K 7

```
RRRRRRRRRRRR     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  UUUUUUUUUUUUUUUU  NNN         NNN     000000000     FFF              FFF
RRR         RRR  UUUUUUUUUUUUUUUU  NNN         NNN     000000000     FFF              FFF
RRR         RRR  UUUUUUUUUUUUUUUU  NNN         NNN     000000000     FFF              FFF
```

```
_$2

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

RLI
RNO
RNO
RTY
SAV
STR
STR
STR
STR
STR
STR

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

FILEID**NDXPAG   K 1

```
NN        NN  DDDDDDDD    XX        XX  PPPPPPPP      AAAAAA      GGGGGGGG
NN        NN  DDDDDDDD    XX        XX  PPPPPPPP      AAAAAA      GGGGGGGG
NN        NN  DD      DD  XX        XX  PP      PP  AA      AA  GG
NN        NN  DD      DD  XX        XX  PP      PP  AA      AA  GG
NNNN      NN  DD      DD    XX    XX    PP      PP  AA      AA  GG
NNNN      NN  DD      DD    XX    XX    PP      PP  AA      AA  GG
NN  NN    NN  DD      DD      XX        PPPPPPPP    AA      AA  GG
NN  NN    NN  DD      DD      XX        PPPPPPPP    AA      AA  GG
NN      NNNN  DD      DD    XX    XX    PP          AAAAAAAAAA  GG    GGGGGG
NN      NNNN  DD      DD    XX    XX    PP          AAAAAAAAAA  GG    GGGGGG
NN        NN  DD      DD  XX        XX  PP          AA      AA  GG        GG      ....
NN        NN  DD      DD  XX        XX  PP          AA      AA  GG        GG      ....
NN        NN  DDDDDDDD    XX        XX  PP          AA      AA    GGGGGG          ....
NN        NN  DDDDDDDD    XX        XX  PP          AA      AA    GGGGGG          ....

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
```

ND
VO

```
0 %TITLE 'NDXPAG -- Output page formatting routines'
0 MODULE NDXPAG (IDENT = 'V04-000'
0                     %BLISS32 [, ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE, NONEXTERNAL = LONG_RELATIVE)]
0               ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:
1 !   DSR (Digital Standard RUNOFF) /DSRPLUS DSRINDEX/INDEX Utility
1 !
1 ! ABSTRACT:
1 !   This module contains routines that format the output index pages.
1 !
1 ! ENVIRONMENT:   Transportable
1 !
1 ! AUTHOR:        JPK
1 !
1 ! CREATION DATE: January 1982
1 !
1 ! MODIFIED BY:
1 !
1 !       009     JPK00022        30-Mar-1983
1 !               Modified NDXVMS, NDXFMT, NDXPAG, NDXVMSMSG and NDXVMSREQ
1 !               to generate TEX output. Added module NDXTEX.
1 !
1 !       008     JPK00021        28-Mar-1983
1 !               Modified NDXT20 to include E2.0 functionality.
1 !               Modified NDXCLIDMP, NDXFMT, NDXPAG, NDXVRS to require RNODEF
1 !               for BLISS36 and to remove any conditional require based on
1 !               DSRPLUS_DEF.
1 !
1 !       007     JPK00018        09-Mar-1983
```

```
1 !                     Modified INDEX to handle new BRN format.
1 !                     Modified NDXOUT to handle specifyable levels on SORT= string.
1 !                     Modified NDXFMT to output new RUNOFF prologue.
1 !                     Modified NDXPAG to output new TMS prologue and RUNOFF epilogue.
1 !
1 !       006           JPK00017        23-Feb-1983
1 !                     Modified NDXINI to initialize the zero'th entries of LLINES,
1 !                     RLINES and TLINES which is where the telltale strings are
1 !                     stored by NDXFMT.
1 !                     Modified NDXFMT to write appropriate prologue for /TELLTALE,
1 !                     save the appropriate lines for left and right telltales, and
1 !                     to mark the end of every entry with a NULL.
1 !                     Modified NDXPAG to change the NULL following each entry to a
1 !                     space if LAYOUT is SEPARATE or to a comma otherwise and to
1 !                     generate and output telltales.
1 !
1 !       005           JPK00015        04-Feb-1983
1 !                     Cleaned up module names, modified revision history to
1 !                     conform with established standards. Updated copyright dates.
1 !
1 !       004           JPK00012        24-Jan-1983
1 !                     Modified NDXVMSMSG.MSG to define error messages for both
1 !                     DSRINDEX and INDEX.
1 !                     Added require of NDXVMSREQ.R32 to NDXOUT, NDXFMT, NDXDAT,
1 !                     INDEX, NDXMSG, NDXXTN, NDXTMS, NDXVMS and NDXPAG for BLISS32.
1 !                     Since this file defines the error message literals,
1 !                     the EXTERNAL REFERENCEs for the error message literals
1 !                     have been removed.
1 !
1 !       003           JPK00011        24-Jan-1983
1 !                     Changed CMDBLK [NDX$G_LEVEL] to CMDBLK [NDX$H_LEVEL]
1 !                     Changed CMDBLK [NDX$H_FORMAT] to CMDBLK [NDX$H_LAYOUT]
1 !                     Changed CMDBLK [NDX$V_TMS11] and CMDBLK [NDX$V_TEX] to CMDBLK [NDX$H_FORMAT]
1 !                     Changed comparisons of (.CHRSIZ EQLA CHRSZA) to
1 !                     (.CMDBLK [NDX$H_FORMAT] EQL TMS11_A).
1 !                     Definitions were changed in NDXCLI and references to the
1 !                     effected fields were changed in NDXPAG, NDXFMT, INDEX, NDXVMS
1 !                     and NDXCLIDMP.
1 !
1 !       002           JPK00003        24-Sep-1982
1 !                     Modified NDXPAG for TOPS-20. A 'SIGNAL' was not conditionalized
1 !                     to produce an $XPO_PUT_MSG if not %BLISS (BLISS32).
1 !                     Modified to add requested /TMS=E changes.
1 !
1 !--
```

```
;   104       0103  1 !
;   105       0104  1 ! TABLE OF CONTENTS:
;   106       0105  1 !
;   107       0106  1 FORWARD ROUTINE
;   108       0107  1     PUTPAG        : NOVALUE,                      ! Write a formatted page
;   109       0108  1     VJUST_COL     : NOVALUE,                      ! Vertical justify a column
;   110       0109  1     LASTPG        : NOVALUE,                      ! Format and balance last page
;   111       0110  1     LAST_CONT,                                    ! Generate a continuation heading for last page
;   112       0111  1     GET_EXT_LEN,                                  ! Get external length of line
;   113       0112  1     INDENT_LEVEL,                                 ! Get indent level of string
;   114       0113  1     GUIDE_HEAD    : NOVALUE,                      ! Build a guide head for TMS11
;   115       0114  1     TMSINI        : NOVALUE,                      ! Initialization for TMS11 output
;   116       0115  1     TELLTALE_HEAD : NOVALUE;                      ! Generate a telltale heading
;   117       0116  1
;   118       0117  1 !
;   119       0118  1 ! INCLUDE FILES:
;   120       0119  1 !
;   121       0120  1
;   122       0121  1 LIBRARY 'NXPORT:XPORT';
;   123       0122  1
;   124       0123  1 SWITCHES LIST (REQUIRE);
;   125       0124  1
;   126       0125  1 REQUIRE 'REQ:NDXCLI';
```

```
!                               IDENT = 0V04-00004
!
!****************************************************************************
!*                                                                          *
!*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
!*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
!*  ALL RIGHTS RESERVED.                                                    *
!*                                                                          *
!*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
!*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
!*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
!*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
!*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
!*  TRANSFERRED.                                                            *
!*                                                                          *
!*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
!*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
!*  CORPORATION.                                                            *
!*                                                                          *
!*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
!*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
!*                                                                          *
!*                                                                          *
!****************************************************************************
!


