```
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
RRRRRRRRRRRR     UUU          UUU  NNN  NNN     NNN  000         000  FFFFFFFFFFFF     FFFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN  NNN     NNN  000         000  FFFFFFFFFFFF     FFFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN  NNN     NNN  000         000  FFFFFFFFFFFF     FFFFFFFFFFFF
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
STR
STR
```

```
IIIIIII    NN      NN  DDDDDDDD    EEEEEEEEEE  XX      XX
IIIIIII    NN      NN  DDDDDDDD    EEEEEEEEEE  XX      XX
  II       NN      NN  DD      DD  EE          XX      XX
  II       NN      NN  DD      DD  EE          XX      XX
  II       NNNN    NN  DD      DD  EE            XX  XX
  II       NNNN    NN  DD      DD  EE            XX  XX
  II       NN  NN  NN  DD      DD  EEEEEEEE        XX
  II       NN  NN  NN  DD      DD  EEEEEEEE        XX
  II       NN    NNNN  DD      DD  EE            XX  XX
  II       NN    NNNN  DD      DD  EE            XX  XX
  II       NN      NN  DD      DD  EE          XX      XX      ....
  II       NN      NN  DD      DD  EE          XX      XX      ....
IIIIIII    NN      NN  DDDDDDDD    EEEEEEEEEE  XX      XX      ....
IIIIIII    NN      NN  DDDDDDDD    EEEEEEEEEE  XX      XX      ....

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 'INDEX -- DSR/DSRPLUS DSRINDEX/INDEX Utility'
;     2          0002  0 MODULE INDEX (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 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    11          0011  1 !*  ALL RIGHTS RESERVED.                                                      *
;    12          0012  1 !*                                                                           *
;    13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    18          0018  1 !*  TRANSFERRED.                                                              *
;    19          0019  1 !*                                                                           *
;    20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    22          0022  1 !*  CORPORATION.                                                              *
;    23          0023  1 !*                                                                           *
;    24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    26          0026  1 !*                                                                           *
;    27          0027  1 !*                                                                           *
;    28          0028  1 !*****************************************************************************
;    29          0029  1 !
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 ! FACILITY:
;    33          0033  1 !   DSR (Digital Standard RUNOFF) /DSRPLUS DSRINDEX/INDEX Utility
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !   This module contains the two major routines that make up the INDEX utility:
;    37          0037  1 !
;    38          0038  1 !       NDXINP - Reads input file and builds internal binary index
;    39          0039  1 !       MAKNDX - Writes formatted index from internal binary index
;    40          0040  1 !
;    41          0041  1 ! ENVIRONMENT:   Transportable
;    42          0042  1 !
;    43          0043  1 ! AUTHOR:        JPK
;    44          0044  1 !
;    45          0045  1 ! CREATION DATE: December 1981
;    46          0046  1 !
;    47          0047  1 ! MODIFIED BY:
;    48          0048  1 !
;    49          0049  1 !       009     JPK00023        20-May-1983
;    50          0050  1 !               Modified INDEX, NDXT20 and NDXVMS to check status of
;    51          0051  1 !               $XPO_PARSE_SPEC to avoid error messages from XPORT.
;    52          0052  1 !
;    53          0053  1 !       008     JPK00018        09-Mar-1983
;    54          0054  1 !               Modified INDEX to handle new BRN format.
;    55          0055  1 !               Modified NDXOUT to handle specifyable levels on SORT= string.
;    56          0056  1 !               Modified NDXFMT to output new RUNOFF prologue.
;    57          0057  1 !               Modified NDXPAG to output new TMS prologue and RUNOFF epilogue.
```

```
;     58          0058  1 !
;     59          0059  1 !       007     JPK00015        04-Feb-1983
;     60          0060  1 !               Cleaned up module names, modified revision history to
;     61          0061  1 !               conform with established standards. Updated copyright dates.
;     62          0062  1 !
;     63          0063  1 !       006     JPK00012        24-Jan-1983
;     64          0064  1 !               Modified NDXVMSMSG.MSG to define error messages for both
;     65          0065  1 !               DSRINDEX and INDEX.
;     66          0066  1 !               Added require of NDXVMSREQ.R32 to NDXOUT, NDXFMT, NDXDAT,
;     67          0067  1 !               INDEX, NDXMSG, NDXXTN, NDXTMS, NDXVMS and NDXPAG for BLISS32.
;     68          0068  1 !               Since this file defines the error message literals,
;     69          0069  1 !               the EXTERNAL REFERENCEs for the error message literals
;     70          0070  1 !               have been removed.
;     71          0071  1 !
;     72          0072  1 !       005     JPK00011        24-Jan-1983
;     73          0073  1 !               Changed CMDBLK [NDX$G_LEVEL] to CMDBLK [NDX$H_LEVEL]
;     74          0074  1 !               Changed CMDBLK [NDX$H_FORMAT] to CMDBLK [NDX$A_LAYOUT]
;     75          0075  1 !               Changed CMDBLK [NDX$V_TMS11] and CMDBLK [NDX$V_TEX] to CMDBLK [NDX$H_FORMAT]
;     76          0076  1 !               Changed comparisons of (.CHRSIZ EQLA CHRSZA) to
;     77          0077  1 !               (.CMDBLK [NDX$H_FORMAT] EQL TMS11_A).
;     78          0078  1 !               Definitions were changed in NDXCLI and references to the
;     79          0079  1 !               effected fields were changed in NDXPAG, NDXFMT, INDEX, NDXVMS
;     80          0080  1 !               and NDXCLIDMP.
;     81          0081  1 !
;     82          0082  1 !       004     JPK00009        24-Jan-1983
;     83          0083  1 !               Modified to enhance performance. The sort buckets have each
;     84          0084  1 !               been divided into 27 sub-buckets; 1 for each letter and 1
;     85          0085  1 !               for non-alphas. Removed reference to BUCKET from INDEX.
;     86          0086  1 !               Definition of the structure was added to NDXPOL. References
;     87          0087  1 !               to BUCKET were changed in modules NDXOUT, NDXINI, NDXFMT
;     88          0088  1 !               and NDXDAT.
;     89          0089  1 !
;     90          0090  1 !       003     JPK00007        24-Sep-1982
;     91          0091  1 !               Reset PAGENO if not concatenated input in INDEX.
;     92          0092  1 !
;     93          0093  1 !       002     JPK00002        16-Sep-1982
;     94          0094  1 !               Fix transaction mapping bug in INDEX. This bug occurred only
;     95          0095  1 !               when processing more than one input file where a file in the
;     96          0096  1 !               middle of the sequence contained no indexing information.
;     97          0097  1 !               If there had been only one transaction previous to the
;     98          0098  1 !               empty file, the resulting page numbers would be incorrect
;     99          0099  1 !               but the error would not be discovered by INDEX. If there
;    100          0100  1 !               were more than one transaction preceeding the empty file,
;    101          0101  1 !               transaction numbers would appear out of sequence which would
;    102          0102  1 !               be reported by INDEX as an internal logic error.
;    103          0103  1 !
;    104          0104  1 !--
;    105          0105  1
;    106          0106  1 !
;    107          0107  1 ! TABLE OF CONTENTS:
;    108          0108  1 !
;    109          0109  1
;    110          0110  1 FORWARD ROUTINE
;    111          0111  1     NDXINP : NOVALUE,                       ! Create internal binary index
;    112          0112  1     MAKNDX : NOVALUE;                       ! Write formatted index
;    113          0113  1
;    114          0114  1 !
```

```
;   115          0115  1 ! INCLUDE FILES:
;   116          0116  1 !
;   117          0117  1
;   118          0118  1 LIBRARY 'NXPORT:XPORT';
;   119          0119  1
;   120          0120  1 REQUIRE 'REQ:NDXCLI';                       ! Command line information block
;   121          0261  1
;   122          0262  1 REQUIRE 'REQ:BRNRTY';                       ! .BRN file record type literals
;   123          0342  1
;   124          0343  1 REQUIRE 'REQ:NDXRTY';                       ! Index binary record type literals
;   125          0405  1
;   126          0406  1 REQUIRE 'REQ:NDXXPL';                       ! Extended indexing attributes
;   127          0504  1
;   128          0505  1 REQUIRE 'REQ:NDXPOL';                       ! Pool data structures
;   129          0723  1
;   130          0724  1 REQUIRE 'REQ:PAGEN';                        ! Page reference data structure definitions
;   131          0805  1
;   132        L 0806  1 %IF %BLISS (BLISS32)                        ! For BLISS32
;   133          0807  1 %THEN
;   134          0808  1
;   135          0809  1 REQUIRE 'REQ:NDXVMSREQ';                          ! Error message numbers
;   136          1090  1
;   137          1091  1 %FI
;   138          1092  1
;   139          1093  1 !
;   140          1094  1 ! MACROS:
;   141          1095  1 !
;   142          1096  1 !
;   143          1097  1 ! EQUATED SYMBOLS:
;   144          1098  1 !
;   145          1099  1
;   146          1100  1 LITERAL
;   147          1101  1     TRUE = 1,
;   148          1102  1     FALSE = 0;
;   149          1103  1
;   150          1104  1 !
;   151          1105  1 ! OWN STORAGE:
;   152          1106  1 !
;   153          1107  1
;   154          1108  1 OWN
;   155          1109  1     XTN_OFFSET,                                ! Used to remap transaction numbers
;   156          1110  1     HIGHEST_XTN_OUT,                           ! Highest remapped transaction number
;   157          1111  1     INDEX_ENTRIES : INITIAL (0);               ! Number of binary index entries processes
;   158          1112  1
;   159          1113  1 OWN
;   160          1114  1     RNX_FILE : $STR_DESCRIPTOR (STRING = '.RNX'),
;   161          1115  1     TMS_FILE : $STR_DESCRIPTOR (STRING = '.TMS'),
;   162          1116  1     TEX_FILE : $STR_DESCRIPTOR (STRING = '.TEX');
;   163          1117  1
;   164          1118  1 !
;   165          1119  1 ! EXTERNAL REFERENCES:
;   166          1120  1 !
;   167          1121  1
;   168          1122  1 EXTERNAL LITERAL
;   169          1123  1     RINTES : UNSIGNED (8);                     ! RUNOFF escape sequence character
;   170          1124  1
;   171          1125  1 EXTERNAL
```

```
;  172          1126  1      CMDBLK : $NDXCMD,                       ! Command line information block
;  173          1127  1      XPLBLK : $XPL_BLOCK,                    ! Extended indexing information block
;  174          1128  1      OUTIOB : $XPO_IOB (),                   ! Output file IOB
;  175          1129  1      PAGEN  : PAGE_DEFINITION,
;  176          1130  1      PAGENO,                                 ! Page counter
;  177          1131  1      NDXPOL,                                 ! Address of indexing pool
;  178          1132  1      NDXSGE,                                 ! End of current segment.
;  179          1133  1      NDXSGF,
;  180          1134  1      XTNCNT,                                 ! Number of XTNTAB entries
;  181          1135  1      XTNLSP,
;  182          1136  1      XTNLSX,
;  183          1137  1      XTNPOL,
;  184          1138  1      XTNSGP,
;  185          1139  1      XTNTAB,                                 ! List of transaction numbers assigned
;  186          1140  1      XPAGEN,
;  187          1141  1      BOOKID;                                 ! Master index book ident string address
;  188          1142  1
;  189          1143  1  EXTERNAL ROUTINE
;  190          1144  1
;  191        L 1145  1  %IF %BLISS (BLISS32)
;  192          1146  1  %THEN
;  193          1147  1
;  194          1148  1      OPEN_ERROR,                             ! Handle file open errors for BLISS32
;  195          1149  1
;  196          1150  1  %FI
;  197          1151  1
;  198          1152  1      DMPENT : NOVALUE,                       ! Dump entry text
;  199          1153  1      ASGXTN : NOVALUE,                       ! Assign transaction # to page reference
;  200          1154  1      XINIT  : NOVALUE,                       ! Initialize pool
;  201          1155  1      FPOOL  : NOVALUE,                       ! Release pool space
