 2505          3270  2          !
; 2506          3271  2          ! RUNOFF output.
; 2507          3272  2          !
; 2508          3273  2          MAX_LEN = .MAX_EXT
; 2509          3274  2      ELSE
; 2510          3275  2          !
; 2511          3276  2          ! For TMS11 or TEX output, we don't count characters.
; 2512          3277  2          !
; 2513          3278  2          ! The line length is expressed in units and each character is
; 2514          3279  2          ! assigned a length in units.
; 2515          3280  2          !
; 2516          3281  2          ! The line length is computed as follows:
; 2517          3282  2          !       (line length in characters - safety margin in characters)
; 2518          3283  2          !       * average character size
; 2519          3284  2          !
; 2520          3285  2          ! The safety margin is present to allow for slightly larger
; 2521          3286  2          ! italicized or bolded characters.
; 2522          3287  2          !
; 2523          3288  2          MAX_LEN = (.MAX_EXT - TMS_SAFETY_MARGIN) * TMSSTD;
; 2524          3289  2
; 2525          3290  2      CHAR_COUNT = .I_LEN;
; 2526          3291  2      TEXT_PTR = .I_PTR;
; 2527          3292  2
; 2528          3293  2      HYPH = FALSE;
; 2529          3294  2      INSERT_BLANK = .L_BLANK;
; 2530          3295  2
; 2531          3296  2      WHILE .CHAR_COUNT GTR 0 DO
; 2532          3297  3          BEGIN
; 2533          3298  3          LOCAL
; 2534          3299  3              L_PTR,
; 2535          3300  3              L_LEN;
; 2536          3301  3
; 2537          3302  3          CLR_RNO_WORD ();                        ! Initialize word variables
; 2538          3303  3
; 2539          3304  3          !
; 2540          3305  3          ! Get next word.
; 2541          3306  3          !
; 2542          3307  3          L_PTR = .TEXT_PTR;
; 2543          3308  3          L_LEN = .CHAR_COUNT;
; 2544          3309  3          INCR I FROM 1 TO .CHAR_COUNT DO
; 2545          3310  4              BEGIN
; 2546          3311  4              !
; 2547          3312  4              ! Skip leading whitespace if any
; 2548          3313  4              !
; 2549          3314  4              LOCAL
; 2550          3315  4                  CH;
; 2551          3316  4
; 2552          3317  4              CH = CH$RCHAR_A (L_PTR);
; 2553          3318  5              IF (.CH EQL %C' ') OR (.CH EQL TAB)
; 2554          3319  4              THEN
; 2555          3320  4                  L_LEN = .L_LEN - 1              ! One less input character
; 2556          3321  4              ELSE
```

```
;  2557     3322  5                BEGIN
;  2558     3323  5                !
;  2559     3324  5                ! End of leading whitespace
;  2560     3325  5                !
;  2561     3326  5                L_PTR = CH$PLUS (.L_PTR, -1);    ! Back up to account for overshoot
;  2562     3327  5                EXITLOOP;
;  2563     3328  4                END;
;  2564     3329  3            END;
;  2565     3330  3
;  2566     3331  3        TEXT_PTR = .L_PTR;                       ! Point past leading whitespace
;  2567     3332  3        CHAR_COUNT = .L_LEN;                     ! Update length of string
;  2568     3333  3
;  2569     3334  3        INCR I FROM 1 TO .CHAR_COUNT DO
;  2570     3335  4            BEGIN
;  2571     3336  4            !
;  2572     3337  4            ! Now copy the word to the word buffer
;  2573     3338  4            !
;  2574     3339  4            LOCAL
;  2575     3340  4                CH;
;  2576     3341  4
;  2577     3342  4            CH = CH$RCHAR_A (L_PTR);
;  2578     3343  4
;  2579     3344  4            IF .CH EQL RINTES
;  2580     3345  4            THEN
;  2581     3346  5                BEGIN
;  2582     3347  5                !
;  2583     3348  5                ! RUNOFF escape sequence
;  2584     3349  5                !
;  2585     3350  5                LOCAL
;  2586     3351  5                    OPERAND;
;  2587     3352  5
;  2588     3353  5                CH = CH$RCHAR_A (L_PTR);         ! Get function
;  2589     3354  5                OPERAND = CH$RCHAR_A (L_PTR);    ! And the operand
;  2590     3355  5                I = .I + 2;                      ! Update number of characters processed
;  2591     3356  5                L_LEN = .L_LEN - 3;              ! Update remaining character count
;  2592     3357  5
;  2593     3358  5                SELECTONE .CH OF
;  2594     3359  5                    SET
;  2595     3360  5
;  2596     3361  5                    [%C'B']:                     ! Bold
;  2597     3362  5                        NQCHAR_NC (%C'*');
;  2598     3363  5
;  2599     3364  5                    [%C'O']:                     ! Overstrike
;  2600     3365  6                        BEGIN
;  2601     3366  6                        NQCHAR_NC (.OPERAND);
;  2602     3367  6                        NQCHAR_NC (%C'^');
;  2603     3368  5                        END;
;  2604     3369  5
;  2605     3370  5                    [%C'U']:                     ! Underline
;  2606     3371  5                        NQCHAR_NC (%C'&');
;  2607     3372  5
;  2608     3373  5                    [%C'N']:
;  2609     3374  5                        IF .OPERAND EQL %C'-'
;  2610     3375  5                        THEN
;  2611     3376  6                            BEGIN                ! Hyphenate
;  2612     3377  6                            HYPHENATE = TRUE;
;  2613     3378  6                            EXITLOOP;
```