!++
! FACILITY:
!   DSR (Digital Standard RUNOFF) /DSRPLUS DSRINDEX/INDEX Utility
!
! ABSTRACT:       INDEX command line definitions
!
! ENVIRONMENT:   Transportable
!
! AUTHOR:         JPK
!
! CREATION DATE: January 1982
!
! MODIFIED BY:
!
!       004     JPK00015        04-Feb-1983
!               Cleaned up module names, modified revision history to
!               conform with established standards. Updated copyright dates.
!
!       003     JPK00011        24-Jan-1983
!               Changed CMDBLK [NDX$G_LEVEL] to CMDBLK [NDX$H_LEVEL]
!               Changed CMDBLK [NDX$H_FORMAT] to CMDBLK [NDX$H_LAYOUT]
!               Changed CMDBLK [NDX$V_TMS11] and CMDBLK [NDX$V_TEX] to CMDBLK [NDX$H_FORMAT]
!               Changed comparisons of (.CHRSIZ EQLA CHRSZA) to
!               (.CMDBLK [NDX$H_FORMAT] EQL TMS11_A).
!               Definitions were changed in NDXCLI and references to the
!               effected fields were changed in NDXPAG, NDXFMT, INDEX, NDXVMS
!               and NDXCLIDMP.
!
!       002     RER00002        20-Jan-1983
!               Modified VMS command line interface module NDXVMS:
!                 - changed /FORMAT qualifier to /LAYOUT.
```

```
;  R0183  1      !                  - changed use of /RESERVE and /REQUIRE for DSRPLUS.
;  R0184  1      !                  - added code for new DSRPLUS qualifiers /FORMAT and
;  R0185  1      !                    /TELLTALE_HEADINGS.
;  R0186  1      !                Added fields to NDXCLI for new qualifiers: NDX$V_TELLTALE
;  R0187  1      !                and NDX$V_TEX.
;  R0188  1      !                Conditionalized output of NDX$V_PAGE_MERGE in NDXCLIDMP to
;  R0189  1      !                account for different DSR and DSRPLUS default values.
;  R0190  1      !
;  R0191  1      !--
;  R0192  1
```

```
!
!  NDXCMD_FIELDS
!
$FIELD ndxcmd_fields =
    SET

    NDX$V_OPTIONS       = [$INTEGER],           ! Command option indicators:

        $OVERLAY (NDX$V_OPTIONS)

        NDX$V_INPUT_CONCAT      = [$BIT],       ! Input file concatenated to previous
        NDX$V_OUTPUT            = [$BIT],       ! Generate output file
        NDX$V_REQUIRE           = [$BIT],       ! Require file specified
        NDX$V_PAGES             = [$BIT],       ! Include page references in index
        NDX$V_OVERRIDE          = [$BIT],       ! Override master index information
        NDX$V_STANDARD_PAGE     = [$BIT],       ! Generate standard page numbers
        NDX$V_CONTINUATION      = [$BIT],       ! Generate continuation headings
        NDX$V_GUIDE             = [$BIT],       ! Generate guide headings
        NDX$V_WORD_SORT         = [$BIT],       ! Sort entries word by word
        NDX$V_LOG               = [$BIT],       ! Generate /LOG message
        NDX$V_MASTER            = [$BIT],       ! Generate a master index
        NDX$V_PAGE_MERGE        = [$BIT],       ! Merge adjacent page references
        NDX$V_TELLTALE          = [$BIT],       ! Generate telltale headings

        $CONTINUE

    NDX$H_FORMAT        = [$SHORT_INTEGER],     ! Output format: DSR, TMS, TEX
    NDX$H_LAYOUT        = [$SHORT_INTEGER],     ! Output layout type
    NDX$H_NONALPHA      = [$SHORT_INTEGER],     ! Treatment of leading nonalphas during sort
    NDX$H_LEVEL         = [$SHORT_INTEGER],     ! Deepest level to include in index
    NDX$G_COLUMN_WID    = [$INTEGER],           ! Column width
    NDX$G_GUTTER_WID    = [$INTEGER],           ! Gutter width
    NDX$G_LINES_PAGE    = [$INTEGER],           ! Lines per page
    NDX$G_RESERVE_LINES = [$INTEGER],           ! Number of lines to reserve when requiring a file
    NDX$G_SEPARATE_WIDTH= [$INTEGER],           ! Width of reference portion of entry
    NDX$T_MASTER_BOOK   = [$DESCRIPTOR(DYNAMIC)], ! Book name descriptor for Master indexing
    NDX$T_INPUT_FILE    = [$DESCRIPTOR(DYNAMIC)], ! Input file name descriptor
    NDX$T_OUTPUT_FILE   = [$DESCRIPTOR(DYNAMIC)], ! Output file name descriptor
    NDX$T_REQUIRE_FILE  = [$DESCRIPTOR(DYNAMIC)], ! Require file name descriptor
    NDX$T_RELATED_FILE  = [$DESCRIPTOR(DYNAMIC)], ! Related file name descriptor is saved here
                                                  ! by NDXINP for later use by MAKNDX
    NDX$T_COMMAND_LINE  = [$DESCRIPTOR(DYNAMIC)]  ! Copy of entire command line

    TES;
!
!  End of NDXCMD_FIELDS
!

LITERAL
    NDXCMD$K_LENGTH = $FIELD_SET_SIZE;

MACRO
    $NDXCMD = BLOCK [NDXCMD$K_LENGTH] FIELD (NDXCMD_FIELDS) %;

$LITERAL                                ! Output formats (NDX$H_FORMAT)
    DSR                 = $DISTINCT,    ! Runoff
    TMS11_A             = $DISTINCT,    ! TMS=A
```

```
;  R0250  1        TMS11_E               = $DISTINCT,     ! TMS=E
;  R0251  1        TEX                   = $DISTINCT;     ! TEX
;  R0252  1
;  R0253  1    $LITERAL                                   ! Output layouts (NDX$H_LAYOUT)
;  R0254  1        TWO_COLUMN            = $DISTINCT,     ! Normal two column format
;  R0255  1        ONE_COLUMN            = $DISTINCT,     ! Normal one column format
;  R0256  1        SEPARATE              = $DISTINCT,     ! Separate reference format
;  R0257  1        GALLEY                = $DISTINCT;     ! TMS11 Galley format
;  R0258  1
;  R0259  1    $LITERAL                                   ! Treatment of leading nonalphas during sort (NDX$H_NONALPHA)
;  R0260  1        BEFORE                = $DISTINCT,     ! Leading nonalphas sort before alphas
;  R0261  1        AFTER                 = $DISTINCT,     ! Leading nonalphas sort after alphas
;  R0262  1        IGNORE                = $DISTINCT;     ! Leading nonalphas are ignored
;  R0263  1
;  R0264  1    !
;  R0265  1    !--     End of NDXCLI.REQ
;  127          0266  1
;  128          0267  1 REQUIRE 'REQ:NDXLIN';
```

```
!                               IDENT = 0V04-00002
!
!****************************************************************************
!*                                                                          *
!*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
!*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
!*  ALL RIGHTS RESERVED.                                                    *
!*                                                                          *
!*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
!*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
!*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
!*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
!*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
!*  TRANSFERRED.                                                            *
!*                                                                          *
!*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
!*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
!*  CORPORATION.                                                            *
!*                                                                          *
!*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
!*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
!*                                                                          *
!*                                                                          *
!****************************************************************************
!