;  202          1156  1      PERMUT : NOVALUE,                       ! Insert index entry in internal binary index
;  203          1157  1      NDXFMT : NOVALUE,                       ! Generate final output index
;  204          1158  1      SAVDAT;                                 ! Insert data in index pool
```

```
;  206     1159  1 %SBTTL 'GLOBAL ROUTINE NDXINP -- Process input file'
;  207     1160  1
;  208     1161  1 GLOBAL ROUTINE NDXINP : NOVALUE =
;  209     1162  1
;  210     1163  1 !++
;  211     1164  1 !
;  212     1165  1 ! FUNCTIONAL DESCRIPTION:
;  213     1166  1 !
;  214     1167  1 !       This routine generates an internal binary index from an input file.
;  215     1168  1 !
;  216     1169  1 !       If the input file is not concatenated to the previous input file,
;  217     1170  1 !       this routine calls MAKNDX to generate the output index from the
;  218     1171  1 !       existing internal binary index before processing the input file.
;  219     1172  1 !
;  220     1173  1 ! FORMAL PARAMETERS:
;  221     1174  1 !
;  222     1175  1 !       None.
;  223     1176  1 !
;  224     1177  1 ! IMPLICIT INPUTS:
;  225     1178  1 !
;  226     1179  1 !       CMDBLK  - Command line information block
;  227     1180  1 !
;  228     1181  1 ! IMPLICIT OUTPUTS:
;  229     1182  1 !
;  230     1183  1 !   CMDBLK [NDX$T_RELATED_FILE] -
;  231     1184  1 !       This string is set to the resultant input file specification for
;  232     1185  1 !       every input file which is not concatenated to the previous one.
;  233     1186  1 !       Thus the related file name for a series of concatenated input files
;  234     1187  1 !       is the resultant file name of the first file in the series.
;  235     1188  1 !
;  236     1189  1 !   XPLBLK -
;  237     1190  1 !       This global data structure is filled in with data read in from
;  238     1191  1 !       an INDEX_ATTRIBUTES record group.
;  239     1192  1 !
;  240     1193  1 !   INDEX_ENTRIES -
;  241     1194  1 !       is set to the number of binary index records processed.
;  242     1195  1 !
;  243     1196  1 !   XTN_OFFSET -
;  244     1197  1 !       is set to the highest transaction number processed in the current
;  245     1198  1 !       input stream.
;  246     1199  1 !
;  247     1200  1 !   BOOKID -
;  248     1201  1 !       is the address of the book identification string if generating
;  249     1202  1 !       a master index.
;  250     1203  1 !
;  251     1204  1 ! ROUTINE VALUE:
;  252     1205  1 ! COMPLETION CODES:
;  253     1206  1 !
;  254     1207  1 !       None
;  255     1208  1 !
;  256     1209  1 ! SIDE EFFECTS:
;  257     1210  1 !
;  258     1211  1 !       None
;  259     1212  1 !--
;  260     1213  1
;  261     1214  2     BEGIN
;  262     1215  2
```

```
;  263     1216  2     LOCAL
;  264     1217  2         PARSE_BLK : $XPO_SPEC_BLOCK,                    ! For parsing input file name
;  265     1218  2         INPUT_IOB : $XPO_IOB () VOLATILE,               ! Input file IOB
;  266     1219  2         FILE_IDENTIFIER,                                ! Will hold input file type code
;  267     1220  2         INDEX_VERSION,                                  ! Will hold index file format revision level
;  268     1221  2         INDEX_RECORDS,                                  ! Number of index entry records in input file
;  269     1222  2         HEADING_LENGTH,                                 ! Length of record group headings
;  270     1223  2         CONFUSE_COUNT,                                  ! Counter of "confused" messages
;  271     1224  2         CORRECT_FORMAT;                                 ! For checking file format
;  272     1225  2
;  273     1226  2     !
;  274     1227  2     ! Initialization
;  275     1228  2     !
;  276     1229  2     INDEX_RECORDS = 0;                                  ! No indexing information seen yet
;  277     1230  2     CONFUSE_COUNT = 0;                                  ! Initialize "confused" counter
;  278     1231  2
;  279     1232  2     !
;  280     1233  2     ! Initialize input IOB and parse input file specification
;  281     1234  2     !
;  282     1235  2     $XPO_IOB_INIT (IOB = INPUT_IOB);
;  283     1236  2
;  284   P 1237  2     IF NOT $XPO_PARSE_SPEC (SPEC_BLOCK = PARSE_BLK, FAILURE = 0,
;  285     1238  3                 FILE_SPEC = CMDBLK [NDX$T_INPUT_FILE])
;  286     1239  2     THEN
;  287     1240  2         !
;  288     1241  2         ! Input file name parse failed. Set file type length non-zero to force
;  289     1242  2         ! quick open failure message.
;  290     1243  2         !
;  291     1244  2         PARSE_BLK [XPO$H_FILE_TYPE] = 1;
;  292     1245  2
;  293     1246  2     IF .PARSE_BLK [XPO$H_FILE_TYPE] NEQ 0
;  294     1247  2     THEN
;  295     1248  2         !
;  296     1249  2         ! User specified a file type. Open file with no defaults.
;  297     1250  2         !
;  298   P 1251  2         $XPO_OPEN (IOB = INPUT_IOB, FILE_SPEC = CMDBLK [NDX$T_INPUT_FILE], ATTRIBUTES = BINARY
;  299   P 1252  2             %IF %BLISS (BLISS32) %THEN , FAILURE = OPEN_ERROR %FI
;  300     1253  3             )
;  301     1254  2     ELSE
;  302     1255  3         BEGIN
;  303     1256  3         !
;  304     1257  3         ! User did not specify a file type. Try .BRN first.
;  305     1258  3         !
;  306   P 1259  3         IF $XPO_OPEN (IOB = INPUT_IOB, FILE_SPEC = CMDBLK [NDX$T_INPUT_FILE],
;  307     1260  4             DEFAULT = '.BRN', ATTRIBUTES = BINARY, FAILURE = 0)
;  308     1261  3         NEQ XPO$_NORMAL
;  309     1262  3         THEN
;  310     1263  4             BEGIN
;  311     1264  4             !
;  312     1265  4             ! No .BRN file, try .BIX
;  313     1266  4             !
;  314     1267  4             $XPO_IOB_INIT (IOB = INPUT_IOB);
;  315   P 1268  4             $XPO_OPEN (IOB = INPUT_IOB, FILE_SPEC = CMDBLK [NDX$T_INPUT_FILE],
;  316   P 1269  4                 DEFAULT = '.BIX', ATTRIBUTES = BINARY
;  317   P 1270  4                 %IF %BLISS (BLISS32) %THEN , FAILURE = OPEN_ERROR %FI
;  318     1271  4                 );
;  319     1272  3             END;
```

```
;  320     1273  2          END;
;  321     1274
;  322     1275  2      IF .CMDBLK [NDX$V_LOG]
;  323     1276  2      THEN
;  324     1277  2
;  325   L 1278  2  %IF %BLISS (BLISS32)
;  326     1279  2  %THEN                                          ! Signal message for BLISS32
;  327     1280
;  328     1281  2          SIGNAL (INDEX$_PROCFILE, 1, INPUT_IOB [IOB$T_RESULTANT]);
;  329     1282
;  330   U 1283  2  %ELSE                                          ! Use $XPO_PUT_MSG otherwise
;  331   U 1284
;  332   U 1285  2          $XPO_PUT_MSG (SEVERITY = SUCCESS,
;  333   U 1286  2              STRING = $STR_CONCAT ('processing file ''', INPUT_IOB [IOB$T_RESULTANT], ''''));
;  334   U 1287
;  335     1288  2  %FI
;  336     1289
;  337     1290  2      IF NOT .CMDBLK [NDX$V_INPUT_CONCAT]
;  338     1291  2      THEN
;  339     1292  3          BEGIN
;  340     1293  3          !
;  341     1294  3          ! Input file is not concatenated to previous input file.
;  342     1295  3          !
;  343     1296  3          ! Call MAKNDX to generate the output file for the previous input file
;  344     1297  3          ! if there were any index entries. MAKNDX will also re-initialize
;  345     1298  3          ! the internal index and transaction pools.
;  346     1299  3          !
;  347     1300  3          MAKNDX ();
;  348     1301  3
;  349     1302  3          !
;  350     1303  3          ! Save resultant input file specification to be used as the related
;  351     1304  3          ! file name when the next output file is created by MAKNDX.
;  352     1305  3          !
;  353     1306  3          $STR_COPY (STRING = INPUT_IOB [IOB$T_RESULTANT], TARGET = CMDBLK [NDX$T_RELATED_FILE]);
;  354     1307  3          !
;  355     1308  3          ! Reset the page counter
;  356     1309  3          !
;  357     1310  3          PAGENO = 0;
;  358     1311  3          !
;  359     1312  3          ! Reset transaction mapping. There are no transactions in the
;  360     1313  3          ! current input stream yet.
;  361     1314  3          !
;  362     1315  3          XTN_OFFSET = 0;
;  363     1316  3          HIGHEST_XTN_OUT = 0;
;  364     1317  2          END;
;  365     1318  2
;  366     1319  2      !
;  367     1320  2      ! Initialize the internal index if necessary
;  368     1321  2      !
;  369     1322  2      IF .NDXPOL EQL 0 THEN XINIT ();
;  370     1323  2
;  371     1324  2      !
;  372     1325  2      ! Setup master index book identifier
;  373     1326  2      !
;  374     1327  3      BEGIN
;  375     1328  3
;  376     1329  3      BIND
```

```
;  377          1330  3          DSC = CMDBLK [NDX$T_MASTER_BOOK] : $STR_DESCRIPTOR ();
;  378          1331  3
;  379          1332  3      BOOKID = SAVDAT (.DSC [STR$A_POINTER], DS_X_STRING, .DSC [STR$H_LENGTH]);
;  380          1333  2      END;
;  381          1334  2
;  382          1335  2      !
;  383          1336  2      ! Read file indentification record.
;  384          1337  2      !
;  385          1338  2      IF $XPO_GET (IOB = INPUT_IOB, FULLWORDS = 1) NEQ XPO$_NORMAL
;  386          1339  2      THEN
;  387          1340  3          BEGIN
;  388          1341  3          !
;  389          1342  3          ! Empty input file.
;  390          1343  3          !
;  391          1344  3
;  392        L 1345  3  %IF %BLISS (BLISS32)
;  393          1346  3  %THEN                                          ! Signal error for BLISS32
;  394          1347  3
;  395          1348  3          SIGNAL (INDEX$_EMPTYIN, 1, INPUT_IOB [IOB$T_RESULTANT]);
;  396          1349  3
;  397        U 1350  3  %ELSE                                          ! Use $XPO_PUT_MSG otherwise
;  398        U 1351  3
;  399        U 1352  3          $XPO_PUT_MSG (SEVERITY = WARNING,
;  400        U 1353  3              STRING = $STR_CONCAT ('empty input file ''', INPUT_IOB [IOB$T_RESULTANT], ''''));
;  401        U 1354  3
;  402          1355  3  %FI
;  403          1356  3
;  404          1357  3          $XPO_CLOSE (IOB = INPUT_IOB);
;  405          1358  3          RETURN;
;  406          1359  2          END;
;  407          1360  2
;  408          1361  2      FILE_IDENTIFIER = .(.INPUT_IOB [IOB$A_DATA])<0, %BPVAL/2, 1>;
;  409          1362  2      INDEX_VERSION = .(.INPUT_IOB [IOB$A_DATA])<%BPVAL/2, %BPVAL/2, 1>;
;  410          1363  2      CORRECT_FORMAT = TRUE;                      ! Assume correct file format
;  411          1364  2
;  412          1365  2      SELECTONE TRUE OF
;  413          1366  2          SET
;  414          1367  2
;  415          1368  2          [.FILE_IDENTIFIER EQL BRN_FILE] :
;  416          1369  3              BEGIN
;  417          1370  3              !
;  418          1371  3              ! Binary RuNoff format file (.BRN)
;  419          1372  3              !
;  420          1373  3              HEADING_LENGTH = 2;                 ! Length of record group headings
;  421          1374  3
;  422          1375  3              IF $XPO_GET (IOB = INPUT_IOB, FULLWORDS = 1) EQL XPO$_NORMAL
;  423          1376  3              THEN
;  424          1377  4                  BEGIN
;  425          1378  4                  !
;  426          1379  4                  ! Validate .BRN format information
;  427          1380  4                  !
;  428          1381  4                  IF ..INPUT_IOB [IOB$A_DATA] NEQ BRN_IDENT THEN CORRECT_FORMAT = FALSE;
;  429          1382  4
;  430          1383  4                  END
;  431          1384  3              ELSE
;  432          1385  3                  CORRECT_FORMAT = FALSE;         ! Bad input file format - no ident info
;  433          1386  3
```

```
;  434          1387  2              END;
;  435          1388  2
;  436          1389  2          [(.FILE_IDENTIFIER EQL NEW_SEQUENCE) AND (.INDEX_VERSION EQL INDEX_FORMAT)] :
;  437          1390  2              !
;  438          1391  2              ! Input file is .BIX format. Set record group heading length.
;  439          1392  2              !
;  440          1393  2              HEADING_LENGTH = 1;
;  441          1394  2
;  442          1395  2          [OTHERWISE] :
;  443          1396  2              !
;  444          1397  2              ! Invalid input file format
;  445          1398  2              !
;  446          1399  2              CORRECT_FORMAT = FALSE;
;  447          1400  2          TES;
;  448          1401  2
;  449          1402  2      IF NOT .CORRECT_FORMAT
;  450          1403  2      THEN
;  451          1404  3          BEGIN
;  452          1405  3          !
;  453          1406  3          ! Report invalid input file format and quit.
;  454          1407  3          !
;  455          1408  3
;  456        L 1409  3  %IF %BLISS (BLISS32)