```
; 2614          3379  5                                 END;
; 2615          3380  5
; 2616          3381  5                             TES;
; 2617          3382  5                         END
; 2618          3383  4                     ELSE
; 2619          3384  5                         BEGIN
; 2620          3385  5                         !
; 2621          3386  5                         ! Not a RUNOFF escape sequence, process character
; 2622          3387  5                         !
; 2623          3388  5                         L_LEN = .L_LEN - 1;             ! One less character
; 2624          3389  5
; 2625          3390  6                         IF (.CH EQL %C' ') OR (.CH EQL TAB)
; 2626          3391  5                         THEN
; 2627          3392  5                             EXITLOOP;                   ! Whitespace - end of word
; 2628          3393  5                         !
; 2629          3394  5                         ! Bold and underline character if requested
; 2630          3395  5                         !
; 2631          3396  5                         IF .BOLD THEN NQCHAR_NC (%C'*');
; 2632          3397  5                         IF .UND  THEN NQCHAR_NC (%C'&');
; 2633          3398  5
; 2634          3399  5                         IF (.CH LSS %C' ') OR (.CH GTR %O'176')
; 2635          3400  6                         THEN
; 2636          3401  5                             !
; 2637          3402  5                             ! Control character
; 2638          3403  5                             !
; 2639          3404  5                             IF .CH NEQ %O'10'
; 2640          3405  5                             THEN
; 2641          3406  5                                 QCHAR_C (.CH)           ! Normal control character
; 2642          3407  6                             ELSE
; 2643          3408  5                                 BACKSPACE (.CH)         ! Process backspace
; 2644          3409  6                         ELSE
; 2645          3410  5                             !
; 2646          3411  5                             ! Not a control character
; 2647          3412  5                             !
; 2648          3413  5                             IF (.CH EQL %C'&')
; 2649          3414  6                             OR (.CH EQL %C'*')
; 2650          3415  6                             OR (.CH EQL %C'_')
; 2651          3416  6                             OR (.CH EQL %C'-')
; 2652          3417  6                             OR (.CH EQL %C'%')
; 2653          3418  6                             THEN
; 2654          3419  5                                 QCHAR_C (.CH)           ! RUNOFF flag so quote it
; 2655          3420  6                             ELSE
; 2656          3421  5                                 NQCHAR_C (.CH);         ! Ordinary character
; 2657          3422  5                         END;
; 2658          3423  4                     END;
; 2659          3424  3
; 2660          3425  3                 TEXT_PTR = .L_PTR;                      ! Point to next character
; 2661          3426  3                 CHAR_COUNT = .L_LEN;                    ! Update length of input string
; 2662          3427  3
; 2663          3428  3                 !
; 2664          3429  3                 ! Compute space needed if word is to be added to current line.
; 2665          3430  3                 !
; 2666          3431  3                 IF .CMDBLK [NDX$H_FORMAT] EQL DSR
; 2667          3432  3                 THEN
; 2668          3433  3                     BEGIN
; 2669          3434  4                     !
; 2670          3435  4
```

```
; 2671          3436  4              ! For RUNOFF, total = line length + word length
; 2672          3437  4              !   + 1 for comma in case this is the last word in the entry
; 2673          3438  4              !   + 1 if previous word was not terminated by a hyphenate sequence
; 2674          3439  4              !   + 1 if current word terminated with a hyphenate sequence
; 2675          3440  4              !
; 2676          3441  4              L_LEN = .EXT_LINE_LENGTH + .EXT_WORD_LENGTH + 1
; 2677          3442  4                  + (IF NOT .HYPH THEN 1 ELSE 0) + (IF .HYPHENATE THEN 1 ELSE 0);
; 2678          3443  4              END
; 2679          3444  3          ELSE
; 2680          3445  4              BEGIN
; 2681          3446  4              !
; 2682          3447  4              ! For TMS and TEX, total length = line length + word length
; 2683          3448  4              !   + length of blank in case this is the last word in the entry
; 2684          3449  4              !   + length of blank if previous word was not terminated by a hyphenate sequence
; 2685          3450  4              !   + length of hyphen if current word terminated with a hyphenate sequence
; 2686          3451  4              !
; 2687          3452  4              L_LEN = .TMS_LINE_LENGTH + .TMS_WORD_LENGTH + .CHRSIZ [%C' ']
; 2688          3453  5                  + (IF NOT .HYPH THEN .CHRSIZ [%C' '] ELSE 0)
; 2689          3454  4                  + (IF .HYPHENATE THEN .CHRSIZ [%C'-'] ELSE 0);
; 2690          3455  3              END;
; 2691          3456  3
; 2692          3457  3          IF .L_LEN GTR .MAX_LEN
; 2693          3458  3          THEN
; 2694          3459  4              BEGIN
; 2695          3460  4              !
; 2696          3461  4              ! Word will not fit on line
; 2697          3462  4              !
; 2698          3463  4              IF .HYPH
; 2699          3464  4              THEN
; 2700          3465  5                  BEGIN
; 2701          3466  5                  !
; 2702          3467  5                  ! Previous word terminated with a hyphenate sequence.
; 2703          3468  5                  ! Add trailing hyphen.
; 2704          3469  5                  !
; 2705          3470  5                  CH$WCHAR_A (%C'-', LINE_POINTER);
; 2706          3471  5                  INT_LINE_LENGTH = .INT_LINE_LENGTH + 1;
; 2707          3472  5                  EXT_LINE_LENGTH = .EXT_LINE_LENGTH + 1;
; 2708          3473  5                  TMS_LINE_LENGTH = .TMS_LINE_LENGTH + .CHRSIZ [%C'-'];
; 2709          3474  4                  END;
; 2710          3475  4
; 2711          3476  4          INS_LINE (.INT_LINE_LENGTH, CH$PTR (RNO_LINE), .EXT_LINE_LENGTH);
; 2712          3477  4
; 2713          3478  4          IF .LINE_TYPE NEQ CONT_HEAD THEN LINE_TYPE = (IF .INDLVL EQL 0 THEN ENTRY_W ELSE SUB_W);
; 2714          3479  4
; 2715          3480  4          !
; 2716          3481  4          ! Indent new line
; 2717          3482  4          !
; 2718          3483  4          INDENT_LINE (.INDENT + 4);
; 2719          3484  4
; 2720          3485  4          !
; 2721          3486  4          ! Verify word will fit on new line
; 2722          3487  4          !
; 2723          3488  4          IF .CMDBLK [NDX$H_FORMAT] EQL DSR
; 2724          3489  4          THEN
; 2725          3490  5              BEGIN
; 2726          3491  5              !
; 2727          3492  5              ! For RUNOFF...
```