!++
! FACILITY:
!   DSR (Digital Standard RUNOFF) /DSRPLUS DSRINDEX/INDEX Utility
!
! ABSTRACT:
!   Contains output line type definitions.
!
! ENVIRONMENT:   Transportable
!
! AUTHOR:        JPK
!
! CREATION DATE: January 1982
!
! MODIFIED BY:
!
!       002     JPK00010        04-Feb-1983
!               Cleaned up module names, modified revision history to
!               conform with established standards. Updated copyright dates.
!
!--

LITERAL                                         !Output line types:
    BKT_E = 1,                                  !Bucket end blank line
    FILL = 2,                                   !Fill line
    GUIDE = 3,                                  !Guide heading
    GUIDE_FILL = 4,                             !Blank line after guide heading
    ENTRY_B = 5,                                !Beginning of top level entry
    ENTRY_W = 6,                                !Wrap line of top level entry
    ENTRY_E = 7,                                !Last line of top level entry
    SUB_B = 8,                                  !Beginning of subentry
    SUB_W = 9,                                  !Wrap line of subentry
```

```
;  R0325  1        SUB_E = 10,                         !Last line of subentry
;  R0326  1        CONT_HEAD = 11;                     !Continuation heading
;  R0327  1
;  R0328  1     !
;  R0329  1     !--     End of NDXLIN.REQ
```

```
:  129            0330  1
:  130          L 0331  1 %IF %BLISS (BLISS32)
:  131            0332  1 %THEN
:  132            0333  1
:  133            0334  1 REQUIRE 'REQ:NDXVMSREQ';
```

```
!
! Version:      'V04-000'
!
!****************************************************************************
!*                                                                          *
!*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
!*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
!*  ALL RIGHTS RESERVED.                                                    *
!*                                                                          *
!*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
!*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
!*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
!*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
!*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
!*  TRANSFERRED.                                                            *
!*                                                                          *
!*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
!*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
!*  CORPORATION.                                                            *
!*                                                                          *
!*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
!*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
!*                                                                          *
!*                                                                          *
!****************************************************************************
!
!
!++
! FACILITY:
!   DSR (Digital Standard RUNOFF) /DSRPLUS DSRINDEX/INDEX Utility
!
! ABSTRACT:
!       This file contains external references to the error message numbers
!       for DSRINDEX/INDEX.
!
!       New messages must be defined in NDXVMSMSG.MSG and referenced here:
!       both in the MACRO section (for DSRINDEX) and the EXTERNAL LITERAL
!       section (for INDEX)
!
! ENVIRONMENT:   VAX/VMS User Mode
!
! AUTHOR:        JPK
!
! CREATION DATE: 01-Feb-1983
!
! MODIFIED BY:
!
!       004     JPK00022        30-Mar-1983
!               Modified NDXVMS, NDXFMT, NDXPAG, NDXVMSMSG and NDXVMSREQ
!               to generate TEX output. Added module NDXTEX.
!
!       003     JPK00021        28-Mar-1983
!               Modified NDXT20 to include E2.0 functionality.
!               Modified NDXCLIDMP, NDXFMT, NDXPAG, NDXVRS to require RNODEF
!               for BLISS36 and to remove any conditional require based on
!               DSRPLUS_DEF.
!
```

```
;  R0392  1      !       002     JPK00010        04-Feb-1983
;  R0393  1      !               Cleaned up module names, modified revision history to
;  R0394  1      !               conform with established standards. Updated copyright dates.
;  R0395  1      !
;  R0396  1      !--
;  R0397  1
;  R0398  1      REQUIRE 'REQ:RNODEF';
```

```
!
! Version:      'V04-000'
!
!****************************************************************************
!*                                                                          *
!*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
!*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
!*  ALL RIGHTS RESERVED.                                                    *
!*                                                                          *
!*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
!*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
!*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
!*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
!*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
!*  TRANSFERRED.                                                            *
!*                                                                          *
!*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
!*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
!*  CORPORATION.                                                            *
!*                                                                          *
!*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
!*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
!*                                                                          *
!*                                                                          *
!****************************************************************************
!

!++
! FACILITY:     DSR (Digital Standard RUNOFF) / DSRPLUS
!
! ABSTRACT:
!       Converts BLISS/VARIANT values into useful names.
!
! ENVIRONMENT:  Transportable BLISS
!
! AUTHOR:       Rich Friday
!
! CREATION DATE: 1978
!
! MODIFIED BY:
!
!       016     KAD00016        Ray Marshall    19-Mar-1984
!               Added GERMAN, FRENCH, & ITALIAN.
!
!       015     KAD00015        Keith Dawson    18-Apr-1983
!               Made the LN01 conditional the default for vanilla DSR --
!                 its value is 0 (no variant supplied).
!
!       014     KAD00014        Keith Dawson    22-Mar-1983
!               Asserted the LN01 conditional when DSRPLUS is asserted.
!
!       013     KAD00013        Keith Dawson    20-Mar-1983
!               Removed all references to .BIX and .BTC files.
!
!       012     KAD00012        Keith Dawson    07-Mar-1983
!               Global edit of all modules. Updated module names, idents,
!               copyright dates. Changed require files to BLISS library.
```

```
;  1      !
;  1      !--
;  1
;  1      !++
;  1      !       D E F I N I T I O N   O F  /VARIANT   B I T S
;  1      !
;  1      !       The bit assignments are as follows:
;  1      !
;  1      !       Bit  Weight   Meaning
;  1      !-------------------------------------------------------------------------
;  1      !       --      0     If no /VARIANT is supplied (as for vanilla DSR),
;  1      !                     compile with LN01 support. LN01 support is also
;  1      !                     implied by the DSRPLUS variant.
;  1      !
;  1      !        0      1     CLEAR =   Unassigned
;  1      !                     SET   =   Unassigned
;  1      !
;  1      !        1      2     CLEAR =   Normal compile
;  1      !                     SET   =   Compile for DSRPLUS
;  1      !
;  1      !       4-6    16     CLEAR =   English (American) version
;  1      !                     SET   =   16 = German (Austrian)
;  1      !                               32 = French
;  1      !                               48 = Italian
;  1      !--
;  1
;  1      !-----------------------------------------------------------------------
;  1      !   This variable (LN01) controls whether or not to compile an LN01-flavored
;  1      !   DSR. It is asserted by default, and also whenever DSRPLUS is asserted.
;  1      !
;  1      !   Modules utilizing LN01 are:
;  1      !
;  1      !       DOOPTS  NOUT
;  1
;  1      COMPILETIME
;  1          ln01 =
;  2              ( (%VARIANT EQL 0) OR  %VARIANT/2 )
;  1          ;
;  1
;  1      !-----------------------------------------------------------------------
;  1      !   This variable (DSRPLUS) controls compilation for the DSRPLUS program.
;  1      !
;  1      !   All modules utilize DSRPLUS.
;  1
;  1      COMPILETIME
;  1          dsrplus =
;  2              ( %VARIANT/2 )
;  1          ;
;  1
;  1      !-----------------------------------------------------------------------
;  1      !   This variable (FLIP) controls compilation of FLIP features of DSRPLUS.
;  1      !   It assures that FLIP features are compiled only on VMS systems.
;  1      !
;  1      !   Modules utilizing FLIP are many and various.
;  1
;  1      COMPILETIME
;  1          flip =
;  R0513  2            ( %VARIANT/2 AND %BLISS(BLISS32) )
;  R0514  1        ;
;  R0515  1
;  R0516  1     !----------------------------------------------------------------------
;  R0517  1     !        4-6    16     CLEAR =   English (American) version
;  R0518  1     !                      SET   =   16 = German (Austrian)
;  R0519  1     !                                32 = French
;  R0520  1     !                                48 = Italian
;  R0521  1     COMPILETIME
;  R0522  1         German  = ( %VARIANT/16 AND NOT %VARIANT/32 AND NOT %VARIANT/64 ) ;
;  R0523  1     COMPILETIME
;  R0524  1         French  = ( NOT %VARIANT/16 AND %VARIANT/32 AND NOT %VARIANT/64 ) ;
;  R0525  1     COMPILFTIME
;  R0526  1         Italian = ( %VARIANT/16 AND %VARIANT/32 AND NOT %VARIANT/64 ) ;
;  R0527  1     !----------------------------------------------------------------------
;  R0528  1     !                        End of RNODEF.REQ
```