;  457          1410  3  %THEN                                          ! Signal error for BLISS32
;  458          1411  3
;  459          1412  3          SIGNAL (INDEX$_INVINPUT, 1, INPUT_IOB [IOB$T_RESULTANT]);
;  460          1413  3
;  461        U 1414  3  %ELSE                                          ! Use $XPO_PUT_MSG otherwise
;  462        U 1415  3
;  463        U 1416  3          $XPO_PUT_MSG (SEVERITY = WARNING,
;  464        U 1417  3              STRING = $STR_CONCAT ('invalid input file format ''', INPUT_IOB [IOB$T_RESULTANT], ''''));
;  465        U 1418  3
;  466          1419  3  %FI
;  467          1420  3
;  468          1421  3          $XPO_CLOSE (IOB = INPUT_IOB);
;  469          1422  3          RETURN;
;  470          1423  2          END;
;  471          1424  2
;  472          1425  2      !
;  473          1426  2      ! Process the input file.
;  474          1427  2      !
;  475          1428  2      WHILE $XPO_GET (IOB = INPUT_IOB, FULLWORDS = .HEADING_LENGTH) EQL XPO$_NORMAL DO
;  476          1429  3          BEGIN
;  477          1430  3
;  478          1431  3          BIND
;  479          1432  3              GROUP_HEAD = .INPUT_IOB [IOB$A_DATA] : VECTOR;
;  480          1433  3
;  481          1434  4          IF (.FILE_IDENTIFIER EQL BRN_FILE) AND (.GROUP_HEAD [0] NEQ BRN_INDEX)
;  482          1435  3          THEN
;  483          1436  4              BEGIN
;  484          1437  4              !
;  485          1438  4              ! File is .BRN format but record group is not an index
;  486          1439  4              ! record group -- skip rest of record group.
;  487          1440  4              !
;  488          1441  4              $XPO_GET (IOB = INPUT_IOB, FULLWORDS = .GROUP_HEAD [1]);
;  489          1442  4              END
;  490          1443  3          ELSE
```

```
;   491          1444  4                BEGIN
;   492          1445  4
;   493          1446  4                LOCAL
;   494          1447  4                    RECORD_TYPE;
;   495          1448  4
;   496          1449  4                !
;   497          1450  4                ! If the input file is a .BRN file, read the index record
;   498          1451  4                ! group identifier.
;   499          1452  4                !
;   500          1453  4                IF .FILE_IDENTIFIER EQL BRN_FILE THEN $XPO_GET (IOB = INPUT_IOB, FULLWORDS = 1);
;   501          1454  4
;   502          1455  4                RECORD_TYPE = .(.INPUT_IOB [IOB$A_DATA])<0, %BPVAL/2>;
;   503          1456  4
;   504          1457  4                SELECTONE TRUE OF
;   505          1458  4                    SET
;   506          1459  4
;   507          1460  4                    [.RECORD_TYPE EQL NEW_SEQUENCE] :
;   508          1461  5                        BEGIN
;   509          1462  5                        INDEX_VERSION = .(.INPUT_IOB [IOB$A_DATA])<%BPVAL/2, %BPVAL/2, 1>;
;   510          1463  5
;   511          1464  6                        IF (.INDEX_VERSION NEQ INDEX_FORMAT)
;   512          1465  6                        OR (.FILE_IDENTIFIER NEQ BRN_FILE)
;   513          1466  5                        THEN
;   514          1467  6                            BEGIN
;   515          1468  6                            !
;   516          1469  6                            ! Report invalid input file format and quit.
;   517          1470  6                            !
;   518          1471  6
;   519        L 1472  6 %IF %BLISS (BLISS32)
;   520          1473  6 %THEN                                       ! Signal error for BLISS32
;   521          1474  6
;   522          1475  6                            SIGNAL (INDEX$_INVINPUT, 1, INPUT_IOB [IOB$T_RESULTANT]);
;   523          1476  6
;   524        U 1477  6 %ELSE                                       ! Use $XPO_PUT_MSG otherwise
;   525        U 1478  6
;   526        U 1479  6                            $XPO_PUT_MSG (SEVERITY = WARNING,
;   527        U 1480  6                                STRING = $STR_CONCAT ('invalid input file format ''',
;   528        U 1481  6                                    INPUT_IOB [IOB$T_RESULTANT], ''''));
;   529        U 1482  6
;   530          1483  6 %FI
;   531          1484  6
;   532          1485  6                            $XPO_CLOSE (IOB = INPUT_IOB);
;   533          1486  6                            RETURN;
;   534          1487  5                            END;
;   535          1488  5
;   536          1489  4                        END;
;   537          1490  4
;   538          1491  4                    [.RECORD_TYPE EQL INDEX_XTN] :
;   539          1492  5                        BEGIN
;   540          1493  5                        !
;   541          1494  5                        ! Index page reference transaction
;   542          1495  5                        !
;   543          1496  5                        CONFUSE_COUNT = 0;
;   544          1497  5                        $XPO_GET (IOB = INPUT_IOB, FULLWORDS = PAGE_SCT_SIZE + 1);
;   545          1498  5
;   546          1499  6                        BEGIN
;   547          1500  6                        BIND
```

```
;  548          1501  6                      XTN_AND_PAGE = .INPUT_IOB [IOB$A_DATA] : VECTOR,
;  549          1502  6                      PAGE_REF = XTN_AND_PAGE [1] : PAGE_DEFINITION;
;  550          1503  6
;  551          1504  6                  IF .XTN_AND_PAGE [0] NEQ 0
;  552          1505  6                  THEN
;  553          1506  7                      BEGIN
;  554          1507  7                      !
;  555          1508  7                      ! Process only non-zero transaction numbers
;  556          1509  7                      !
;  557          1510  7                      HIGHEST_XTN_OUT = .XTN_AND_PAGE [0] + .XTN_OFFSET;      ! Remap transaction number
;  558          1511  7                      ASGXTN (PAGE_REF, .HIGHEST_XTN_OUT);                    ! Store xtn in internal pool
;  559          1512  6                      END;
;  560          1513  6
;  561          1514  5                  END;
;  562          1515  5
;  563          1516  4                  END;
;  564          1517  4
;  565          1518  4              [.RECORD_TYPE EQL INDEX_ENTRY] :
;  566          1519  5                  BEGIN
;  567          1520  5                  !
;  568          1521  5                  ! Index entry record group
;  569          1522  5                  !
;  570          1523  5                  LOCAL
;  571          1524  5                      DESCRIPTOR : VECTOR [3];
;  572          1525  5
;  573          1526  5                  CONFUSE_COUNT = 0;
;  574          1527  5                  $XPO_GET (IOB = INPUT_IOB, FULLWORDS = 3);  ! Get descriptor
;  575          1528  5
;  576          1529  6                  BEGIN
;  577          1530  6                  BIND
;  578          1531  6                      X = .INPUT_IOB [IOB$A_DATA] : VECTOR;
;  579          1532  6
;  580          1533  6                  INCR I FROM 1 TO 3 DO
;  581          1534  6                      DESCRIPTOR [.I - 1] = .X [.I - 1];      ! Copy descriptor
;  582          1535  6
;  583          1536  5                  END;
;  584          1537  5
;  585        P 1538  5                  $XPO_GET (IOB = INPUT_IOB,                              ! Get entry text
;  586          1539  5                      FULLWORDS = CH$ALLOCATION (.DESCRIPTOR [0]));
;  587          1540  5
;  588          1541  5                  IF .DESCRIPTOR [0] GTR 1200
;  589          1542  5                  THEN
;  590          1543  6                      BEGIN
;  591          1544  6                      !
;  592          1545  6                      ! Entry text is too long.
;  593          1546  6                      ! Tell the user and truncate it.
;  594          1547  6                      !
;  595        L 1548  6 %IF %BLISS (BLISS32)
;  596          1549  6 %THEN                                                  ! Signal errors for BLISS32
;  597          1550  6
;  598          1551  6                      SIGNAL (INDEX$_TRUNCATED);
;  599          1552  6
;  600        U 1553  6 %ELSE                                                  ! Use $XPO_PUT_MSG otherwise
;  601        U 1554  6
;  602        U 1555  6                      $XPO_PUT_MSG (SEVERITY = WARNING, STRING = 'string too long - truncated');
;  603        U 1556  6
;  604          1557  6 %FI
```

```
;  605          1558  6                    DESCRIPTOR [0] = 1200;
;  606          1559  6                    DMPENT (.DESCRIPTOR [0], CH$PTR (.INPUT_IOB [IOB$A_DATA]));
;  607          1560  6                    END;
;  608          1561  5
;  609          1562  5                !
;  610          1563  5                ! Remap transaction number for .INDEX and .XPLUS
;  611          1564  5                ! commands, i.e., those with non-zero transaction numbers.
;  612          1565  5                !
;  613          1566  5                IF .DESCRIPTOR [1] NEQ 0 THEN DESCRIPTOR [1] = .DESCRIPTOR [1] + .XTN_OFFSET;
;  614          1567  5
;  615          1568  5
;  616          1569  5                IF .CMDBLK [NDX$V_MASTER] NEQ 0
;  617          1570  5                THEN
;  618          1571  5                    BEGIN
;  619          1572  6                    !
;  620          1573  6                    ! Generating a master index
;  621          1574  6                    !
;  622          1575  6                    IF .CMDBLK [NDX$V_OVERRIDE]
;  623          1576  6                    OR (.XPLBLK [XPL$V_VALID] AND .XPLBLK [XPL$V_MASTER])
;  624          1577  7                    THEN
;  625          1578  6                        BEGIN
;  626          1579  7                        !
;  627          1580  7                        ! Override master indexing information or
;  628          1581  7                        ! this entry is a master entry.
;  629          1582  7                        !
;  630          1583  7                        ! Insert entry into pool, permuting if necessary.
;  631          1584  7                        ! Increment index entry counter.
;  632          1585  7                        !
;  633          1586  7                        PERMUT (.DESCRIPTOR [0], CH$PTR (.INPUT_IOB [IOB$A_DATA]),
;  634          1587  7                            .DESCRIPTOR [1], .DESCRIPTOR [2]);
;  635          1588  7
;  636          1589  7                        INDEX_RECORDS = .INDEX_RECORDS + 1;
;  637          1590  7                        END;
;  638          1591  6                    END
;  639          1592  6                ELSE
;  640          1593  5                    !
;  641          1594  5                    ! Not a master index
;  642          1595  5                    !
;  643          1596  5                    IF .CMDBLK [NDX$V_OVERRIDE]
;  644          1597  5                    OR (NOT (.XPLBLK [XPL$V_VALID] AND .XPLBLK [XPL$V_MASTER]))
;  645          1598  6                    THEN
;  646          1599  5                        BEGIN
;  647          1600  6                        !
;  648          1601  6                        ! Override master indexing information or
;  649          1602  6                        ! entry is not a master index entry.
;  650          1603  6                        !
;  651          1604  6                        ! Insert it into pool, permuting if necessary.
;  652          1605  6                        ! Increment index entry record counter.
;  653          1606  6                        !
;  654          1607  6                        PERMUT (.DESCRIPTOR [0], CH$PTR (.INPUT_IOB [IOB$A_DATA]),
;  655          1608  6                            .DESCRIPTOR [1], .DESCRIPTOR [2]);
;  656          1609  6
;  657          1610  6                        INDEX_RECORDS = .INDEX_RECORDS + 1;
;  658          1611  6                        END;
;  659          1612  5
;  660          1613  5
;  661          1614  5                !
```

```
;  662          1615  5                    ! Mark attributes block invalid.
;  663          1616  5                    !
;  664          1617  5                    XPLBLK [XPL$V_VALID] = FALSE;
;  665          1618  4                    END;
;  666          1619  4
;  667          1620  4                [(.RECORD_TYPE EQL INDEX_ATTRIBUTES) AND (.FILE_IDENTIFIER EQL BRN_FILE)] :
;  668          1621  5                    BEGIN
;  669          1622  5                    !
;  670          1623  5                    ! Extended indexing attributes record group
;  671          1624  5                    !
;  672          1625  5                    LOCAL
;  673          1626  5                        STR_LEN;
;  674          1627  5
;  675          1628  5                    CONFUSE_COUNT = 0;
;  676          1629  5
;  677          1630  5                    !
;  678          1631  5                    ! Get options flags.
;  679          1632  5                    !