```
;  2728          3493  5                !
;  2729          3494  5                L_LEN = .EXT_LINE_LENGTH + .EXT_WORD_LENGTH + 1
;  2730          3495  5                    + (IF .HYPHENATE THEN 1 ELSE 0);
;  2731          3496  5                END
;  2732          3497  4            ELSE
;  2733          3498  5                BEGIN
;  2734          3499  5                !
;  2735          3500  5                ! For TMS and TEX...
;  2736          3501  5                !
;  2737          3502  5                L_LEN = .TMS_LINE_LENGTH + .TMS_WORD_LENGTH + .CHRSIZ [%C' ']
;  2738          3503  5                    + (IF .HYPHENATE THEN .CHRSIZ [%C'-'] ELSE 0);
;  2739          3504  4                END;
;  2740          3505  4
;  2741          3506  4            IF .L_LEN GTR .MAX_LEN
;  2742          3507  4            THEN
;  2743          3508  4
;  2744        L 3509  4 %IF %BLISS (BLISS32)
;  2745          3510  4 %THEN                                           ! Signal errors in BLISS32
;  2746          3511  4
;  2747          3512  4                SIGNAL (INDEX$_DOESNTFIT, 2, .INT_WORD_LENGTH, CH$PTR (RNO_WORD));
;  2748          3513  4
;  2749        U 3514  4 %ELSE                                           ! Use $XPO_PUT_MSG otherwise
;  2750        U 3515  4
;  2751        U 3516  4                $XPO_PUT_MSG (SEVERITY = WARNING,
;  2752        U 3517  4                    STRING = $STR_CONCAT ('the word ''', (.INT_WORD_LENGTH, CH$PTR (RNO_WORD)),
;  2753        U 3518  4                        ''' will not fit at current indentation level'));
;  2754        U 3519  4
;  2755          3520  4 %FI
;  2756          3521  4
;  2757          3522  4            END
;  2758          3523  3        ELSE
;  2759          3524  4            BEGIN
;  2760          3525  4            !
;  2761          3526  4            ! Word will fit on line.
;  2762          3527  4            !
;  2763          3528  4            IF CH$NEQ (1, .BLANKS, .INT_LINE_LENGTH, CH$PTR (RNO_LINE), %C' ')
;  2764          3529  5            AND (NOT .HYPH)
;  2765          3530  4            AND .INSERT_BLANK
;  2766          3531  4            THEN
;  2767          3532  5                BEGIN
;  2768          3533  5                !
;  2769          3534  5                ! Line is non-blank, previous word did not terminate
;  2770          3535  5                ! with a hyphenate sequence, and we are supposed to insert a blank.
;  2771          3536  5                !
;  2772          3537  5                ! Add a blank to the line before adding the word.
;  2773          3538  5                !
;  2774          3539  5                CH$WCHAR_A (%C' ', LINE_POINTER);
;  2775          3540  5                INT_LINE_LENGTH = .INT_LINE_LENGTH + 1;
;  2776          3541  5                EXT_LINE_LENGTH = .EXT_LINE_LENGTH + 1;
;  2777          3542  5                TMS_LINE_LENGTH = .TMS_LINE_LENGTH + .CHRSIZ [%C' '];
;  2778          3543  4                END;
;  2779          3544  3            END;
;  2780          3545  3
;  2781          3546  3        !
;  2782          3547  3        ! Append word to line
;  2783          3548  3        !
;  2784          3549  3        CH$MOVE (.INT_WORD_LENGTH, CH$PTR (RNO_WORD), .LINE_POINTER);
```

```
;  2785          3550  3          LINE_POINTER = CH$PLUS (.LINE_POINTER, .INT_WORD_LENGTH);
;  2786          3551  3          INT_LINE_LENGTH = .INT_LINE_LENGTH + .INT_WORD_LENGTH;
;  2787          3552  3          EXT_LINE_LENGTH = .EXT_LINE_LENGTH + .EXT_WORD_LENGTH;
;  2788          3553  3          TMS_LINE_LENGTH = .TMS_LINE_LENGTH + .TMS_WORD_LENGTH;
;  2789          3554  3
;  2790          3555  3          HYPH = .HYPHENATE;                  ! Save hyphenation status of current word
;  2791          3556  3          INSERT_BLANK = TRUE;
;  2792          3557  2          END;
;  2793          3558  2
;  2794          3559  1      END;


                                                OFFC 00000 FILL_LINE:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11    ; 3209
                                5E      FB40    CE  9E 00002            MOVAB   -1216(SP), SP
                                01 00000000G    EF  B1 00007            CMPW    CMDBLK+4, #1                    ; 3268
                                                06  12 0000E            BNEQ    1$
                                6E        10    AC  D0 00010            MOVL    MAX_EXT, MAX_LEN                ; 3273
                                                0D  11 00014            BRB     2$
                      50        10    AC              04  C3 00016 1$:  SUBL3   #4, MAX_EXT, R0                ; 3288
                      6E              50 00000000G    8F  C5 0001B      MULL3   #TMSSTD, R0, MAX_LEN
                                59        04    AC  D0 00023 2$:  MOVL    I_LEN, CHAR_COUNT               ; 3290
                                04    AE  08    AC  D0 00027            MOVL    I_PTR, TEXT_PTR                 ; 3291
                                                56  D4 0002C            CLRL    HYPH                            ; 3293
                                08    AE  1C    AC  D0 0002E            MOVL    L_BLANK, INSERT_BLANK           ; 3294
                                                59  D5 00033 3$:  TSTL    CHAR_COUNT                      ; 3296
                                                01  14 00035            BGTR    4$
                                                    04 00037            RET
                                57        10    AE  9E 00038 4$:  MOVAB   RNO_WORD, WORD_POINTER          ; 3302
                                                58  D4 0003C            CLRL    INT_WORD_LENGTH
                                                5A  7C 0003E            CLRQ    TMS_WORD_LENGTH
                                          0C    AE  D4 00040            CLRL    HYPHENATE
                                51        04    AE  D0 00043            MOVL    TEXT_PTR, L_PTR                 ; 3307
                                53              59  D0 00047            MOVL    CHAR_COUNT, L_LEN               ; 3308
                                                50  D4 0004A            CLRL    I                               ; 3309
                                                19  11 0004C            BRB     8$
                                52              81  9A 0004E 5$:  MOVZBL  (L_PTR)+, CH                    ; 3317
                                20              52  D1 00051            CMPL    CH, #32                         ; 3318
                                                09  13 00054            BEQL    6$
                   00000000G    8F              52  D1 00056            CMPL    CH, #TAB
                                                04  12 0005D            BNEQ    7$
                                                53  D7 0005F 6$:  DECL    L_LEN                           ; 3320
                                                04  11 00061            BRB     8$
                                                51  D7 00063 7$:  DECL    L_PTR                           ; 3326
                                                04  11 00065            BRB     9$                              ; 3322
                   E3           50              59  F3 00067 8$:  AOBLEQ  CHAR_COUNT, I, 5$                ; 3309
                          04    AE              51  D0 0006B 9$:  MOVL    L_PTR, TEXT_PTR                 ; 3331
                                59              53  D0 0006F            MOVL    L_LEN, CHAR_COUNT               ; 3332
                                                54  D4 00072            CLRL    I                               ; 3334
                                              00E7  31 00074 10$:  BRW     27$
                                50                  81  9A 00077 11$:  MOVZBL  (L_PTR)+, CH                    ; 3342
                   00000000G    8F                  50  D1 0007A            CMPL    CH, #RINTES                     ; 3344
                                                    50  12 00081            BNEQ    16$
                                50                  81  9A 00083            MOVZBL  (L_PTR)+, CH                    ; 3353
```