```

%IF NOT DSRPLUS
%THEN

MACRO
    INDEX$_BADLOGIC  = DSRINDEX$_BADLOGIC  %,
    INDEX$_BADVALUE  = DSRINDEX$_BADVALUE  %,
    INDEX$_INSVIRMEM = DSRINDEX$_INSVIRMEM %,
    INDEX$_LINELENG  = DSRINDEX$_LINELENG  %,
    INDEX$_NOREF     = DSRINDEX$_NOREF     %,
    INDEX$_OPENIN    = DSRINDEX$_OPENIN    %,
    INDEX$_OPENOUT   = DSRINDEX$_OPENOUT   %,
    INDEX$_TOOMANY   = DSRINDEX$_TOOMANY   %,
    INDEX$_VALERR    = DSRINDEX$_VALERR    %,
    INDEX$_CANTBAL   = DSRINDEX$_CANTBAL   %,
    INDEX$_CLOSEQUOT = DSRINDEX$_CLOSEQUOT %,
    INDEX$_CONFQUAL  = DSRINDEX$_CONFQUAL  %,
    INDEX$_CTRLCHAR  = DSRINDEX$_CTRLCHAR  %,
    INDEX$_DOESNTFIT = DSRINDEX$_DOESNTFIT %,
    INDEX$_DUPBEGIN  = DSRINDEX$_DUPBEGIN  %,
    INDEX$_EMPTYIN   = DSRINDEX$_EMPTYIN   %,
    INDEX$_IGNORED   = DSRINDEX$_IGNORED   %,
    INDEX$_INVINPUT  = DSRINDEX$_INVINPUT  %,
    INDEX$_INVRECORD = DSRINDEX$_INVRECORD %,
    INDEX$_LASTCONT  = DSRINDEX$_LASTCONT  %,
    INDEX$_NOBEGIN   = DSRINDEX$_NOBEGIN   %,
    INDEX$_NOEND     = DSRINDEX$_NOEND     %,
    INDEX$_NOINDEX   = DSRINDEX$_NOINDEX   %,
    INDEX$_NOLIST    = DSRINDEX$_NOLIST    %,
    INDEX$_OVERSTRK  = DSRINDEX$_OVERSTRK  %,
    INDEX$_SKIPPED   = DSRINDEX$_SKIPPED   %,
    INDEX$_SYNTAX    = DSRINDEX$_SYNTAX    %,
    INDEX$_TEXFILE   = DSRINDEX$_TEXFILE   %,
    INDEX$_TOODEEP   = DSRINDEX$_TOODEEP   %,
    INDEX$_TOOFEW    = DSRINDEX$_TOOFEW    %,
    INDEX$_TRUNCATED = DSRINDEX$_TRUNCATED %,
    INDEX$_COMPLETE  = DSRINDEX$_COMPLETE  %,
    INDEX$_CREATED   = DSRINDEX$_CREATED   %,
    INDEX$_IDENT     = DSRINDEX$_IDENT     %,
    INDEX$_PROCFILE  = DSRINDEX$_PROCFILE  %,
    INDEX$_TEXT      = DSRINDEX$_TEXT      %,
    INDEX$_TEXTD     = DSRINDEX$_TEXTD     %,
    INDEX$_TMS11     = DSRINDEX$_TMS11     %;

%FI

EXTERNAL LITERAL
    INDEX$_BADLOGIC,     ! <internal logic error detected>
    INDEX$_BADVALUE,     ! <'!AS' is an invalid keyword value>
    INDEX$_INSVIRMEM,    ! <insufficient virtual memory>
    INDEX$_LINELENG,     ! <maximum line length is 120>
    INDEX$_NOREF,        ! <page reference not found>
    INDEX$_OPENIN,       ! <error opening '!AS' for input>
    INDEX$_OPENOUT,      ! <error opening '!AS' for output>
    INDEX$_TOOMANY,      ! <too many values supplied>
    INDEX$_VALERR,       ! <specified value is out of legal range>
    INDEX$_CANTBAL,      ! <can't balance last page>
```

```
;  R0586  1           INDEX$_CLOSEQUOT,     ! <missing close quote>
;  R0587  1           INDEX$_CONFQUAL,      ! <conflicting qualifiers>
;  R0588  1           INDEX$_CTRLCHAR,      ! <the following line contains control characters - ignored>
;  R0589  1           INDEX$_DOESNTFIT,     ! <'!AD' will not fit at the current indentation level>
;  R0590  1           INDEX$_DUPBEGIN,      ! <duplicate .XPLUS (BEGIN) - inserted as .XPLUS ()>
;  R0591  1           INDEX$_EMPTYIN,       ! <empty input file '.AS'>
;  R0592  1           INDEX$_IGNORED,       ! <'!AS' ignored>
;  R0593  1           INDEX$_INVINPUT,      ! <invalid input file format in file '!AS'>
;  R0594  1           INDEX$_INVRECORD,     ! <invalid record type in file '!AS'>
;  R0595  1           INDEX$_LASTCONT,      ! <can't generate continuation heading on last page>
;  R0596  1           INDEX$_NOBEGIN,       ! <.XPLUS (END) with no .XPLUS (BEGIN) - inserted as .XPLUS ()>
;  R0597  1           INDEX$_NOEND,         ! <.XPLUS (BEGIN) has no corresponding .XPLUS (END)>
;  R0598  1           INDEX$_NOINDEX,       ! <no index information in file '!AS'>
;  R0599  1           INDEX$_NOLIST,        ! <parameter list not allowed>
;  R06    1           INDEX$_OVERSTRK,      ! <the following line contains an overstrike sequence>
;  R0601  1           INDEX$_SKIPPED,       ! <!UL reference!%S inside page range - ignored>
;  R0602  1           INDEX$_SYNTAX,        ! <error parsing '!AS'>
;  R0603  1           INDEX$_TEXFILE,       ! <error processing line !UL of TEX character file '!AS'>
;  R0604  1           INDEX$_TOODEEP,       ! <maximum subindex depth exceeded>
;  R0605  1           INDEX$_TOOFEW,        ! <not enough values supplied>
;  R0606  1           INDEX$_TRUNCATED,     ! <string too long - truncated>
;  R0607  1           INDEX$_COMPLETE,      ! <processing complete '!AS'>
;  R0608  1           INDEX$_CREATED,       ! <'!AS' created>
;  R0609  1           INDEX$_IDENT,         ! <INDEX version !AD>
;  R0610  1           INDEX$_PROCFILE,      ! <processing file '!AS'>
;  R0611  1           INDEX$_TEXT,          ! <!AS>
;  R0612  1           INDEX$_TEXTD,         ! <entry text: '!AD'>
;  R0613  1           INDEX$_TMS11;         ! <output file full - continuing with file '!AS'>
;  R0614  1
```

```
  1
  1 %FI
  1
  1 SWITCHES LIST (NOREQUIRE);
  1
  1 !
  1 ! MACROS:
  1 !
  1
  1 MACRO
M 1     PUT_LINE (S) =
  1         $XPO_PUT (IOB = OUTIOB, STRING = S) %;
  1
  1 !
  1 ! EQUATED SYMBOLS:
  1 !
  1 LITERAL
  1     TRUE = 1,
  1     FALSE = 0;
  1
  1 !
  1 ! OWN STORAGE:
  1 !
  1 OWN
  1     BLANKS : INITIAL (CH$PTR (UPLIT ('    '))); !Pointer to blanks
  1
  1 OWN
  1     TMS_TMP     : $STR_DESCRIPTOR (CLASS = DYNAMIC, STRING = (0, 0)),
  1     TMS_TITLE   : $STR_DESCRIPTOR (STRING = '[f7p18]INDEX/l[va96]'),
  1     TMS_GUIDE   : $STR_DESCRIPTOR (STRING = '[f7p12]'),
  1     TMS_LEFT    : $STR_DESCRIPTOR (STRING = '/l'),
  1     TMS_RIGHT   : $STR_DESCRIPTOR (STRING = '/r'),
  1     TMS_TXT_FMT : $STR_DESCRIPTOR (CLASS = DYNAMIC, STRING = (0, 0)),
  1     TMS_TELLTALE: $STR_DESCRIPTOR (CLASS = DYNAMIC, STRING = (0, 0)),
  1     TMS_FOOT    : $STR_DESCRIPTOR (CLASS = DYNAMIC, STRING = (0, 0)),
  1     TMS_PAGE    : $STR_DESCRIPTOR (CLASS = DYNAMIC, STRING = (0, 0));
  1 !
  1 ! EXTERNAL REFERENCES:
  1 !
  1 EXTERNAL LITERAL
  1     MAXLST,                                          ! Maximum subindex depth
  1     TMSSTD,                                          ! Average TMS character size
  1     MSPACE,                                          ! TMS 'em' space size
  1     TMSCOL;                                          ! Default TMS column width
  1
  1 EXTERNAL
  1     CMDBLK : $NDXCMD,                                ! Command line information block
  1     NDXVRL,                                          ! Length of version string
  1     NDXVRP,                                          ! CH$PTR to version string
  1     OUTIOB : $XPO_IOB (),                            ! Output file IOB
  1     PAGENO,                                          ! Page number
  1     TMSTOF : $STR_DESCRIPTOR (),                     ! Top of file string
  1     TMSSIZ,                                          ! Ideal TMS file size in blocks
  1     CHRSIZ : REF VECTOR,                             ! Vector of character sizes for TMS
  1     LSTSTK : VECTOR,                                 ! Subindex stack
  1     ALLOWD,                                          ! Usuable lines per page
  1     LCOUNT,                                          ! Number of lines in left column
```