;  680          1633  5                    $XPO_GET (IOB = INPUT_IOB, FULLWORDS = 1);
;  681          1634  5                    XPLBLK [XPL$V_OPTIONS] = .(.INPUT_IOB [IOB$A_DATA]);
;  682          1635  5
;  683          1636  5                    IF .CMDBLK [NDX$V_PAGE_MERGE]
;  684          1637  5                    THEN
;  685          1638  6                        BEGIN
;  686          1639  6                        !
;  687          1640  6                        ! Merging adjacent page references (ala TCX)
;  688          1641  6                        ! ignore BEGIN-END attributes
;  689          1642  6                        !
;  690          1643  6                        XPLBLK [XPL$V_BEGIN] = FALSE;
;  691          1644  6                        XPLBLK [XPL$V_END] = FALSE;
;  692          1645  5                        END;
;  693          1646  5
;  694          1647  5                    !
;  695          1648  5                    ! Get SORT string if any.
;  696          1649  5                    !
;  697          1650  5                    $XPO_GET (IOB = INPUT_IOB, FULLWORDS = 1);
;  698          1651  5                    STR_LEN = .(.INPUT_IOB [IOB$A_DATA]);
;  699          1652  5
;  700          1653  5                    IF .STR_LEN NEQ 0 THEN $XPO_GET (IOB = INPUT_IOB, FULLWORDS = CH$ALLOCATION (.STR_LEN));
;  701          1654  5
;  702        P 1655  5                    $STR_COPY (STRING = (.STR_LEN, CH$PTR (.INPUT_IOB [IOB$A_DATA])),
;  703          1656  5                        TARGET = XPLBLK [XPL$T_SORT]);
;  704          1657  5
;  705          1658  5                    !
;  706          1659  5                    ! Get APPEND string if any.
;  707          1660  5                    !
;  708          1661  5                    $XPO_GET (IOB = INPUT_IOB, FULLWORDS = 1);
;  709          1662  5                    STR_LEN = .(.INPUT_IOB [IOB$A_DATA]);
;  710          1663  5
;  711          1664  5                    IF .STR_LEN NEQ 0 THEN $XPO_GET (IOB = INPUT_IOB, FULLWORDS = CH$ALLOCATION (.STR_LEN));
;  712          1665  5
;  713        P 1666  5                    $STR_COPY (STRING = (.STR_LEN, CH$PTR (.INPUT_IOB [IOB$A_DATA])),
;  714          1667  5                        TARGET = XPLBLK [XPL$T_APPEND]);
;  715          1668  5
;  716          1669  5                    !
;  717          1670  5                    ! Attributes block now has valid information
;  718          1671  5                    !
```

```
;  719          1672  5                    XPLBLK [XPL$V_VALID] = TRUE;
;  720          1673  4                    END;
;  721          1674  4
;  722          1675  4                [OTHERWISE] :
;  723          1676  5                    BEGIN
;  724          1677  5                    !
;  725          1678  5                    ! Invalid record group
;  726          1679  5                    !
;  727          1680  5                    CONFUSE_COUNT = .CONFUSE_COUNT + 1;
;  728          1681  5
;  729          1682  5                    IF .CONFUSE_COUNT LEQ 5
;  730          1683  5                    THEN
;  731          1684  5
;  732        L 1685  5 %IF %BLISS (BLISS32)
;  733          1686  5 %THEN                                       ! Signal error for BLISS32
;  734          1687  5
;  735          1688  5                        SIGNAL (INDEX$_INVRECORD, 1, INPUT_IOB [IOB$T_RESULTANT]);
;  736          1689  5
;  737        U 1690  5 %ELSE                                       ! Use $XPO_PUT_MSG otherwise
;  738        U 1691  5
;  739        U 1692  5                        $XPO_PUT_MSG (SEVERITY = WARNING,
;  740        U 1693  5                            STRING = $STR_CONCAT ('invalid record group type in file ''', INPUT_IOB [IOB$T_R
;  741        U 1694  5
;  742          1695  5 %FI
;  743          1696  5
;  744          1697  4                    END;
;  745          1698  4                TES;
;  746          1699  4
;  747          1700  3                END;
;  748          1701  3
;  749          1702  2                END;
;  750          1703  2
;  751          1704  2            IF .INDEX_RECORDS EQL 0
;  752          1705  2            THEN
;  753          1706  2
;  754        L 1707  2 %IF %BLISS (BLISS32)
;  755          1708  2 %THEN                                       ! Signal error for BLISS32
;  756          1709  2
;  757          1710  2                SIGNAL (INDEX$_NOINDEX, 1, INPUT_IOB [IOB$T_RESULTANT]);
;  758          1711  2
;  759        U 1712  2 %ELSE                                       ! Use $XPO_PUT_MSG otherwise
;  760        U 1713  2
;  761        U 1714  2                $XPO_PUT_MSG (SEVERITY = WARNING,
;  762        U 1715  2                    STRING = $STR_CONCAT ('no index information in file ''', INPUT_IOB [IOB$T_RESULTANT], ''''));
;  763        U 1716  2
;  764          1717  2 %FI
;  765          1718  2
;  766          1719  2            IF .CMDBLK [NDX$V_LOG]
;  767          1720  2            THEN
;  768          1721  2
;  769        L 1722  2 %IF %BLISS (BLISS32)
;  770          1723  2 %THEN                                       ! Signal error for BLISS32
;  771          1724  2
;  772          1725  2                SIGNAL (INDEX$_COMPLETE, 1, INPUT_IOB [IOB$T_RESULTANT]);
;  773          1726  2
;  774        U 1727  2 %ELSE                                       ! Use $XPO_PUT_MSG otherwise
;  775        U 1728  2
```

```
;  776          U 1729  2          $XPO_PUT_MSG (SEVERITY = SUCCESS,
;  777          U 1730  2              STRING = $STR_CONCAT ('processing complete ''', INPUT_IOB [IOB$T_RESULTANT], ''''));
;  778          U 1731  2
;  779            1732  2 %FI
;  780            1733  2
;  781            1734  2      INDEX_ENTRIES = .INDEX_ENTRIES + .INDEX_RECORDS;     ! Accumulate number of index entries
;  782            1735  2      XTN_OFFSET = .HIGHEST_XTN_OUT;                       ! Remember last transaction number
;  783            1736  2      $XPO_CLOSE (IOB = INPUT_IOB);                        ! Close the input file
;  784            1737  1      END;


                                                        .TITLE  INDEX INDEX -- DSR/DSRPLUS DSRINDEX/INDEX Utili
;                                                                     ty
                                                        .IDENT  \V04-000\

                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                        58  4E  52  2E  00000 P.AAA:  .ASCII  \.RNX\                                ;
                                        53  4D  54  2E  00004 P.AAB:  .ASCII  \.TMS\                                ;
                                        58  45  54  2E  00008 P.AAC:  .ASCII  \.TEX\                                ;
                                        4E  52  42  2E  0000C P.AAD:  .ASCII  \.BRN\                                ;
                                        58  49  42  2E  00010 P.AAE:  .ASCII  \.BIX\                                ;

                                                        .PSECT  $OWN$,NOEXE,2

                                                00000 XTN_OFFSET:
                                                        .BLKB   4
                                                00004 HIGHEST_XTN_OUT:
                                                        .BLKB   4
                                        00000000  00008 INDEX_ENTRIES:
                                                        .LONG   0                                                   ;
                                            0004  0000C RNX_FILE:
                                                        .WORD   4                                                   ;
                                          01  0E  0000E         .BYTE   14, 1                                       ;
                                        00000000' 00010         .ADDRESS P.AAA                                      ;
                                            0004  00014 TMS_FILE:
                                                        .WORD   4                                                   ;
                                          01  0E  00016         .BYTE   14, 1                                       ;
                                        00000000' 00018         .ADDRESS P.AAB                                      ;
                                            0004  0001C TEX_FILE:
                                                        .WORD   4                                                   ;
                                          01  0E  0001E         .BYTE   14, 1                                       ;
                                        00000000' 00020         .ADDRESS P.AAC                                      ;
                                            0004  00024 $IOB$DEFAULT:
                                                        .WORD   4                                                   ;
                                          01  0E  00026         .BYTE   14, 1                                       ;
                                        00000000' 00028         .ADDRESS P.AAD                                      ;
                                            0004  0002C $IOB$DEFAULT:
                                                        .WORD   4                                                   ;
                                          01  0E  0002E         .BYTE   14, 1                                       ;
                                        00000000' 00030         .ADDRESS P.AAE                                      ;

                                                        .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$_TMS11
                                                .EXTRN  RINTES, CMDBLK, XPLBLK
                                                .EXTRN  OUTIOB, PAGEN, PAGENO
                                                .EXTRN  NDXPOL, NDXSGE, NDXSGF
                                                .EXTRN  XTNCNT, XTNLSP, XTNLSX
                                                .EXTRN  XTNPOL, XTNSGP, XTNTAB
                                                .EXTRN  XPAGEN, BOOKID, OPEN_ERROR
                                                .EXTRN  DMPENT, ASGXTN, XINIT
                                                .EXTRN  FPOOL, PERMUT, NDXFMT
                                                .EXTRN  SAVDAT, XPO$PARSE_SPEC
                                                .EXTRN  XPO$OPEN, XST$COPY
                                                .EXTRN  STR$FAILURE, XPO$GET
                                                .EXTRN  XPO$FAILURE, XPO$CLOSE

                                                .PSECT  $CODE$,NOWRT,2

                                    0FFC 00000  .ENTRY  NDXINP, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-  ; 1161
                                                        R11                                         ;
                5B 00000000G   EF   9E 00002    MOVAB   XPO$GET, R11                                ;
                5A 00000000G   EF   9E 00009    MOVAB   XPO$FAILURE, R10                            ;
                5E      FEB8   CE   9E 00010    MOVAB   -328(SP), SP                                ;
                               57   7C 00015    CLRQ    CONFUSE_COUNT                               ; 1230
00F4    8F      00             6E   00   2C 00017  MOVC5   #0, (SP), #0, #244, IOB$                ; 1235
                                        0C   AE      0001E                                          ;
                            0C  AE 0301003D   8F   D0 00020  MOVL    #50397245, IOB$                ;
                            2A  AE      020E  8F   B0 00028  MOVW    #526, IOB$RESULTANT+2          ;
```

```
                                                    7E  7C 0002E          CLRQ    -(SP)                                          ; 1238
                                                7E      01  CE 00030      MNEGL   #1, -(SP)                                      ;
                                                    B8  AD  9F 00033      PUSHAB  PARSE_BLK                                      ;
                                            00000000G   EF  9F 00036      PUSHAB  $STR$FILE_SPEC                                 ;
                                    00000000G   EF      05  FB 0003C      CALLS   #5, XPO$PARSE_SPEC                             ;
                                                04      50  E8 00043      BLBS    R0, 1$                                         ;
                                            E0  AD          01  B0 00046  MOVW    #1, PARSE_BLK+40                               ; 1244
                                                50  00000000G  EF  9E 0004A 1$:   MOVAB   $IOB$FILE_SPEC, R0                    ; 1253
                                            10  AE          50  D0 00051      MOVL    R0, IOB$+4                                     ;