```
                          52              81  9A 00086          MOVZBL  (L_PTR)+, OPERAND                       ; 3354
                          54              02  C0 00089          ADDL2   #2, I                                   ; 3355
                          53              03  C2 0008C          SUBL2   #3, L_LEN                               ; 3356
                00000042  8F              50  D1 0008F          CMPL    CH, #66                                 ; 3361
                                          05  12 00096          BNEQ    12$
                          67              2A  90 00098          MOVB    #42, (WORD_POINTER)                     ; 3362
                                          7E  11 0009B          BRB     20$
                0000004F  8F              50  D1 0009D 12$:     CMPL    CH, #79                                 ; 3364
                                          0A  12 000A4          BNEQ    13$
                          87              52  90 000A6          MOVB    OPERAND, (WORD_POINTER)+                ; 3366
                                          58  D6 000A9          INCL    INT_WORD_LENGTH
                          67              25  90 000AB          MOVB    #37, (WORD_POINTER)                     ; 3367
                                          78  11 000AE          BRB     22$
                00000055  8F              50  D1 000B0 13$:     CMPL    CH, #85                                 ; 3370
                                          05  12 000B7          BNEQ    14$
                          67              26  90 000B9          MOVB    #38, (WORD_POINTER)                     ; 3371
                                          6A  11 000BC          BRB     22$
                0000004E  8F              50  D1 000BE 14$:     CMPL    CH, #78                                 ; 3373
                                          AD  12 000C5          BNEQ    10$
                          2D              52  D1 000C7          CMPL    OPERAND, #45                            ; 3374
                                          A8  12 000CA          BNEQ    10$
                       0C AE              01  D0 000CC          MOVL    #1, HYPHENATE                           ; 3377
                                    0091  31  00D0 15$:         BRW     28$                                     ; 3376
                                          53  D7 000D3 16$:     DECL    L_LEN                                   ; 3388
                          20              50  D1 000D5          CMPL    CH, #32                                 ; 3390
                                          F6  13 000D8          BEQL    15$
                00000000G 8F              50  D1 000DA          CMPL    CH, #TAB
                                          ED  13 000E1          BEQL    15$
                          05        14    AC  E9 000E3          BLBC    BOLD, 17$                               ; 3397
                          87              2A  90 000E7          MOVB    #42, (WORD_POINTER)+
                                          58  D6 000EA          INCL    INT_WORD_LENGTH
                          05        18    AC  E9 000EC 17$:     BLBC    UND, 18$                                ; 3398
                          87              26  90 000F0          MOVB    #38, (WORD_POINTER)+
                                          58  D6 000F3          INCL    INT_WORD_LENGTH
                          20              50  D1 000F5 18$:     CMPL    CH, #32                                 ; 3400
                                          09  19 000F8          BLSS    19$
                0000007E  8F              50  D1 000FA          CMPL    CH, #126
                                          27  15 00101          BLEQ    23$
                          08              50  D1 00103 19$:     CMPL    CH, #8                                  ; 3405
                                          15  13 00106          BEQL    21$
                          87        5F    8F  90 00108          MOVB    #95, (WORD_POINTER)+                    ; 3407
                                          58  D6 0010C          INCL    INT_WORD_LENGTH
                          67              50  90 0010E          MOVB    CH, (WORD_POINTER)
                                          5B  D6 00111          INCL    EXT_WORD_LENGTH
                          5A 00000000GFF  40  C0 00113          ADDL2   @CHRSIZ[CH], TMS_WORD_LENGTH
                                          3D  11 0011B 20$:     BRB     26$                                     ; 3405
                          87        5F    8F  90 0011D 21$:     MOVB    #95, (WORD_POINTER)+                    ; 3409
                                          58  D6 00121          INCL    INT_WORD_LENGTH
                          67              50  90 00123          MOVB    CH, (WORD_POINTER)
                                          5B  D7 00126          DECL    EXT_WORD_LENGTH
                                          30  11 00128 22$:     BRB     26$                                     ; 3405
                          26              50  D1 0012A 23$:     CMPL    CH, #38                                 ; 3414
                                          18  13 0012D          BEQL    24$
                          2A              50  D1 0012F          CMPL    CH, #42                                 ; 3415
                                          13  13 00132          BEQL    24$
                0000005F  8F              50  D1 00134          CMPL    CH, #95                                 ; 3416
                                          0A  13 0013B          BEQL    24$
```

```
                                2E              50  D1 0013D          CMPL    CH, #46                                         ; 3417
                                                05  13 00140          BEQL    24$
                                25              50  D1 00142          CMPL    CH, #37                                         ; 3418
                                                06  12 00145          BNEQ    25$
                                87        5F    8F  90 00147 24$:     MOVB    #95, (WORD_POINTER)+                            ; 3420
                                                58  D6 0014B          INCL    INT_WORD_LENGTH
                                67              50  90 0014D 25$:     MOVB    CH, (WORD_POINTER)                              ; 3422
                                5A 00000000GFF  40  C0 00150          ADDL2   @CHRSIZ[CH], TMS_WORD_LENGTH
                                                5B  D6 00158          INCL    EXT_WORD_LENGTH                                 ; 3420
                                                57  D6 0015A 26$:     INCL    WORD_POINTER                                    ; 3407
                                                58  D6 0015C          INCL    INT_WORD_LENGTH
        FF13          54        01              59  F1 0015E 27$:     ACBL    CHAR_COUNT, #1, I, 11$                          ; 3334
                        04      AE              51  D0 00164 28$:     MOVL    L_PTR, TEXT_PTR                                 ; 3426
                                59              53  D0 00168          MOVL    L_LEN, CHAR_COUNT                               ; 3427
                                54              56  D2 0016B          MCOML   HYPH, R4                                        ; 3442
                                01 00000000G    EF  B1 0016E          CMPW    CMDBLK+4, #1                                    ; 3432
                                                27  12 00175          BNEQ    33$
                      50        5B 00000000'    EF  C1 00177          ADDL3   EXT_LINE_LENGTH, EXT_WORD_LENGTH, R0            ; 3441
                                05              54  E9 0017F          BLBC    R4, 29$                                         ; 3442
                                51              01  D0 00182          MOVL    #1, R1
                                                02  11 00185          BRB     30$
                                                51  D4 00187 29$:     CLRL    R1
                                50              51  C0 00189 30$:     ADDL2   R1, R0
                                05        0C    AE  E9 0018C          BLBC    HYPHENATE, 31$
                                51              01  D0 00190          MOVL    #1, R1
                                                02  11 00193          BRB     32$
                                                51  D4 00195 31$:     CLRL    R1
                                53        01 A140  9E 00197 32$:     MOVAB   1(R1)[R0], L_LEN
                                                34  11 0019C          BRB     38$                                             ; 3432
                      51        5A 00000000'    EF  C1 0019E 33$:     ADDL3   TMS_LINE_LENGTH, TMS_WORD_LENGTH, R1            ; 3452
                                50 00000000G    EF  D0 001A6          MOVL    CHRSIZ, R0
                                51      0080    C0  C0 001AD          ADDL2   128(R0), R1
                                07              54  E9 001B2          BLBC    R4, 34$                                         ; 3453
                                52      0080    C0  D0 001B5          MOVL    128(R0), R2
                                                02  11 001BA          BRB     35$
                                                52  D4 001BC 34$:     CLRL    R2
                                51              52  C0 001BE 35$:     ADDL2   R2, R1
                                07        0C    AE  E9 001C1          BLBC    HYPHENATE, 36$                                  ; 3454
                                50      00B4    C0  D0 001C5          MOVL    180(R0), R0
                                                02  11 001CA          BRB     37$
                                                50  D4 001CC 36$:     CLRL    R0
        53                      51              50  C1 001CE 37$:     ADDL3   R0, R1, L_LEN
                                6E              53  D1 001D2 38$:     CMPL    L_LEN, MAX_LEN                                  ; 3457
                                                03  14 001D5          BGTR    39$
                                              00D2  31 001D7          BRW     50$
                                29              56  E9 001DA 39$:     BLBC    HYPH, 40$                                       ; 3463
                00000000'       FF              2D  90 001DD          MOVB    #45, @LINE_POINTER                              ; 3470
                                    00000000'   EF  D6 001E4          INCL    LINE_POINTER
                                    00000000'   EF  D6 001EA          INCL    INT_LINE_LENGTH                                 ; 3471
                                    00000000'   EF  D6 001F0          INCL    EXT_LINE_LENGTH                                 ; 3472
                                50 00000000G    EF  D0 001F6          MOVL    CHRSIZ, R0                                      ; 3473
                00000000'       EF      00B4    C0  C0 001FD          ADDL2   180(R0), TMS_LINE_LENGTH
                                    00000000'   EF  DD 00206 40$:     PUSHL   EXT_LINE_LENGTH                                 ; 3476
                                    00000000'   EF  9F 0020C          PUSHAB  RNO_LINE
                                    00000000'   EF  DD 00212          PUSHL   INT_LINE_LENGTH
                    F997        CF              03  FB 00218          CALLS   #3, INS_LINE
                                0B 00000000'    EF  D1 0021D          CMPL    LINE_TYPE, #11                                  ; 3478
```