```
1       RCOUNT,                                     ! Number of lines in right column
1       TCOUNT,                                     ! Number of lines in temp column
1 !
1 ! NOTE: The vectors and blockvectors below have two extra entries allocated
1 !       to avoid needing to subtract 1 all the time (for entry zero),
1 !       and so that there will always be an available line at the end of the column
1 !
1       LTYPE  : VECTOR,                            ! Left column line types
1       LLINES : BLOCKVECTOR [, STR$K_D_BLN],       ! Left column string descriptors
1       RTYPE  : VECTOR,                            ! Right column line types
1       RLINES : BLOCKVECTOR [, STR$K_D_BLN],       ! Right column string descriptors
1       TTYPE  : VECTOR,                            ! Temp column line types
1       TLINES : BLOCKVECTOR [, STR$K_D_BLN];       ! Right column for last page
1
1 EXTERNAL ROUTINE
1       RNOTMS  : NOVALUE,                  ! Convert a line from RUNOFF to TMS format
1       TMSPUT  : NOVALUE,                  ! Put a line to TMS output file
1       RNOTEX  : NOVALUE,                  ! Convert a line from RUNOFF to TEX
1       PADLIN  : NOVALUE;                  ! Pad a line with blanks
```

```
1 %SBTTL 'PUTPAG -- output formatted page'
1 GLOBAL ROUTINE PUTPAG (LAST) : NOVALUE =
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine writes a formatted page to the output file.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       LAST            - TRUE if last page
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       ALLOWD          - number of lines on this page
1 !       LLINES          - left column lines
1 !       RLINES          - right column lines
1 !       TLINES          - temp column lines (used for last page)
1 !       CMDBLK          - command line information block
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       LCOUNT          - set to zero
1 !       RCOUNT          - set to zero
1 !       ALLOWD          - set to value of CMDBLK [NDX$G_LINES_PAGE]
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       None
1 !
1 ! SIDE EFFECTS:
1 !
1 !       None
1 !--
2     BEGIN
2     LOCAL
2         R_COL_LINES : REF BLOCKVECTOR [, STR$K_D_BLN],
2         R_COL_TYPE  : REF VECTOR,
2         IDEAL;
2
2     IDEAL = TRUE;
2     PAGENO = .PAGENO + 1;
2
2     IF .CMDBLK [NDX$V_TELLTALE]
2     THEN
3         BEGIN                                       ! Generate telltale heading
3         TELLTALE_HEAD ();
3         IDEAL = FALSE;
2         END;
2
2     IF .CMDBLK [NDX$H_LAYOUT] EQL TWO_COLUMN
2     THEN
3         BEGIN
3         VJUST_COL (LCOUNT, LLINES, LTYPE);          ! Vertical justify left column
3
3         IF .LAST
```

```
3               THEN
4                   BEGIN
4                   !
4                   ! Doing last page.
4                   ! Right column stuff is stored in temp column after page is balanced
4                   !
4                   VJUST_COL (TCOUNT, TLINES, TTYPE);          ! Vertical justify column
4
4                   R_COL_LINES = TLINES [0,0,0,0,0];           ! Set up pointers to lines
4                   R_COL_TYPE = TTYPE [0];                     ! and line types
4                   END
3               ELSE
4                   BEGIN
4                   !
4                   ! Not last page
4                   !
4                   VJUST_COL (RCOUNT, RLINES, RTYPE);          ! Vertical justify column
4
4                   R_COL_LINES = RLINES [0,0,0,0,0];           ! Set up pointers to lines
4                   R_COL_TYPE = RTYPE [0];                     ! and line types
3                   END;
2               END;
2
2           INCR I FROM 1 TO .ALLOWD DO
3               BEGIN
3               BIND
3                   L = LLINES [.I, 0,0,0,0] : $STR_DESCRIPTOR ();
3
3               LOCAL
3                   LEN,
3                   PTR;
3
3               LEN = .L [STR$H_LENGTH];
3
3               !
3               ! Change NULLs to COMMAs (or spaces if SEPARATE)
3               !
3               PTR = CH$FIND_CH (.L [STR$H_LENGTH], .L [STR$A_POINTER], 0);
3               IF NOT CH$FAIL (.PTR)
3               THEN
4                   BEGIN
4
4                   IF .CMDBLK [NDX$H_LAYOUT] EQL SEPARATE
4                   THEN                                        ! If SEPARATE format
4                       CH$WCHAR (%C' ', .PTR)                  ! - replace with a blank
4                   ELSE                                        ! Otherwise:
4                       CH$WCHAR (%C',', .PTR);                 ! Replace with a comma
4
3                   END;
3
3               IF .CMDBLK [NDX$H_LAYOUT] EQL TWO_COLUMN
3               THEN
4                   BEGIN
4                   !
4                   ! Two column output
4                   !
4                   BIND
```

```
4         R = R_COL_LINES [.I, 0,0,0,0] : $STR_DESCRIPTOR ();
4
4     !
4     ! Change NULLs to COMMAs
4     !
4     PTR = CH$FIND_CH (.R [STR$H_LENGTH], .R [STR$A_POINTER], 0);
4     IF NOT CH$FAIL (.PTR) THEN CH$WCHAR (%C',', .PTR);
4
4     SELECTONE .CMDBLK [NDX$H_FORMAT] OF
4         SET
4
4         [TMS11_A, TMS11_E]:
5             BEGIN
5             !
5             ! TMS11 output
5             !
5             IF .R_COL_TYPE [.I] EQL GUIDE
5             THEN
5                 GUIDE_HEAD (R)
5             ELSE
5                 RNOTMS (.R [STR$H_LENGTH], .R [STR$A_POINTER], R);
5
4             END;
4
4         [TEX]:
5             BEGIN
5             !
5             ! TEX output
5             !
5             IF .R_COL_TYPE [.I] EQL GUIDE
5             THEN
5                 GUIDE_HEAD (R)
5             ELSE
5                 RNOTEX (.R [STR$H_LENGTH], .R [STR$A_POINTER], R);
5
4             END;
4
4         [DSR]:
5             BEGIN
5             !
5             ! RUNOFF output
5             !
5             $STR_APPEND (STRING = R, TARGET = L);       ! Concatenate right column to left
5
5             !
5             ! Remove trailing spaces from line
5             !
5             LEN = .L [STR$H_LENGTH];
5             PTR = CH$PLUS (.L [STR$A_POINTER], .LEN - 1);
5
5             DECR I FROM .L [STR$H_LENGTH] TO 2 DO
5                 IF CH$RCHAR (.PTR) NEQ %C' '
5                 THEN
5                     EXITLOOP
5                 ELSE
6                     BEGIN
6                     LEN = .LEN - 1;
```