                                                    E0  AD  B5 00055      TSTW    PARSE_BLK+40                                   ; 1246
                                                        4C  12 00058      BNEQ    2$                                             ;
                                            14  AE  00000000'  EF  9E 0005A  MOVAB   $IOB$DEFAULT, IOB$+8                       ; 1260
                                            3C  AE          01  88 00062      BISB2   #1, IOB$+48                                    ;
                                            38  AE          01  90 00066      MOVB    #1, IOB$+44                                    ;
                                                        7E  7C 0006A      CLRQ    -(SP)                                          ;
                                                    14  AE  9F 0006C      PUSHAB  IOB$                                           ;
                                    00000000G   EF          03  FB 0006F  CALLS   #3, XPO$OPEN                                    ;
                                    00208001    8F          50  D1 00076  CMPL    R0, #2129921                                   ; 1261
                                                            41  13 0007D  BEQL    3$                                             ;
    00F4    8F              00          6E                  00  2C 0007F  MOVC5   #0, (SP), #0, #244, IOB$                       ; 1267
                                                    0C  AE          00086                                                          ;
                                            0C  AE  0301003D    8F  D0 00088  MOVL    #50397245, IOB$                          ;
                                            2A  AE      020E    8F  B0 00090  MOVW    #526, IOB$RESULTANT+2                    ;
                                            10  AE  00000000G   EF  9E 00096  MOVAB   $IOB$FILE_SPEC, IOB$+4                   ; 1271
                                            14  AE  00000000'   EF  9E 0009E  MOVAB   $IOB$DEFAULT, IOB$+8                     ;
                                            3C  AE          01  88 000A6 2$:   BISB2   #1, IOB$+48                                ;
                                            38  AE          01  90 000AA      MOVB    #1, IOB$+44                                ;
                                                00000000G   EF  9F 000AE      PUSHAB  OPEN_ERROR                                 ;
                                                        7E  D4 000B4      CLRL    -(SP)                                      ;
                                                    14  AE  9F 000B6      PUSHAB  IOB$                                       ;
                                    00000000G   EF          03  FB 000B9  CALLS   #3, XPO$OPEN                                ;
                                12  00000000G   EF          01  E1 000C0 3$:   BBC     #1, CMDBLK+1, 4$                           ; 1275
                                                    28  AE  9F 000C8      PUSHAB  INPUT_IOB+28                               ; 1281
                                                        01  DD 000CB      PUSHL   #1                                         ;
                                                00000000G   8F  DD 000CD      PUSHL   #DSRINDEX$_PROCFILE                        ;
                                    00000000G   00          03  FB 000D3  CALLS   #3, LIB$SIGNAL                              ;
                                                2D  00000000G   EF  E8 000DA 4$:   BLBS    CMDBLK, 5$                             ; 1290
                                    00000000V   EF          00  FB 000E1  CALLS   #0, MAKNDX                                   ; 1300
                                                00000000G   EF  9F 000E8      PUSHAB  STR$FAILURE                                ; 1306
                                                        7E  D4 000EE      CLRL    -(SP)                                      ;
                                                00000000G   EF  9F 000F0      PUSHAB  $STR$TARGET                                ;
                                                    34  AE  9F 000F6      PUSHAB  $STR$STRING                                ;
                                                        7E  D4 000F9      CLRL    -(SP)                                      ;
                                    00000000G   EF          05  FB 000FB  CALLS   #5, XST$COPY                                ;
                                                00000000G   EF  D4 00102      CLRL    PAGENO                                     ; 1310
                                                00000000'   EF  7C 00108      CLRQ    XTN_OFFSET                                 ; 1315
                                                00000000G   EF  D5 0010E 5$:   TSTL    NDXPOL                                     ; 1322
                                                            07  12 00114      BNEQ    6$                                         ;
                                    00000000G   EF          00  FB 00116  CALLS   #0, XINIT                                   ;
                                                7E  00000000G   EF  3C 0011D 6$:   MOVZWL  DSC, -(SP)                           ; 1332
                                                        7E  D4 00124      CLRL    -(SP)                                      ;
                                                00000000G   EF  DD 00126      PUSHL   DSC+4                                      ;
                                    00000000G   EF          03  FB 0012C  CALLS   #3, SAVDAT                                  ;
                                    00000000G   EF          50  D0 00133  MOVL    R0, BOOKID                                 ;
                                            40  AE          04  B0 0013A  MOVW    #4, IOB$+52                                ; 1338
                                            42  AE          02  90 0013E  MOVB    #2, IOB$+54                                ;
                                            38  AE          06  90 00142  MOVB    #6, IOB$+44                                ;
```

```
                                 5A   DD 00146          PUSHL   R10                                                    ;
                                 7E   D4 00148          CLRL    -(SP)                                                  ;
                              14 AE   9F 0014A          PUSHAB  IOB$                                                   ;
                           6B         03 FB 0014D       CALLS   #3, XPO$GET                                            ;
                00208001   8F         50 D1 00150       CMPL    R0, #2129921                                           ;
                                      0E 13 00157       BEQL    7$                                                     ;
                              28 AE   9F 00159          PUSHAB  INPUT_IOB+28                                           ; 1348
                                      01 DD 0015C       PUSHL   #1                                                     ;
                        00000000G     8F DD 0015E       PUSHL   #DSRINDEX$_EMPTYIN                                     ;
                                00FA  31 00164          BRW     19$                                                    ;
                           56   44   BE 32 00167 7$:     CVTWL   @INPUT_IOB+56, FILE_IDENTIFIER                         ; 1361
             55       44   BE   F0   8F 78 0016B        ASHL    #-16, @INPUT_IOB+56, INDEX_VERSION                     ; 1362
                           53         01 D0 00171       MOVL    #1, CORRECT_FORMAT                                     ; 1363
                                      54 D4 00174       CLRL    R4                                                     ; 1368
                FFFFFFFF   8F         56 D1 00176       CMPL    FILE_IDENTIFIER, #-1                                   ;
                                      2C 12 0017D       BNEQ    8$                                                     ;
                                      54 D6 0017F       INCL    R4                                                     ;
                           52         02 D0 00181       MOVL    #2, HEADING_LENGTH                                     ; 1373
                      40   AE         04 B0 00184       MOVW    #4, IOB$+52                                            ; 1375
                      42   AE         02 90 00188       MOVB    #2, IOB$+54                                            ;
                      38   AE         06 90 0018C       MOVB    #6, IOB$+44                                            ;
                                      5A DD 00190       PUSHL   R10                                                    ;
                                      7E D4 00192       CLRL    -(SP)                                                  ;
                              14 AE   9F 00194          PUSHAB  IOB$                                                   ;
                           6B         03 FB 00197       CALLS   #3, XPO$GET                                            ;
                00208001   8F         50 D1 0019A       CMPL    R0, #2129921                                           ;
                                      2A 12 001A1       BNEQ    11$                                                    ;
                           02   44   BE D1 001A3        CMPL    @INPUT_IOB+56, #2                                      ; 1381
                                      26 13 001A7       BEQL    12$                                                    ;
                                      22 11 001A9       BRB     11$                                                    ; 1385
                                      51 D4 001AB 8$:   CLRL    R1                                                     ; 1389
                           01         56 D1 001AD       CMPL    FILE_IDENTIFIER, #1                                    ;
                                      02 12 001B0       BNEQ    9$                                                     ;
                                      51 D6 001B2       INCL    R1                                                     ;
                                      50 D4 001B4 9$:   CLRL    R0                                                     ;
                           02         55 D1 001B6       CMPL    INDEX_VERSION, #2                                      ;
                                      02 12 001B9       BNEQ    10$                                                    ;
                                      50 D6 001BB       INCL    R0                                                     ;
                           59         51 D2 001BD 10$:  MCOML   R1, R9                                                 ;
                           50         59 CA 001C0       BICL2   R9, R0                                                 ;
                           01         50 D1 001C3       CMPL    R0, #1                                                 ;
                                      05 12 001C6       BNEQ    11$                                                    ;
                           52         01 D0 001C8       MOVL    #1, HEADING_LENGTH                                     ; 1393
                                      02 11 001CB       BRB     12$                                                    ;
                                      53 D4 001CD 11$:  CLRL    CORRECT_FORMAT                                         ; 1399
                           03         53 E8 001CF 12$:  BLBS    CORRECT_FORMAT, 13$                                    ; 1402
                                0081  31 001D2          BRW     18$                                                    ;
                           52         04 C4 001D5 13$:  MULL2   #4, R2                                                 ; 1428
                      40   AE         52 B0 001D8 14$:  MOVW    R2, IOB$+52                                            ;
                      42   AE         02 90 001DC       MOVB    #2, IOB$+54                                            ;
                      38   AE         06 90 001E0       MOVB    #6, IOB$+44                                            ;
                                      5A DD 001E4       PUSHL   R10                                                    ;
                                      7E D4 001E6       CLRL    -(SP)                                                  ;
                              14 AE   9F 001E8          PUSHAB  IOB$                                                   ;
                           6B         03 FB 001EB       CALLS   #3, XPO$GET                                            ;
                00208001   8F         50 D1 001EE       CMPL    R0, #2129921                                           ;