```
                                        17  13 00224          BEQL    43$                                             ;
                            00000000G   EF  D5 00226          TSTL    INDLVL                                          ;
                                        05  12 0022C          BNEQ    41$                                             ;
                         50             06  D0 0022E          MOVL    #6, R0                                          ;
                                        03  11 00231          BRB     42$                                             ;
                         50             09  D0 00233 41$:     MOVL    #9, R0                                          ;
                00000000'   EF          50  D0 00236 42$:     MOVL    R0, LINE_TYPE                                   ;
             7E       0C    AC          04  C1 0023D 43$:     ADDL3   #4, INDENT, -(SP)                               ; 3483
                00000000V   EF          01  FB 00242          CALLS   #1, INDENT_LINE                                 ;
                         01 00000000G   EF  B1 00249          CMPW    CMDBLK+4, #1                                    ; 3488
                                        1A  12 00250          BNEQ    46$                                             ;
             50                5B 00000000' EF  C1 00252      ADDL3   EXT_LINE_LENGTH, EXT_WORD_LENGTH, R0            ; 3494
                         05       0C    AE  E9 0025A          BLBC    HYPHENATE, 44$                                  ; 3495
                         51             01  D0 0025E          MOVL    #1, R1                                          ;
                                        02  11 00261          BRB     45$                                             ;
                                        51  D4 00263 44$:     CLRL    R1                                              ;
                         53       01 A140  9E 00265 45$:      MOVAB   1(R1)[R0], L_LEN                                ;
                                        25  11 0026A          BRB     49$                                             ; 3488
             51                5A 00000000' EF  C1 0026C 46$: ADDL3   TMS_LINE_LENGTH, TMS_WORD_LENGTH, R1            ; 3502
                               50 00000000G EF  D0 00274      MOVL    CHRSIZ, R0                                      ;
                               51    0080   C0  C0 0027B      ADDL2   128(R0), R1                                     ;
                               07      0C   AE  E9 00280      BLBC    HYPHENATE, 47$                                  ; 3503
                               50    00B4   C0  D0 00284      MOVL    180(R0`, R0                                     ;
                                        02  11 00289          BRB     48$                                             ;
                                        50  D4 0028B 47$:     CLRL    R0                                              ;
             53                51       50  C1 0028D 48$:     ADDL3   R0, R1, L_LEN                                   ;
                               6E       53  D1 00291 49$:     CMPL    L_LEN, MAX_LEN                                  ; 3506
                                        5A  15 00294          BLEQ    51$                                             ;
                                  10    AE  9F 00296          PUSHAB  RNO_WORD                                        ; 3512
                                        58  DD 00299          PUSHL   INT_WORD_LENGTH                                 ;
                                        02  DD 0029B          PUSHL   #2                                              ;
                            00000000G   8F  DD 0029D          PUSHL   #DSRINDEX$_DOESNTFIT                            ;
                00000000G   00          04  FB 002A3          CALLS   #4, LIB$SIGNAL                                  ;
                                        44  11 002AA          BRB     51$                                             ; 3457
00000000'  EF         20 00000000'  FF  01  2D 002AC 50$:     CMPC5   #1, @BLANKS, #32, INT_LINE_LENGTH, RNO_LINE     ; 3528
                            00000000'   EF     002B9                                                                  ;
                                        30  13 002BE          BEQL    51$                                             ;
                               2D       54  E9 002C0          BLBC    R4, 51$                                         ; 3529
                               29    08 AE  E9 002C3          BLBC    INSERT_BLANK, 51$                               ; 3530
                00000000'   FF          20  90 002C7          MOVB    #32, @LINE_POINTER                              ; 3539
                            00000000'   EF  D6 002CE          INCL    LINE_POINTER                                    ;
                            00000000'   EF  D6 002D4          INCL    INT_LINE_LENGTH                                 ; 3540
                            00000000'   EF  D6 002DA          INCL    EXT_LINE_LENGTH                                 ; 3541
                         50 00000000G   EF  D0 002E0          MOVL    CHRSIZ, R0                                      ; 3542
                00000000'   EF    0080  C0  C0 002E7          ADDL2   128(R0), TMS_LINE_LENGTH                        ;
       00000000'  FF      10    AE      58  28 002F0 51$:     MOVC3   INT_WORD_LENGTH, RNO_WORD, @LINE_POINTER        ; 3549
                00000000'   EF          58  C0 002F9          ADDL2   INT_WORD_LENGTH, LINE_POINTER                   ; 3550
                00000000'   EF          58  C0 00300          ADDL2   INT_WORD_LENGTH, INT_LINE_LENGTH                ; 3551
                00000000'   EF          5B  C0 00307          ADDL2   EXT_WORD_LENGTH, EXT_LINE_LENGTH                ; 3552
                00000000'   EF          5A  C0 0030E          ADDL2   TMS_WORD_LENGTH, TMS_LINE_LENGTH                ; 3553
                         56       0C    AE  D0 00315          MOVL    HYPHENATE, HYPH                                 ; 3555
                  08     AE             01  D0 00319          MOVL    #1, INSERT_BLANK                                ; 3556
                                  FD13  31 0031D              BRW     3$                                              ; 3296
                                        04  00320             RET                                                     ; 3559