```
;  382          0862  6                                   PTR = CH$PLUS (.PTR, -1);
;  383          0863  5                                   END;
;  384          0864  4                           END;
;  385          0865  4
;  386          0866  4                   TES;
;  387          0867  4
;  388          0868  3           END;
;  389          0869  3
;  390          0870  4       IF NOT ((.CMDBLK [NDX$H_LAYOUT] EQL GALLEY) AND (.LTYPE [.I] EQL FILL))
;  391          0871  3       THEN
;  392          0872  4           BEGIN
;  393          0873  4           !
;  394          0874  4           ! Not doing GALLEY output
;  395          0875  4           ! or doing galley output and line type is not FILL
;  396          0876  4           !
;  397          0877  4           SELECTONE .CMDBLK [NDX$H_FORMAT] OF
;  398          0878  4               SET
;  399          0879  4
;  400          0880  4               [TMS11_A, TMS11_E]:
;  401          0881  5                   BEGIN
;  402          0882  5                   !
;  403          0883  5                   ! TMS11 output
;  404          0884  5                   !
;  405          0885  5                   IF .LTYPE [.I] EQL GUIDE
;  406          0886  5                   THEN
;  407          0887  6                       BEGIN
;  408          0888  6                       IF .CMDBLK [NDX$H_LAYOUT] EQL GALLEY THEN IDEAL = TRUE;
;  409          0889  6
;  410          0890  6                       GUIDE_HEAD (L);
;  411          0891  6                       END
;  412          0892  5                   ELSE
;  413          0893  6                       BEGIN
;  414          0894  6                       IF .CMDBLK [NDX$H_LAYOUT] EQL GALLEY THEN IDEAL = FALSE;
;  415          0895  6
;  416          0896  6                       RNOTMS (.L [STR$H_LENGTH], .L [STR$A_POINTER], L);
;  417          0897  5                       END;
;  418          0898  5
;  419          0899  5                   IF .CMDBLK [NDX$H_LAYOUT] EQL TWO_COLUMN
;  420          0900  5                   THEN
;  421          0901  6                       BEGIN
;  422          0902  6                       !
;  423          0903  6                       ! Two column output - append right column to left
;  424          0904  6                       !
;  425          0905  6                       BIND
;  426          0906  6                           R = R_COL_LINES [.I, 0,0,0,0] : $STR_DESCRIPTOR ();
;  427          0907  6
;  428          0908  6                       $STR_APPEND (STRING = $STR_CONCAT ('/u/u', R), TARGET = L);
;  429          0909  5                       END;
;  430          0910  5
;  431          0911  5                   $STR_APPEND (STRING = TMS_LEFT, TARGET = L);
;  432          0912  5                   TMSPUT (.L [STR$H_LENGTH], .L [STR$A_POINTER], OUTIOB, .IDEAL);
;  433          0913  4                   END;
;  434          0914  4
;  435          0915  4               [TEX]:
;  436          0916  5                   BEGIN
;  437          0917  5                   !
;  438          0918  5                   ! TEX output
```

```
  5                             !
  5                             IF .LTYPE [.I] EQL GUIDE
  5                             THEN
  5                                 GUIDE_HEAD (L)
  5                             ELSE
  5                                 RNOTEX (.L [STR$H_LENGTH], .L [STR$A_POINTER], L);
  5
  5                             PUT_LINE ($STR_CONCAT (L, '\hfill'));
  4                             END;
  4
  4                         [DSR]:
  4                             !
  4                             ! RUNOFF output
  4                             !
  4                             PUT_LINE ((.LEN, .L [STR$A_POINTER]));      ! Write the line
  4
  4                         TES;
  4
  4                     IDEAL = FALSE;
  3                     END;
  2                 END;
  2
  2             IF .CMDBLK [NDX$H_LAYOUT] NEQ GALLEY
  2             THEN
  3                 BEGIN
  3                 !
  3                 ! Not doing GALLEY output
  3                 !
  3                 SELECTONE .CMDBLK [NDX$H_FORMAT] OF
  3                     SET
  3
  3                     [TMS11_A, TMS11_E]:
  4                         BEGIN
  4                         !
  4                         ! Write page break format for TMS11
  4                         !
  4                         LOCAL
  4                             JUSTIFY;
  4
  4                         JUSTIFY = (IF .PAGENO THEN TMS_RIGHT ELSE TMS_LEFT);
P 4                         $STR_COPY (TARGET = TMS_TMP,
  4                             STRING = $STR_CONCAT (TMS_FOOT, $STR_ASCII (.PAGENO), '[fr]', .JUSTIFY, TMS_PAGE));
  4
  4                         TMSPUT (.TMS_TMP [STR$H_LENGTH], .TMS_TMP [STR$A_POINTER], OUTIOB, FALSE);
  3                         END;
  3
  3                     [DSR]:
  3                         !
  3                         ! RUNOFF output
  3                         !
  3                         IF (NOT .LAST) THEN PUT_LINE ('.PAGE');
  3
  3                     [TEX]:
  4                         BEGIN
  4                         !
  4                         ! TEX output
  4                         !
```

```
;  496          0976  4                     PUT_LINE ('\endcolumn');
;  497          0977  4
;  498          0978  4                     INCR I FROM 1 TO .ALLOWD DO
;  499          0979  4                         !
;  500          0980  4                         ! Write out right column
;  501          0981  4                         !
;  502          0982  4                         PUT_LINE ($STR_CONCAT (R_COL_LINES [.I, 0,0,0,0], '\hfill'));
;  503          0983  4
;  504          0984  4                     PUT_LINE ($STR_CONCAT ('\botpage {Index-', $STR_ASCII (.PAGENO), '}'));
;  505          0985  3                     END;
;  506          0986  3
;  507          0987  3                 TES;
;  508          0988  3
;  509          0989  2             END;
;  510          0990  2
;  511          0991  2         LCOUNT = 0;
;  512          0992  2         RCOUNT = 0;
;  513          0993  2         ALLOWD = .CMDBLK [NDX$G_LINES_PAGE];
;  514          0994  1         END;


                                                                            .TITLE  NDXPAG NDXPAG -- Output page formatting routine
;                                                                                                                           s
                                                                            .IDENT  \V04-000\

                                                                            .PSECT  $PLIT$,NOWRT,NOEXE,2

                                             20  20  20  20  00000 P.AAA:   .ASCII  \    \                                      ;
5B  6C  2F  58  45  44  4E  49  5D  38  31  70  37  66  5B  00004 P.AAB:   .ASCII  \[f7p18]INDEX/l[va96]\                     ;
                                         5D  36  39  61  76  00013                                                           ;
                                 5D  32  31  70  37  66  5B  00018 P.AAC:   .ASCII  \[f7p12]\                                  ;
                                                     6C  2F  0001F P.AAD:   .ASCII  \/l\                                       ;
                                                     72  2F  00021 P.AAE:   .ASCII  \/r\                                       ;
                                             75  2F  75  2F  00023 P.AAG:   .ASCII  \/u/u\                                     ;
                             6C  6C  69  66  68  5C  00027 P.AAJ:   .ASCII  <92>\hfill\                                 ;
                                         5D  72  66  5B  0002D P.AAM:   .ASCII  \[fr]\                                     ;
                                     45  47  41  50  2E  00031 P.AAO:   .ASCII  \.PAGE\                                    ;
                 6E  6D  75  6C  6F  63  64  6E  65  5C  00036 P.AAP:   .ASCII  <92>\endcolumn\                             ;
                                 6C  6C  69  66  68  5C  00040 P.AAR:   .ASCII  <92>\hfill\                                 ;
78  65  64  6E  49  7B  20  65  67  61  70  74  6F  62  5C  00046 P.AAV:   .ASCII  <92>\botpage {Index-\                       ;
                                                         2D  00055                                                           ;
                                                         7D  00056 P.AAW:   .ASCII  \}\                                        ;

                                                                            .PSECT  $OWN$,NOEXE,2

                                      00000000'  00000 BLANKS: .ADDRESS P.AAA                                               ;
                                          0000   00004 TMS_TMP:.WORD    0                                                   ;
                                        02  0E   00006         .BYTE    14, 2                                               ;
                                      00000000   00008         .LONG    0                                                   ;
                                          0014   0000C TMS_TITLE:
                                                               .WORD    20                                                  ;
                                        01  0E   0000E         .BYTE    14, 1                                               ;
                                      00000000'  00010         .ADDRESS P.AAB                                               ;
                                          0007   00014 TMS_GUIDE:
                                                               .WORD    7                                                   ;
                                        01  0E   00016         .BYTE    14, 1                                               ;
                                      00000000'  00018         .ADDRESS P.AAC                                               ;
```