                                      03 13 001F5       BEQL    15$                                                    ;
```

```
                                    02C6  31  001F7          BRW     38$
                                50      44  AE  D0  001FA 15$:  MOVL    INPUT_IOB+56, R0                    ; 1432
                                38          54  E9  001FE      BLBC    R4, 17$                             ; 1434
                                01          60  D1  00201      CMPL    (R0), #1
                                            1A  13  00204      BEQL    16$
                40  AE      04  A0          04  A5  00206      MULW3   #4, 4(R0), IOB$+52                  ; 1441
                            42  AE          02  90  0020C      MOVB    #2, IOB$+54
                            38  AE          06  90  00210      MOVB    #6, IOB$+44
                                            5A  DD  00214      PUSHL   R10
                                            7E  D4  00216      CLRL    -(SP)
                                        14  AE  9F  00218      PUSHAB  IOB$
                                6B          03  FB  0021B      CALLS   #3, XPO$GET
                                            B8  11  0021E      BRB     14$                                 ; 1434
                                16          54  E9  00220 16$:  BLBC    R4, 17$                             ; 1453
                            40  AE          04  B0  00223      MOVW    #4, IOB$+52
                            42  AE          02  90  00227      MOVB    #2, IOB$+54
                            38  AE          06  90  0022B      MOVB    #6, IOB$+44
                                            5A  DD  0022F      PUSHL   R10
                                            7E  D4  00231      CLRL    -(SP)
                                        14  AE  9F  00233      PUSHAB  IOB$
                                6B          03  FB  00236      CALLS   #3, XPO$GET
                                51      44  BE  3C  00239 17$:  MOVZWL  @INPUT_IOB+56, RECORD_TYPE         ; 1455
                                01          51  D1  0023D      CMPL    RECORD_TYPE, #1                     ; 1460
                                            29  12  00240      BNEQ    20$
            55      44  BE      F0          8F  78  00242      ASHL    #-16, @INPUT_IOB+56, INDEX_VERSION  ; 1462
                                02          55  D1  00248      CMPL    INDEX_VERSION, #2                   ; 1464
                                            09  12  0024B      BNEQ    18$
                FFFFFFFF    8F              56  D1  0024D      CMPL    FILE_IDENTIFIER, #-1                ; 1465
                                            82  13  00254      BEQL    14$
                                    28      AE  9F  00256 18$:  PUSHAB  INPUT_IOB+28                        ; 1475
                                            01  DD  00259      PUSHL   #1
                                00000000G   8F  DD  0025B      PUSHL   #DSRINDEX$_INVINPUT
                00000000G   00              03  FB  00261 19$:  CALLS   #3, LIB$SIGNAL
                                        0297    31  00268      BRW     41$                                 ; 1485
                                02          51  D1  0026B 20$:  CMPL    RECORD_TYPE, #2                     ; 1491
                                            3F  12  0026E      BNEQ    22$
                                            57  D4  00270      CLRL    CONFUSE_COUNT                       ; 1496
                            40  AE          14  B0  00272      MOVW    #20, IOB$+52                        ; 1497
                            42  AE          02  90  00276      MOVB    #2, IOB$+54
                            38  AE          06  90  0027A      MOVB    #6, IOB$+44
                                            5A  DD  0027E      PUSHL   R10
                                            7E  D4  00280      CLRL    -(SP)
                                        14  AE  9F  00282      PUSHAB  IOB$
                                6B          03  FB  00285      CALLS   #3, XPO$GET
                                50      44  AE  D0  00288      MOVL    INPUT_IOB+56, R0                    ; 1501
                                            60  D5  0028C      TSTL    (R0)                                ; 1504
                                            1C  13  0028E      BEQL    21$
00000000'   EF              60  00000000'   EF  C1  00290      ADDL3   XTN_OFFSET, (R0), HIGHEST_XTN_OUT   ; 1510
                                00000000'   EF  DD  0029C      PUSHL   HIGHEST_XTN_OUT                     ; 1511
                                        04  A0  9F  002A2      PUSHAB  4(R0)
                00000000G   EF              02  FB  002A5      CALLS   #2, ASGXTN
                                        FF29    31  002AC 21$:  BRW     14$                                 ; 1457
                                03          51  D1  002AF 22$:  CMPL    RECORD_TYPE, #3                     ; 1518
                                            03  13  002B2      BEQL    23$
                                        00D4    31  002B4      BRW     30$
                                            57  D4  002B7 23$:  CLRL    CONFUSE_COUNT                       ; 1526
                            40  AE          0C  B0  002B9      MOVW    #12, IOB$+52                        ; 1527
```

```
                        42   AE              02  90 002BD          MOVB    #2, IOB$+54                                 ;
                        38   AE              06  90 002C1          MOVB    #6, IOB$+44                                 ;
                                             5A  DD 002C5          PUSHL   R10                                         ;
                                             7E  D4 002C7          CLRL    -(SP)                                       ;
                                         14  AE  9F 002C9          PUSHAB  IOB$                                        ;
                                     6B          03  FB 002CC          CALLS   #3, XPO$GET                             ;
                                     51      44  AE  D0 002CF          MOVL    INPUT_IOB+56, R1                        ; 1531
                                     50          01  D0 002D3          MOVL    #1, I                                   ; 1533
                        FC AE40          FC A140  D0 002D6 24$:      MOVL    -4(R1)[I], DESCRIPTOR-4[I]              ; 1534
                  F5                 50          03  F3 002DD          AOBLEQ  #3, I, 24$                              ;
                  50                 6E          03  C1 002E1          ADDL3   #3, DESCRIPTOR, R0                      ; 1539
                                     50          04  C6 002E5          DIVL2   #4, R0                                  ;
              40  AE                 50          04  A5 002E8          MULW3   #4, R0, IOB$+52                         ;
                        42   AE              02  90 002ED          MOVB    #2, IOB$+54                                 ;
                        38   AE              06  90 002F1          MOVB    #6, IOB$+44                                 ;
                                             5A  DD 002F5          PUSHL   R10                                         ;
                                             7E  D4 002F7          CLRL    -(SP)                                       ;
                                         14  AE  9F 002F9          PUSHAB  IOB$                                        ;
                                     6B          03  FB 002FC          CALLS   #3, XPO$GET                             ;
                    000004B0         8F          6E  D1 002FF          CMPL    DESCRIPTOR, #1200                       ; 1541
                                                 1F  15 00306          BLEQ    25$                                     ;
                                 00000000G       8F  DD 00308          PUSHL   #DSRINDEX$_TRUNCATED                    ; 1551
                    00000000G    00             01  FB 0030E          CALLS   #1, LIB$SIGNAL                          ;
                                     6E      04B0    8F  3C 00315          MOVZWL  #1200, DESCRIPTOR                       ; 1559
                                             44  AE  DD 0031A          PUSHL   INPUT_IOB+56                            ; 1560
                                             04  AE  DD 0031D          PUSHL   DESCRIPTOR                              ;
                    00000000G    EF              02  FB 00320          CALLS   #2, DMPENT                              ;
                                             04  AE  D5 00327 25$:      TSTL    DESCRIPTOR+4                            ; 1567
                                                 08  13 0032A          BEQL    26$                                     ;
                          04     AE  00000000'   EF  C0 0032C          ADDL2   XTN_OFFSET, DESCRIPTOR+4                ;
              19 00000000G       EF              02  E1 00334 26$:      BBC     #2, CMDBLK+1, 27$                       ; 1570
              28 00000000G       EF              04  E0 0033C          BBS     #4, CMDBLK, 28$                         ; 1576
                                 36  00000000G     EF  E9 00344          BLBC    XPLBLK, 29$                             ; 1577
              2E 00000000G       EF              05  E1 0034B          BBC     #5, XPLBLK, 29$                         ;
                                                 17  11 00353          BRB     28$                                     ; 1588
              0F 00000000G       EF              04  E0 00355 27$:      BBS     #4, CMDBLK, 28$                         ; 1597
                                 08  00000000G     EF  E9 0035D          BLBC    XPLBLK, 28$                             ; 1598
              15 00000000G       EF              05  E0 00364          BBS     #5, XPLBLK, 29$                         ;
                                             08  AE  DD 0036C 28$:      PUSHL   DESCRIPTOR+8                            ; 1609
                                             08  AE  DD 0036F          PUSHL   DESCRIPTOR+4                            ;
                                             4C  AE  DD 00372          PUSHL   INPUT_IOB+56                            ; 1608
                                             0C  AE  DD 00375          PUSHL   DESCRIPTOR                              ;
                    00000000G    EF              04  FB 00378          CALLS   #4, PERMUT                              ;
                                                 58  D6 0037F          INCL    INDEX_RECORDS                           ; 1611
                    00000000G    EF              01  8A 00381 29$:      BICB2   #1, XPLBLK                              ; 1617
                                                FE4D 31  00388          BRW     14$                                     ; 1457
                                                 50  D4 0038B 30$:      CLRL    R0                                      ; 1620
                                     04              51  D1 0038D          CMPL    RECORD_TYPE, #4                         ;
                                                 02  12 00390          BNEQ    31$                                     ;
                                                 50  D6 00392          INCL    R0                                      ;
                                     53              54  D2 00394 31$:      MCOML   R4, R3                                  ;
                                     50              53  CA 00397          BICL2   R3, R0                                  ;
                                     01              50  D1 0039A          CMPL    R0, #1                                  ;
                                                 03  13 0039D          BEQL    32$                                     ;
                                                0102 31  0039F          BRW     36$                                     ;
                                                 57  D4 003A2 32$:      CLRL    CONFUSE_COUNT                           ; 1628
                          40     AE              04  B0 003A4          MOVW    #4, IOB$+52                             ; 1633
```

```
                    42   AE              02   90 003A8          MOVB    #2, IOB$+54                                    ;
                    38   AE              06   90 003AC          MOVB    #6, IOB$+44                                    ;
                                         5A   DD 003B0          PUSHL   R10                                            ;
                                         7E   D4 003B2          CLRL    -(SP)                                          ;