; Routine Size:  801 bytes,    Routine Base:  $CODE$ + 0F3C
```

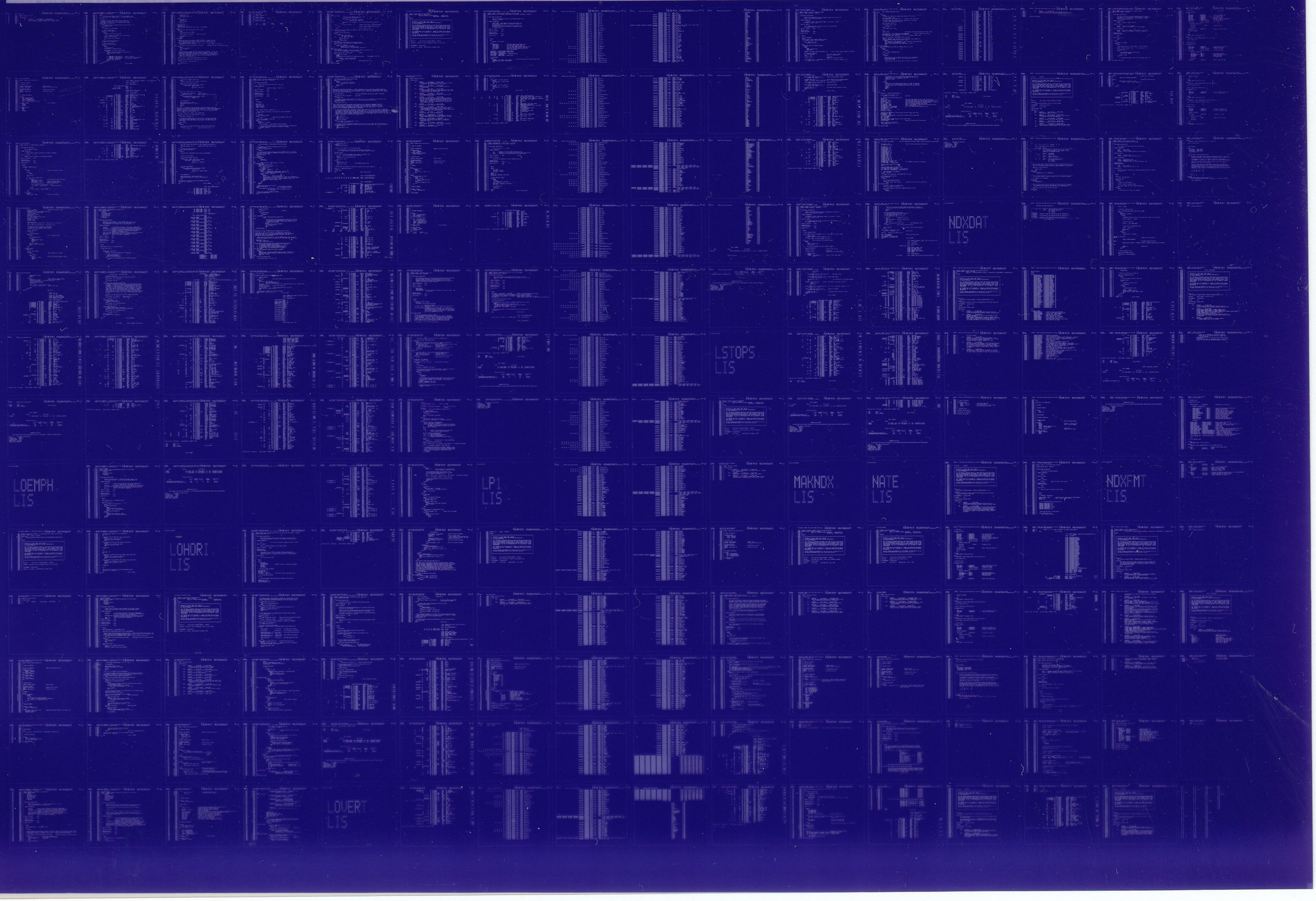
```
;  2796          3560  1 %SBTTL 'INDENT_LINE -- Indent new line'
;  2797          3561  1 ROUTINE INDENT_LINE (INDENT) : NOVALUE =
;  2798          3562  1 !++
;  2799          3563  1 !
;  2800          3564  1 ! FUNCTIONAL DESCRIPTION:
;  2801          3565  1 !
;  2802          3566  1 !       This routine clears out RNO_LINE and fills it with the number
;  2803          3567  1 !       of blanks specified by INDENT.
;  2804          3568  1 !
;  2805          3569  1 ! FORMAL PARAMETERS:
;  2806          3570  1 !
;  2807          3571  1 !       INDENT - Number of blanks to indent line
;  2808          3572  1 !
;  2809          3573  1 ! IMPLICIT INPUTS:
;  2810          3574  1 !
;  2811          3575  1 !       CMDBLK              - Command line information block
;  2812          3576  1 !       TMSSTD              - Standard TMS character size
;  2813          3577  1 !       MSPACE              - Size of /m sequence
;  2814          3578  1 !
;  2815          3579  1 ! IMPLICIT OUTPUTS:
;  2816          3580  1 !
;  2817          3581  1 !       EXT_LINE_LENGTH = .INDENT
;  2818          3582  1 !       INT_LINE_LENGTH = .INDENT
;  2819          3583  1 !       LINE_POINTER    = Pointer to next character position in RNO_LINE
;  2820          3584  1 !       TMS_LINE_LENGTH = Length of line in TMS relative units
;  2821          3585  1 !
;  2822          3586  1 ! ROUTINE VALUE:
;  2823          3587  1 ! COMPLETION CODES:
;  2824          3588  1 !
;  2825          3589  1 !       None
;  2826          3590  1 !
;  2827          3591  1 ! SIDE EFFECTS:
;  2828          3592  1 !
;  2829          3593  1 !       None
;  2830          3594  1 !--
;  2831          3595  2     BEGIN
;  2832          3596  2     LOCAL
;  2833          3597  2         IND;
;  2834          3598  2
;  2835          3599  2     IND = .INDENT;
;  2836          3600  2     CLR_RNO_LINE ();
;  2837          3601  2     EXT_LINE_LENGTH = .IND;
;  2838          3602  2
;  2839          3603  3     IF  (.IND GEQ .CMDBLK [NDX$G_COLUMN_WID] + .CMDBLK [NDX$G_GUTTER_WID])
;  2840          3604  3     AND ((.CMDBLK [NDX$H_FORMAT] EQL TMS11_A) OR (.CMDBLK [NDX$H_FORMAT] EQL TMS11_E))
;  2841          3605  2     THEN
;  2842          3606  3         BEGIN
;  2843          3607  3         !
;  2844          3608  3         ! TMS11 output and padding out to second column
;  2845          3609  3         ! for /MASTER/LAYOUT=SEPARATE
;  2846          3610  3         !
;  2847          3611  3         CH$WCHAR_A (TAB, LINE_POINTER);         ! Use tabs to get to second column
;  2848          3612  3         CH$WCHAR_A (TAB, LINE_POINTER);
;  2849          3613  3         INT_LINE_LENGTH = .INT_LINE_LENGTH + 2;
;  2850          3614  3
;  2851          3615  3         IND = .IND - .CMDBLK [NDX$G_COLUMN_WID] - .CMDBLK [NDX$G_GUTTER_WID];
;  2852          3616  3
```