```
             0002  0001C TMS_LEFT:
                                   .WORD    2
          01   0E  0001E           .BYTE    14, 1
        00000000'  00020           .ADDRESS P.AAD
             0002  00024 TMS_RIGHT:
                                   .WORD    2
          01   0E  00026           .BYTE    14, 1
        00000000'  00028           .ADDRESS P.AAE
             0000  0002C TMS_TXT_FMT:
                                   .WORD    0
          02   0E  0002E           .BYTE    14, 2
         00000000  00030           .LONG    0
             0000  00034 TMS_TELLTALE:
                                   .WORD    0
          02   0E  00036           .BYTE    14, 2
         00000000  00038           .LONG    0
             0000  0003C TMS_FOOT:
                                   .WORD    0
          02   0E  0003E           .BYTE    14, 2
         00000000  00040           .LONG    0
             0000  00044 TMS_PAGE:
                                   .WORD    0
          02   0E  00046           .BYTE    14, 2
         00000000  00048           .LONG    0
             0004  0004C $STR$STRING0:
                                   .WORD    4
          01   0E  0004E           .BYTE    14, 1
        00000000'  00050           .ADDRESS P.AAG
             0006  00054 $STR$STRING1:
                                   .WORD    6
          01   0E  00056           .BYTE    14, 1
        00000000'  00058           .ADDRESS P.AAJ
             0004  0005C $STR$STRING2:
                                   .WORD    4
          01   0E  0005E           .BYTE    14, 1
        00000000'  00060           .ADDRESS P.AAM
             0005  00064 $IOB$OUTPUT:
                                   .WORD    5
          01   0E  00066           .BYTE    14, 1
        00000000'  00068           .ADDRESS P.AAO
             000A  0006C $IOB$OUTPUT:
                                   .WORD    10
          01   0E  0006E           .BYTE    14, 1
        00000000'  00070           .ADDRESS P.AAP
             0006  00074 $STR$STRING1:
                                   .WORD    6
          01   0E  00076           .BYTE    14, 1
        00000000'  00078           .ADDRESS P.AAR
             0010  0007C $STR$STRING0:
                                   .WORD    16
          01   0E  0007E           .BYTE    14, 1
        00000000'  00080           .ADDRESS P.AAV
             0001  00084 $STR$STRING2:
                                   .WORD    1
          01   0E  00086           .BYTE    14, 1
        00000000'  00088           .ADDRESS P.AAW
```

```
$STR$STRING=          TMS_LEFT
$STR$STRING0=         TMS_FOOT
$STR$STRING4=         TMS_PAGE
$STR$TARGET=          TMS_TMP
        .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  MAXLST, TMSSTD, MSPACE
        .EXTRN  TMSCOL, CMDBLK, NDXVRL
        .EXTRN  NDXVRP, OUTIOB, PAGENO
        .EXTRN  TMSTOF, TMSSIZ, CHRSIZ
        .EXTRN  LSTSTK, ALLOWD, LCOUNT
        .EXTRN  RCOUNT, TCOUNT, LTYPE
        .EXTRN  LLINES, RTYPE, RLINES
        .EXTRN  TTYPE, TLINES, RNOTMS
        .EXTRN  TMSPUT, RNOTEX, PADLIN
        .EXTRN  XST$APPEND, STR$FAILURE
        .EXTRN  XST$JOIN, XPO$PUT
        .EXTRN  XPO$FAILURE, XST$COPY
        .EXTRN  XST$ASCII

        .PSECT  $CODE$,NOWRT,2
```

```
                                               0FFC 00000          .ENTRY  PUTPAG, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-  ; 0692
                                                                           R11                                         ;
                          5E                08   C2 00002          SUBL2   #8, SP                                      ;
                          58                01   D0 00005          MOVL    #1, IDEAL                                   ; 0732
                             00000000G      EF   D6 00008          INCL    PAGENO                                      ; 0733
             09 00000000G EF                04   E1 0000E          BBC     #4, CMDBLK+1, 1$                            ; 0735
                00000000V EF                00   FB 00016          CALLS   #0, TELLTALE_HEAD                           ; 0738
                                            58   D4 0001D          CLRL    IDEAL                                       ; 0739
                          01 00000000G      EF   B1 0001F  1$:     CMPW    CMDBLK+6, #1                                ; 0742
                                            6D   12 00026          BNEQ    3$                                          ;
                             00000000G      EF   9F 00028          PUSHAB  LTYPE                                       ; 0745
                             00000000G      EF   9F 0002E          PUSHAB  LLINES                                      ;
                             00000000G      EF   9F 00034          PUSHAB  LCOUNT                                      ;
                00000000V EF                03   FB 0003A          CALLS   #3, VJUST_COL                               ;
                          29          04    AC   E9 00041          BLBC    LAST, 2$                                    ; 0747
                             00000000G      EF   9F 00045          PUSHAB  TTYPE                                       ; 0754
                             00000000G      EF   9F 0004B          PUSHAB  TLINES                                      ;
                             00000000G      EF   9F 00051          PUSHAB  TCOUNT                                      ;
                00000000V EF                03   FB 00057          CALLS   #3, VJUST_COL                               ;
                          59 00000000G      EF   9E 0005E          MOVAB   TLINES, R_COL_LINES                         ; 0756
                          5A 00000000G      EF   9E 00065          MOVAB   TTYPE, R_COL_TYPE                           ; 0757
                                            27   11 0006C          BRB     3$                                          ; 0747
                             00000000G      EF   9F 0006E  2$:     PUSHAB  RTYPE                                       ; 0764
                             00000000G      EF   9F 00074          PUSHAB  RLINES                                      ;
                             00000000G      EF   9F 0007A          PUSHAB  RCOUNT                                      ;
                00000000V EF                03   FB 00080          CALLS   #3, VJUST_COL                               ;
                          59 00000000G      EF   9E 00087          MOVAB   RLINES, R_COL_LINES                         ; 0766
                          5A 00000000G      EF   9E 0008E          MOVAB   RTYPE, R_COL_TYPE                           ; 0767
                          5B 00000000G      EF   D0 00095  3$:     MOVL    ALLOWD, R11                                 ; 0771
                                            53   D4 0009C          CLRL    I                                           ;
                                          0203   31 0009E          BRW     32$                                         ;
                          54 00000000GEF43  7E   00A1      4$:     MOVAQ   LLINES[I], R4                               ; 0774
                          56                64   3C 000A9          MOVZWL  (R4), LEN                                   ; 0780
                          57          04    A4   9E 000AC          MOVAB   4(R4), R7                                   ; 0785
         00   B7          64                00   3A 000B0          LOCC    #0, (R4), @0(R7)                            ;
                                            02   12 000B5          BNEQ    5$                                          ;
                                            51   D4 000B7          CLRL    R1                                          ;
                          55                51   D0 000B9  5$:     MOVL    R1, PTR                                     ;
                                            11   13 000BC          BEQL    7$                                          ; 0786
                          03 00000000G      EF   B1 000BE          CMPW    CMDBLK+6, #3                                ; 0790
                                            05   12 000C5          BNEQ    6$                                          ;
                          65                20   90 000C7          MOVB    #32, (PTR)                                  ; 0792
                                            03   11 000CA          BRB     7$                                          ; 0794
                          65                2C   90 000CC  6$:     MOVB    #44, (PTR)                                  ;
                          01 00000000G      EF   B1 000CF  7$:     CMPW    CMDBLK+6, #1                                ; 0798
                                            67   12 000D6          BNEQ    14$                                         ;
                          52              6943   7E 000D8          MOVAQ   (R_COL_LINES)[I], R2                        ; 0805
         04   B2          62                00   3A 000DC          LOCC    #0, (R2), @4(R2)                            ; 0810
                                            02   12 000E1          BNEQ    8$                                          ;
                                            51   D4 000E3          CLRL    R1                                          ;
                          55                51   D0 000E5  8$:     MOVL    R1, PTR                                     ;
                                            03   13 000E8          BEQL    9$                                          ; 0811
                          65                2C   90  0EA           MOVB    #44, (PTR)                                  ;
                          50 00000000G      EF   32 000ED  9$:     CVTWL   CMDBLK+4, R0                                ; 0813
                          02                50   B1 000F4          CMPW    R0, #2                                      ; 0816
                                            1C   19 000F7          BLSS    10$                                         ;
                          03                50   B1 000F9          CMPW    R0, #3                                      ;
```