                                   14    AE   9F 003B4          PUSHAB  IOB$                                           ;
                                   6B          03   FB 003B7    CALLS   #3, XPO$GET                                    ;
                 00000000G         EF        44   BE   D0 003BA          MOVL    @INPUT_IOB+56, XPLBLK                  ; 1634
              07 00000000G         EF              03   E1 003C2          BBC     #3, CMDBLK+1, 33$                      ; 1636
                 00000000G         EF              18   8A 003CA          BICB2   #24, XPLBLK                            ; 1644
                         40   AE              04   B0 003D1 33$:    MOVW    #4, IOB$+52                            ; 1650
                    42   AE              02   90 003D5          MOVB    #2, IOB$+54                                    ;
                    38   AE              06   90 003D9          MOVB    #6, IOB$+44                                    ;
                                         5A   DD 003DD          PUSHL   R10                                            ;
                                         7E   D4 003DF          CLRL    -(SP)                                          ;
                                   14    AE   9F 003E1          PUSHAB  IOB$                                           ;
                                   6B          03   FB 003E4    CALLS   #3, XPO$GET                                    ;
                                   53        44   BE   D0 003E7          MOVL    @INPUT_IOB+56, STR_LEN                 ; 1651
                                             1E   13 003EB          BEQL    34$                                    ; 1653
                                   50        03   A3   9E 003ED          MOVAB   3(R3), R0                              ;
                                   50              04   C6 003F1          DIVL2   #4, R0                                 ;
                40   AE            50              04   A5 003F4          MULW3   #4, R0, IOB$+52                        ;
                    42   AE              02   90 003F9          MOVB    #2, IOB$+54                                    ;
                    38   AE              06   90 003FD          MOVB    #6, IOB$+44                                    ;
                                         5A   DD 00401          PUSHL   R10                                            ;
                                         7E   D4 00403          CLRL    -(SP)                                          ;
                                   14    AE   9F 00405          PUSHAB  IOB$                                           ;
                                   6B          03   FB 00408    CALLS   #3, XPO$GET                                    ;
                         04   AE              53   B0 0040B 34$:    MOVW    STR_LEN, $STR$STRING                   ; 1656
                         06   AE              0E   90 0040F          MOVB    #14, $STR$STRING+2                     ;
                         07   AE              01   90 00413          MOVB    #1, $STR$STRING+3                      ;
                         08   AE        44    AE   D0 00417          MOVL    INPUT_IOB+56, $STR$STRING+4            ;
                                   00000000G   EF   9F 0041C          PUSHAB  STR$FAILURE                            ;
                                               7E   D4 00422          CLRL    -(SP)                                  ;
                                   00000000G   EF   9F 00424          PUSHAB  $STR$TARGET                            ;
                                         10    AE   9F 0042A          PUSHAB  $STR$STRING                            ;
                                               7E   D4 0042D          CLRL    -(SP)                                  ;
                    00000000G         EF              05   FB 0042F    CALLS   #5, XST$COPY                           ;
                         40   AE              04   B0 00436          MOVW    #4, IOB$+52                            ; 1661
                         42   AE              02   90 0043A          MOVB    #2, IOB$+54                            ;
                         38   AE              06   90 0043E          MOVB    #6, IOB$+44                            ;
                                              5A   DD 00442          PUSHL   R10                                    ;
                                              7E   D4 00444          CLRL    -(SP)                                  ;
                                        14    AE   9F 00446          PUSHAB  IOB$                                   ;
                                        6B          03   FB 00449    CALLS   #3, XPO$GET                            ;
                                        53        44   BE   D0 0044C          MOVL    @INPUT_IOB+56, STR_LEN         ; 1662
                                                  1E   13 00450          BEQL    35$                            ; 1664
                                        50        03   A3   9E 00452          MOVAB   3(R3), R0                      ;
                                        50              04   C6 00456          DIVL2   #4, R0                         ;
                     40   AE            50              04   A5 00459          MULW3   #4, R0, IOB$+52                ;
                         42   AE              02   90 0045E          MOVB    #2, IOB$+54                            ;
                         38   AE              06   90 00462          MOVB    #6, IOB$+44                            ;
                                              5A   DD 00466          PUSHL   R10                                    ;
                                              7E   D4 00468          CLRL    -(SP)                                  ;
                                        14    AE   9F 0046A          PUSHAB  IOB$                                   ;
                                        6B          03   FB 0046D    CALLS   #3, XPO$GET                            ;
                         04   AE              53   B0 00470 35$:    MOVW    STR_LEN, $STR$STRING                   ; 1667
                         06   AE              0E   90 00474          MOVB    #14, $STR$STRING+2                     ;
```

```
                 07  AE              01  90 00478          MOVB    #1, $STR$STRING+3                          ;
                 08  AE          44  AE  D0 0047C          MOVL    INPUT_IOB+56, $STR$STRING+4                ;
                          00000000G  EF  9F 00481          PUSHAB  STR$FAILURE                                ;
                                     7E  D4 00487          CLRL    -(SP)                                      ;
                          00000000G  EF  9F 00489          PUSHAB  $STR$TARGET                                ;
                                 10  AE  9F 0048F          PUSHAB  $STR$STRING                                ;
                                     7E  D4 00492          CLRL    -(SP)                                      ;
          00000000G  EF              05  FB 00494          CALLS   #5, XST$COPY                               ;
          00000000G  EF              01  88 0049B          BISB2   #1, XPLBLK                                 ; 1672
                                     19  11 004A2          BRB     37$                                        ; 1457
                                     57  D6 004A4 36$:     INCL    CONFUSE_COUNT                              ; 1680
                     05              57  D1 004A6          CMPL    CONFUSE_COUNT, #5                          ; 1682
                                     12  14 004A9          BGTR    37$                                        ;
                                 28  AE  9F 004AB          PUSHAB  INPUT_IOB+28                               ; 1688
                                     01  DD 004AE          PUSHL   #1                                         ;
                          00000000G  8F  DD 004B0          PUSHL   #DSRINDEX$_INVRECORD                       ;
          00000000G  00              03  FB 004B6          CALLS   #3, LIB$SIGNAL                             ;
                                   FD18  31 004BD 37$:     BRW     14$                                        ; 1428
                                     58  D5 004C0 38$:     TSTL    INDEX_RECORDS                              ; 1704
                                     12  12 004C2          BNEQ    39$                                        ;
                                 28  AE  9F 004C4          PUSHAB  INPUT_IOB+28                               ; 1710
                                     01  DD 004C7          PUSHL   #1                                         ;
                          00000000G  8F  DD 004C9          PUSHL   #DSRINDEX$_NOINDEX                         ;
          00000000G  00              03  FB 004CF          CALLS   #3, LIB$SIGNAL                             ;
       12 00000000G  EF              01  E1 004D6 39$:     BBC     #1, CMDBLK+1, 40$                          ; 1719
                                 28  AE  9F 004DE          PUSHAB  INPUT_IOB+28                               ; 1725
                                     01  DD 004E1          PUSHL   #1                                         ;
                          00000000G  8F  DD 004E3          PUSHL   #DSRINDEX$_COMPLETE                        ;
          00000000G  00              03  FB 004E9          CALLS   #3, LIB$SIGNAL                             ;
          00000000'  EF              58  C0 004F0 40$:     ADDL2   INDEX_RECORDS, INDEX_ENTRIES               ; 1734
          00000000'  EF  00000000'   EF  D0 004F7          MOVL    HIGHEST_XTN_OUT, XTN_OFFSET                ; 1735
                 38  AE              02  90 00502 41$:     MOVB    #2, IOB$+44                                ; 1736
                                     5A  DD 00506          PUSHL   R10                                        ;
                                     7E  D4 00508          CLRL    -(SP)                                      ;
                                 14  AE  9F 0050A          PUSHAB  IOB$                                       ;
          00000000G  EF              03  FB 0050D          CALLS   #3, XPO$CLOSE                              ;
                                         04 00514          RET                                                ; 1737

; Routine Size:  1301 bytes,    Routine Base:  $CODE$ + 0000


;  785          1738  1
```

```
;   787     1739  1 %SBTTL 'GLOBAL ROUTINE MAKNDX -- Make output index'
;   788     1740  1
;   789     1741  1 GLOBAL ROUTINE MAKNDX : NOVALUE =
;   790     1742  1
;   791     1743  1 !++
;   792     1744  1 !
;   793     1745  1 ! FUNCTIONAL DESCRIPTION:
;   794     1746  1 !
;   795     1747  1 !       This routine generates an output index if any index entries
;   796     1748  1 !       were processed by NDXINP.
;   797     1749  1 !
;   798     1750  1 !       The internal index and transaction pool space is also released
;   799     1751  1 !       if necessary.
;   800     1752  1 !
;   801     1753  1 ! FORMAL PARAMETERS:
;   802     1754  1 !
;   803     1755  1 !       None.
;   804     1756  1 !
;   805     1757  1 ! IMPLICIT INPUTS:
;   806     1758  1 !
;   807     1759  1 !       CMDBLK          - Command line information block
;   808     1760  1 !       INDEX_ENTRIES   - Number of index entries processed by NDXINP
;   809     1761  1 !
;   810     1762  1 ! IMPLICIT OUTPUTS:
;   811     1763  1 !
;   812     1764  1 !       INDEX_ENTRIES   - is set to zero.
;   813     1765  1 !
;   814     1766  1 !       The internal index and transaction pool space is released if
;   815     1767  1 !       necessary and the global pool variables are initialized.
;   816     1768  1 !
;   817     1769  1 ! ROUTINE VALUE:
;   818     1770  1 ! COMPLETION CODES:
;   819     1771  1 !
;   820     1772  1 !       None.
;   821     1773  1 !
;   822     1774  1 ! SIDE EFFECTS:
;   823     1775  1 !
;   824     1776  1 !       None.
;   825     1777  1 !--
;   826     1778  1
;   827     1779  2     BEGIN
;   828     1780  2
;   829     1781  2     IF (.INDEX_ENTRIES NEQ 0) AND .CMDBLK [NDX$V_OUTPUT]
;   830     1782  2     THEN
;   831     1783  3         BEGIN
;   832     1784  3         !
;   833     1785  3         ! Some index entries were processed by NDXINP and /OUTPUT specified
;   834     1786  3         !
;   835     1787  3         LOCAL
;   836     1788  3             DEFAULT_TYPE;                       ! Pointer to default file type string
;   837     1789  3
;   838     1790  3         SELECTONE .CMDBLK [NDX$H_FORMAT] OF
;   839     1791  3             SET
;   840     1792  3
;   841     1793  3             [DSR]:
;   842     1794  3                 DEFAULT_TYPE = RNX_FILE;        ! Default type is .RNX for DSR
;   843     1795  3
```

```