```
; 2853        3617  3          TMS_LINE_LENGTH = (.CMDBLK [NDX$G_COLUMN_WID] + .CMDBLK [NDX$G_GUTTER_WID]) * TMSSTD
; 2854        3618  3              + (.IND / 2) * MSPACE;
; 2855        3619  3          END
; 2856        3620  2      ELSE
; 2857        3621  2          TMS_LINE_LENGTH = (.IND / 2) * MSPACE;
; 2858        3622  2
; 2859        3623  2      CH$FILL (%C' ', .IND, .LINE_POINTER);
; 2860        3624  2      INT_LINE_LENGTH = .INT_LINE_LENGTH + .IND;
; 2861        3625  2      LINE_POINTER = CH$PLUS (CH$PTR (RNO_LINE), .INT_LINE_LENGTH);
; 2862        3626  1      END;


                                        03FC 00000 INDENT_LINE:
                                                   .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9                    ; 3561
                          59 00000000G   8F   D0 00002     MOVL    #MSPACE, R9
                          58 00000000G   EF   9E 00009     MOVAB   CMDBLK+16, R8
                          57 00000000'   EF   9E 00010     MOVAB   LINE_POINTER, R7
                          56       04    AC   D0 00017     MOVL    INDENT, IND                             ; 3599
                          67       04    A7   9E 0001B     MOVAB   RNO_LINE, LINE_POINTER                  ; 3600
                                 04B4    C7   7C 0001F     CLRQ    INT_LINE_LENGTH
                                 04BC    C7   D4 00023     CLRL    TMS_LINE_LENGTH
                               04B8      C7               56 D0 00027     MOVL    IND, EXT_LINE_LENGTH    ; 3601
                    51           FC      A8               68 C1 0002C     ADDL3   CMDBLK+16, CMDBLK+12, R1 ; 3603
                                 51                       56 D1 00031     CMPL    IND, R1
                                                          3E 19 00034     BLSS    2$
                                 02        F4             A8 B1 00036     CMPW    CMDBLK+4, #2            ; 3604
                                                          06 13 0003A     BEQL    1$
                                 03        F4             A8 B1 0003C     CMPW    CMDBLK+4, #3
                                                          32 12 00040     BNEQ    2$
                                    00     B7        00G    8F 90 00042 1$:  MOVB    #TAB, @LINE_POINTER     ; 3611
                                                          67 D6 00047     INCL    LINE_POINTER
                                    00     B7        00G    8F 90 00049     MOVB    #TAB, @LINE_POINTER     ; 3612
                                                          67 D6 0004E     INCL    LINE_POINTER
                                 04B4      C7               02 C0 00050     ADDL2   #2, INT_LINE_LENGTH     ; 3613
                          50                56        FC     A8 C3 00055     SUBL3   CMDBLK+12, IND, R0      ; 3615
                          56                50               68 C3 0005A     SUBL3   CMDBLK+16, R0, IND
                                            51 00000000G    8F C4 0005E     MULL2   #TMSSTD, R1             ; 3617
                          50                56               02 C7 00065     DIVL3   #2, IND, R0             ; 3618
                                            50               59 C4 00069     MULL2   R9, R0
                               04BC       C7               51 50 C1 0006C     ADDL3   R0, R1, TMS_LINE_LENGTH
                                                          0A 11 00072     BRB     3$                      ; 3603
                          50                56               02 C7 00074 2$:  DIVL3   #2, IND, R0     ; 3621
                               04BC       C7               50 59 C5 00078     MULL3   R9, R0, TMS_LINE_LENGTH
                    56                    20               6E 00 2C 0007E 3$:  MOVC5   #0, (SP), #32, IND, @LINE_POINTER ; 3623
                                                       00 B7       00083
                                 04B4      C7               56 C0 00085     ADDL2   IND, INT_LINE_LENGTH    ; 3624
                                            50        04     A7 9E 0008A     MOVAB   RNO_LINE, R0            ; 3625
                                            67     04B4 D740    9E 0008E     MOVAB   @INT_LINE_LENGTH[R0], LINE_POINTER
                                                          04 00094     RET                             ; 3626

; Routine Size:  149 bytes,    Routine Base:  $CODE$ + 125D
```

```
; 2864          3627  1 %SBTTL 'PADLIN -- Pad line to column width'
; 2865          3628  1 GLOBAL ROUTINE PADLIN (INT_LEN, PTR, EXT_LEN, DSC) : NOVALUE =
; 2866          3629  1 !++
; 2867          3630  1 !
; 2868          3631  1 ! FUNCTIONAL DESCRIPTION:
; 2869          3632  1 !
; 2870          3633  1 !       This routine pads the input line to the column width
; 2871          3634  1 !
; 2872          3635  1 ! FORMAL PARAMETERS:
; 2873          3636  1 !
; 2874          3637  1 !       INT_LEN - Internal length of line
; 2875          3638  1 !       PTR     - CH$PTR to line
; 2876          3639  1 !       EXT_LEN - Number of print positions used by line
; 2877          3640  1 !       DSC     - Address of string descriptor where padded line is returned
; 2878          3641  1 !
; 2879          3642  1 ! IMPLICIT INPUTS:
; 2880          3643  1 !
; 2881          3644  1 !       CMDBLK [NDX$G_GUTTER_WID] - Gutter width
; 2882          3645  1 !       CMDBLK [NDX$G_COLUMN_WID] - Column width
; 2883          3646  1 !
; 2884          3647  1 ! IMPLICIT OUTPUTS:
; 2885          3648  1 !
; 2886          3649  1 !       None
; 2887          3650  1 !
; 2888          3651  1 ! ROUTINE VALUE:
; 2889          3652  1 ! COMPLETION CODES:
; 2890          3653  1 !
; 2891          3654  1 !       None
; 2892          3655  1 !
; 2893          3656  1 ! SIDE EFFECTS:
; 2894          3657  1 !
; 2895          3658  1 !       None
; 2896          3659  1 !--
; 2897          3660  2     BEGIN
; 2898          3661  2
; 2899          3662  2     IF .CMDBLK [NDX$H_FORMAT] NEQ DSR           ! For TMS and TEX output
; 2900          3663  3     OR (.CMDBLK [NDX$H_LAYOUT] NEQ TWO_COLUMN)  ! For all output except TWO_COLUMN
; 2901          3664  3     OR (.EXT_LEN EQL 0)                         ! Or previously padded output
; 2902          3665  3     OR (.EXT_LEN GTR .CMDBLK [NDX$G_COLUMN_WID])
; 2903          3666  2     THEN
; 2904          3667  2         !
; 2905          3668  2         ! Line does not need padding. Just copy it.
; 2906          3669  2         !
; 2907          3670  3         $STR_COPY (STRING = (.INT_LEN, .PTR), TARGET = .DSC)
; 2908          3671  2     ELSE
; 2909          3672  3         BEGIN
; 2910          3673  3         !
; 2911          3674  3         ! Compute output length =
; 2912          3675  3         !       internal length + (column width - external length) + gutter width
; 2913          3676  3         !
; 2914          3677  3         LOCAL
; 2915          3678  3             LINE : VECTOR [CH$ALLOCATION (1200)],
; 2916          3679  3             LEN;
; 2917          3680  3
; 2918          3681  3         LEN = .INT_LEN + (.CMDBLK [NDX$G_COLUMN_WID] - .EXT_LEN) + .CMDBLK [NDX$G_GUTTER_WID];
; 2919          3682  3
; 2920          3683  3         CH$COPY (.INT_LEN, .PTR, %C' ', .LEN, CH$PTR (LINE));
```

```
;  2921          3684  3          $STR_COPY (STRING = (.LEN, CH$PTR (LINE)), TARGET = .DSC);
;  2922          3685  2          END;
;  2923          3686  2
;  2924          3687  1      END;


                                                    01FC 00000          .ENTRY  PADLIN, Save R2,R3,R4,R5,R6,R7,R8          ; 3628
                                            58 00000000G    EF   9E 00002       MOVAB   STR$FAILURE, R8                    ;
                                            57 00000000G    EF   9E 00009       MOVAB   CMDBLK+4, R7                       ;
                                            5E      FB48    CE   9E 00010       MOVAB   -1208(SP), SP                      ;
                                            01              67   B1 00015       CMPW    CMDBLK+4, #1                       ; 3662
                                                            12   12 00018       BNEQ    1$                                 ;
                                            01      02      A7   B1 0001A       CMPW    CMDBLK+6, #1                       ; 3663
                                                            0C   12 0001E       BNEQ    1$                                 ;
                                                    0C      AC   D5 00020       TSTL    EXT_LEN                            ; 3664
                                                            07   13 00023       BEQL    1$                                 ;
                                            08      A7      0C   AC   D1 00025       CMPL    EXT_LEN, CMDBLK+12             ; 3665
                                                            1E   15 0002A       BLEQ    2$                                 ;
                                            F8      AD      04   AC   B0 0002C 1$:     MOVW    INT_LEN, $STR$STRING           ; 3670
                                            FA      AD              0E   90 00031       MOVB    #14, $STR$STRING+2         ;
                                            FB      AD              01   90 00035       MOVB    #1, $STR$STRING+3          ;
                                            FC      AD      08      AC   D0 00039       MOVL    PTR, $STR$STRING+4         ;
                                                                    58   DD 0003E       PUSHL   R8                         ;
                                                                    7E   D4 00040       CLRL    -(SP)                      ;
                                                            10      AC   DD 00042       PUSHL   DSC                        ;
                                                            F8      AD   9F 00045       PUSHAB  $STR$STRING                ;
                                                                    32   11 00048       BRB     3$                         ;
                                                    50      08  A7      0C   AC   C3 0004A 2$:     SUBL3   EXT_LEN, CMDBLK+12, R0 ; 3681
                                                                    50   04   AC   C0 00050       ADDL2   INT_LEN, R0            ;
                                                    56              50   0C   A7   C1 00054       ADDL3   CMDBLK+16, R0, LEN     ;
                                        56          20      08  BC      04   AC   2C 00059       MOVC5   INT_LEN, @PTR, #32, LEN, LINE ; 3683
                                                                        08   AE        00060                                    ;
                                                            6E              56   B0 00062       MOVW    LEN, $STR$STRING       ; 3684
                                                    02      AE              0E   90 00065       MOVB    #14, $STR$STRING+2     ;
                                                    03      AE              01   90 00069       MOVB    #1, $STR$STRING+3      ;
                                                    04      AE      08      AE   9E 0006D       MOVAB   LINE, $STR$STRING+4    ;
                                                                    58   DD 00072       PUSHL   R8                         ;
                                                                    7E   D4 00074       CLRL    -(SP)                      ;
                                                            10      AC   DD 00076       PUSHL   DSC                        ;
                                                            0C      AE   9F 00079       PUSHAB  $STR$STRING                ;
                                                                    7E   D4 0007C 3$:     CLRL    -(SP)                      ;
                                                00000000G   EF              05   FB 0007E       CALLS   #5, XST$COPY           ;
                                                                            04   00085       RET                            ; 3687

; Routine Size:  134 bytes,    Routine Base:  $CODE$ + 12F2

;  2925          3688  1
;  2926          3689  1 END                                   !End of module
;  2927          3690  0 ELUDOM
```

```
                                .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY
;
;       Name                    Bytes                   Attributes
;
;  $PLIT$                        972  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                        1428  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                       4984  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;
;                               Library Statistics
;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]XPORT.L32;1         590       163        27       252        00:00.1
;
;                                       COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:NDXFMT/OBJ=OBJ$:NDXFMT MSRC$:NDXFMT/UPDATE=(ENH$:NDXFMT)
;
; Size:          4984 code + 2400 data bytes
; Run Time:        02:25.3
; Elapsed Time:    04:02.8
; Lines/CPU Min:    1523
; Lexemes/CPU-Min: 64870
; Memory Used:  556 pages
; Compilation Complete
```



0344 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

NDXMSG
LIS

NDXOUT
LIS

NDXINI
LIS