;   844          1796  3              [TMS11_A, TMS11_E]:
;   845          1797  3                  DEFAULT_TYPE = TMS_FILE;        ! Default type is .TMS for TMS
;   846          1798  3
;   847          1799  3              [TEX]:
;   848          1800  3                  DEFAULT_TYPE = TEX_FILE;        ! Default type is .TEX for TEX
;   849          1801  3
;   850          1802  3              TES;
;   851          1803  3
;   852          1804  3          $XPO_IOB_INIT (IOB = OUTIOB);
;   853        P 1805  3          $XPO_OPEN (IOB = OUTIOB,                ! Open output file.
;   854        P 1806  3              FILE_SPEC = CMDBLK [NDX$T_OUTPUT_FILE], DEFAULT = .DEFAULT_TYPE,
;   855        P 1807  3              RELATED = CMDBLK [NDX$T_RELATED_FILE], OPTIONS = OUTPUT
;   856        P 1808  3              %IF %BLISS (BLISS32) %THEN , FAILURE = OPEN_ERROR %FI
;   857          1809  3              );
;   858          1810  3
;   859          1811  3          NDXFMT ();                              ! Generate final output index.
;   860          1812  3
;   861          1813  3          IF .CMDBLK [NDX$V_LOG]
;   862          1814  3          THEN                                    ! Display /LOG message
;   863          1815  3
;   864        L 1816  3 %IF %BLISS (BLISS32)
;   865          1817  3 %THEN                                            ! Signal error for BLISS32
;   866          1818  3
;   867          1819  3              SIGNAL (INDEX$_CREATED, 1, OUTIOB [IOB$T_RESULTANT]);
;   868          1820  3
;   869        U 1821  3 %ELSE                                            ! Use $XPO_PUT_MSG otherwise
;   870        U 1822  3
;   871        U 1823  3              $XPO_PUT_MSG (SEVERITY = SUCCESS,
;   872        U 1824  3                  STRING = $STR_CONCAT ('created file ''', OUTIOB [IOB$T_RESULTANT], ''''));
;   873        U 1825  3
;   874          1826  3 %FI
;   875          1827  3
;   876          1828  3          $XPO_CLOSE (IOB = OUTIOB);              ! Close output file.
;   877          1829  2          END;
;   878          1830  2
;   879          1831  2      IF .NDXPOL NEQ 0
;   880          1832  2      THEN
;   881          1833  3          BEGIN
;   882          1834  3          !
;   883          1835  3          ! Release internal binary index pool.
;   884          1836  3          !
;   885          1837  3          FPOOL (NDXPOL);
;   886          1838  3          NDXSGE = 0;
;   887          1839  3          NDXSGF = 0;
;   888          1840  2          END;
;   889          1841  2
;   890          1842  2      IF .XTNPOL NEQ 0
;   891          1843  2      THEN
;   892          1844  3          BEGIN
;   893          1845  3          !
;   894          1846  3          ! Release internal transaction pool.
;   895          1847  3          !
;   896          1848  3          FPOOL (XTNPOL);
;   897          1849  3          XTNCNT = 0;
;   898          1850  3          XTNLSP = 0;
;   899          1851  3          XTNLSX = 0;
;   900          1852  3          XTNSGP = 0;
```

```
;  901     1853  3          XTNTAB = 0;
;  902     1854  3          XPAGEN = 0;
;  903     1855  2          END;
;  904     1856  2
;  905     1857  2      INDEX_ENTRIES = 0;                          ! Reset index entry counter
;  906     1858  1      END;


                                                 0FFC 00000          .ENTRY  MAKNDX, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-      ; 1741
                                                                             R11
                                      5B 00000000G   EF  9E 00002    MOVAB   FPOOL, R11                                      ;
                                      5A 00000000G   EF  9E 00009    MOVAB   NDXPOL, R10                                     ;
                                      59 00000000'   EF  9E 00010    MOVAB   INDEX_ENTRIES, R9                               ;
                                      58 00000000G   EF  9E 00017    MOVAB   CMDBLK, R8                                      ;
                                      57 00000000G   EF  9E 0001E    MOVAB   IOB$, R7                                        ;
                                                     69  D5 00025    TSTL    INDEX_ENTRIES                                   ; 1781
                                                     03  12 00027    BNEQ    2$                                              ;
                                                 009B    31 00029 1$:  BRW     7$                                              ;
                                   F9                68      01  E1 0002C 2$:  BBC     #1, CMDBLK, 1$                          ;
                                                     50      04  A8 32 00030    CVTWL   CMDBLK+4, R0                            ; 1790
                                                     01          50  B1 00034    CMPW    R0, #1                              ; 1793
                                                                 06  12 00037    BNEQ    3$                                  ;
                                                     56      04  A9 9E 00039    MOVAB   RNX_FILE, DEFAULT_TYPE                  ; 1794
                                                                 19  11 0003D    BRB     5$                                  ;
                                                     02          50  B1 0003F 3$:  CMPW    R0, #2                              ; 1796
                                                                 0B  19 00042    BLSS    4$                                  ;
                                                     03          50  B1 00044    CMPW    R0, #3                              ;
                                                                 06  14 00047    BGTR    4$                                  ;
                                                     56      0C  A9 9E 00049    MOVAB   TMS_FILE, DEFAULT_TYPE                  ; 1797
                                                                 09  11 0004D    BRB     5$                                  ;
                                                     04          50  B1 0004F 4$:  CMPW    R0, #4                              ; 1799
                                                                 04  12 00052    BNEQ    5$                                  ;
                                                     56      14  A9 9E 00054    MOVAB   TEX_FILE, DEFAULT_TYPE                  ; 1800
                                   00F4   8F         00      6E      00  2C 00058 5$:  MOVC5   #0, (SP), #0, #244, IOB$         ; 1804
                                                                 67      0005F                                                   ;
                                                     67 0301003D     8F  D0 00060    MOVL    #50397245, IOB$                 ;
                                                1E   A7         020E    8F  B0 00067    MOVW    #526, IOB$RESULTANT+2           ;
                                                04   A7           30    A8  9E 0006D    MOVAB   $IOB$FILE_SPEC, IOB$+4          ; 1809
                                                08   A7                 56  D0 00072    MOVL    DEFAULT_TYPE, IOB$+8            ;
                                                0C   A7           40    A8  9E 00076    MOVAB   $IOB$RELATED, IOB$+12           ;
                                                2E   A7                 02  88 0007B    BISB2   #2, IOB$+46                     ;
                                                2C   A7                 01  90 0007F    MOVB    #1, IOB$+44                     ;
                                                         00000000G      EF  9F 00083    PUSHAB  OPEN_ERROR                      ;
                                                                        7E  D4 00089    CLRL    -(SP)                           ;
                                                                        57  DD 0008B    PUSHL   R7                              ;
                                                00000000G   EF                  03  FB 0008D    CALLS   #3, XPO$OPEN            ;
                                                00000000G   EF                  00  FB 00094    CALLS   #0, NDXFMT              ; 1811
                                             12      01     A8                  01  E1 0009B    BBC     #1, CMDBLK+1, 6$        ; 1813
                                                                   1C    A7  9F 000A0    PUSHAB  OUTIOB+28               ; 1819
                                                                         01  DD 000A3    PUSHL   #1                              ;
                                                         00000000G      8F  DD 000A5    PUSHL   #DSRINDEX$_CREATED              ;
                                                00000000G   00                  03  FB 000AB    CALLS   #3, LIB$SIGNAL          ;
                                                     2C     A7                  02  90 000B2 6$:  MOVB    #2, IOB$+44           ; 1828
                                                         00000000G      EF  9F 000B6    PUSHAB  XPO$FAILURE                     ;
```

```
                                 7E  D4 000BC          CLRL    -(SP)                                          ;
                                 57  DD 000BE          PUSHL   R7                                             ;
                     00000000G   EF                   03  FB 000C0          CALLS   #3, XPO$CLOSE                                  ;
                                 6A  D5 000C7 7$:      TSTL    NDXPOL                                         ; 1831
                                 11  13 000C9          BEQL    8$                                             ;
                                 5A  DD 000CB          PUSHL   R10                                            ; 1837
                           6B                         01  FB 000CD          CALLS   #1, FPOOL                                      ;
                               00000000G   EF  D4 000D0          CLRL    NDXSGE                                 ; 1838
                               00000000G   EF  D4 000D6          CLRL    NDXSGF                                 ; 1839
                               00000000G   EF  D5 000DC 8$:      TSTL    XTNPOL                                 ; 1842
                                 2D  13 000E2          BEQL    9$                                             ;
                               00000000G   EF  9F 000E4          PUSHAB  XTNPOL                                 ; 1848
                           6B                         01  FB 000EA          CALLS   #1, FPOOL                                      ;
                               00000000G   EF  D4 000ED          CLRL    XTNCNT                                 ; 1849
                               00000000G   EF  D4 000F3          CLRL    XTNLSP                                 ; 1850
                               00000000G   EF  D4 000F9          CLRL    XTNLSX                                 ; 1851
                               00000000G   EF  D4 000FF          CLRL    XTNSGP                                 ; 1852
                               00000000G   EF  D4 00105          CLRL    XTNTAB                                 ; 1853
                               00000000G   EF  D4 0010B          CLRL    XPAGEN                                 ; 1854
                                 69  D4 00111 9$:      CLRL    INDEX_ENTRIES                                  ; 1857
                                 04  00113          RET                                                    ; 1858

; Routine Size:  276 bytes,    Routine Base:  $CODE$ + 0515

;  907          1859  1
;  908          1860  1 END                                                  ! End of module
;  909          1861  1
;  910          1862  0 ELUDOM


                                                                .EXTRN  LIB$SIGNAL

;                                PSECT SUMMARY

;      Name                      Bytes                         Attributes

;  $OWN$                            52  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                           20  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                         1577  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       187        31       252          00:00.1
```

```
;                                       COMMAND QUALIFIERS

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

; Size:          1577 code + 72 data bytes
; Run Time:         01:01.3
; Elapsed Time:     02:02.7
; Lines/CPU Min:     1823
; Lexemes/CPU-Min: 78031
; Memory Used:  515 pages
; Compilation Complete
```

GETQC
LIS

GNAME
LIS

INDEX
LIS

GLBDAT
LIS

GETLIN
LIS

GETONE
LIS

LAYOUT
LIS

GTABS
LIS

GLNM
LIS

GETQS
LIS

IFIFNE
LIS

GETDD
LIS

GSLU
LIS

LIT
LIS

LIST
LIS

GETNUM
LIS

HEADER
LIS