```
                                     17  14 000FC        BGTR     10$                                                       ;
                03                 6A43  D1 000FE        CMPL     (R_COL_TYPE)[I], #3                                       ; 0821
                                     1C  13 00102        BEQL     11$                                                       ;
                                     52  DD 00104        PUSHL    R2                                                        ; 0825
                             04      A2  DD 00106        PUSHL    4(R2)                                                     ;
                7E                   62  3C 00109        MOVZWL   (R2), -(SP)                                               ;
       00000000G EF                  03  FB 0010C        CALLS    #3, RNOTMS                                                ;
                                     5F  11 00113        BRB      17$                                                       ; 0813
                04                   50  B1 00115 10$:   CMPW     R0, #4                                                    ; 0829
                                     22  12 00118        BNEQ     13$                                                       ;
                03                 6A43  D1 0011A        CMPL     (R_COL_TYPE)[I], #3                                       ; 0834
                                     0B  12 0011E        BNEQ     12$                                                       ;
                                     52  DD 00120 11$:   PUSHL    R2                                                        ; 0836
       00000000V EF                  01  FB 00122        CALLS    #1, GUIDE_HEAD                                            ;
                                     49  11 00129        BRB      17$                                                       ;
                                     52  DD 0012B 12$:   PUSHL    R2                                                        ; 0838
                             04      A2  DD 0012D        PUSHL    4(R2)                                                     ;
                7E                   62  3C 00130        MOVZWL   (R2), -(SP)                                               ;
       00000000G EF                  03  FB 00133        CALLS    #3, RNOTEX                                                ;
                                     38  11 0013A        BRB      17$                                                       ; 0813
                01                   50  B1 0013C 13$:   CMPW     R0, #1                                                    ; 0842
                                     33  12 0013F 14$:   BNEQ     17$                                                       ;
                   00000000G         EF  9F 00141        PUSHAB   STR$FAILURE                                               ; 0847
                                     7E  D4 00147        CLRL     -(SP)                                                     ;
                                     14  BB 00149        PUSHR    #^M<R2,R4>                                                ;
                                     7E  D4 0014B        CLRL     -(SP)                                                     ;
       00000000G EF                  05  FB 0014D        CALLS    #5, XST$APPEND                                            ;
                56                   64  3C 00154        MOVZWL   (R4), LEN                                                 ; 0852
   51           67                   56  C1 00157        ADDL3    LEN, (R7), R1                                             ; 0853
                55           FF      A1  9E 0015B        MOVAB    -1(R1), PTR                                               ;
                50                   64  3C 0015F        MOVZWL   (R4), I                                                   ; 0855
                                     0B  11 00162        BRB      16$                                                       ;
                20                   65  91 00164 15$:   CMPB     (PTR), #32                                                ; 0856
                                     0B  12 00167        BNEQ     17$                                                       ;
                                     56  D7 00169        DECL     LEN                                                       ; 0861
                                     55  D7 0016B        DECL     PTR                                                       ; 0862
                                     50  D7 0016D        DECL     I                                                         ; 0856
                02                   50  D1 0016F 16$:   CMPL     I, #2                                                     ;
                                     F0  18 00172        BGEQ     15$                                                       ;
                                     51  D4 00174 17$:   CLRL     R1                                                        ; 0870
                04 00000000G         EF  B1 00176        CMPW     CMDBLK+6, #4                                              ;
                                     0F  12 0017D        BNEQ     18$                                                       ;
                                     51  D6 0017F        INCL     R1                                                        ;
                02 00000000GEF43         D1 00181        CMPL     LTYPE[I], #2                                              ;
                                     03  12 00189        BNEQ     18$                                                       ;
                                   0116  31 0018B        BRW      32$                                                       ;
                50 00000000G         EF  32 0018E 18$:   CVTWL    CMDBLK+4, R0                                              ; 0877
                02                   50  B1 00195        CMPW     R0, #2                                                    ; 0880
                                     03  18 00198        BGEQ     20$                                                       ;
                                   008E  31 0019A 19$:   BRW      26$                                                       ;
                03                   50  B1 0019D 20$:   CMPW     R0, #3                                                    ;
                                     F8  14 001A0        BGTR     19$                                                       ;
                03 00000000GEF43         D1 001A2        CMPL     LTYPE[I], #3                                              ; 0885
                                     11  12 001AA        BNEQ     22$                                                       ;
                03                   51  E9 001AC        BLBC     R1, 21$                                                   ; 0888
                58                   01  D0 001AF        MOVL     #1, IDEAL                                                 ;
                                     54  DD 001B2 21$:   PUSHL    R4                                                        ; 0890
                   00000000V  EF              01  FB 001B4         CALLS    #1, GUIDE_HEAD                     :
                                              13  11 001BB         BRB      24$                                : 0885
                              02              51  E9 001BD 22$:    BLBC     R1, 23$                            : 0894
                                              58  D4 001C0         CLRL     IDEAL                              :
                                              54  DD 001C2 23$:    PUSHL    R4                                 : 0896
                                              67  DD 001C4         PUSHL    (R7)                               :
                              7E              64  3C 001C6         MOVZWL   (R4), -(SP)                        :
                   00000000G  EF              03  FB 001C9         CALLS    #3, RNOTMS                         :
                              01 00000000G    EF  B1 001D0 24$:    CMPW     CMDBLK+6, #1                       : 0899
                                              23  12 001D7         BNEQ     25$                                :
                                            6943  7F 001D9         PUSHAQ   (R_COL_LINES)[I]                   : 0908
                                 00000000'    EF  9F 001DC         PUSHAB   $STR$STRING0                       :
                   00000000G  EF              02  FB 001E2         CALLS    #2, XST$JOIN                       :
                                 00000000G    EF  9F 001E9         PUSHAB   STR$FAILURE                        :
                                              7E  D4 001EF         CLRL     -(SP)                              :
                                              11  BB 001F1         PUSHR    #^M<R0,R4>                         :
                                              7E  D4 001F3         CLRL     -(SP)                              :
                   00000000G  EF              05  FB 001F5         CALLS    #5, XST$APPEND                     :
                                 00000000G    EF  9F 001FC 25$:    PUSHAB   STR$FAILURE                        : 0911
                                              7E  D4 00202         CLRL     -(SP)                              :
                                              54  DD 00204         PUSHL    R4                                 :
                                 00000000'    EF  9F 00206         PUSHAB   $STR$STRING                        :
                                              7E  D4 0020C         CLRL     -(SP)                              :
                   00000000G  EF              05  FB 0020E         CALLS    #5, XST$APPEND                     :
                                              58  DD 00215         PUSHL    IDEAL                              : 0912
                                 00000000G    EF  9F 00217         PUSHAB   OUTIOB                             :
                                              67  DD 0021D         PUSHL    (R7)                               :
                              7E              64  3C 0021F         MOVZWL   (R4), -(SP)                        :
                   00000000G  EF              04  FB 00222         CALLS    #4, TMSPUT                         :
                                              77  11 00229         BRB      31$                                : 0877
                              04              50  B1 0022B 26$:    CMPW     R0, #4                             : 0915
                                              3B  12 0022E         BNEQ     29$                                :
                              03 00000000GEF43    D1 00230         CMPL     LTYPE[I], #3                       : 0920
                                              0B  12 00238         BNEQ     27$                                :
                                              54  DD 0023A         PUSHL    R4                                 : 0922
                   00000000V  EF              01  FB 0023C         CALLS    #1, GUIDE_HEAD                     :
                                              0E  11 00243         BRB      28$                                :
                                              54  DD 00245 27$:    PUSHL    R4                                 : 0924
                                              67  DD 00247         PUSHL    (R7)                               :
                              7E              64  3C 00249         MOVZWL   (R4), -(SP)                        :
                   00000000G  EF              03  FB 0024C         CALLS    #3, RNOTEX                         :
                                 00000000'    EF  9F 00253 28$:    PUSHAB   $STR$STRING1                       : 0926
                                              54  DD 00259         PUSHL    R4                                 :
                   00000000G  EF              02  FB 0025B         CALLS    #2, XST$JOIN                       :
                   00000000G  EF              50  D0 00262         MOVL     R0, IOB$+68                        :
                                              1B  11 00269         BRB      30$                                :
                              01              50  B1 0026B 29$:    CMPW     R0, #1                             : 0929
                                              32  12 0026E         BNEQ     31$                                :
                              6E              56  B0 00270         MOVW     LEN, $IOB$OUTPUT                   : 0933
                      02      AE              0E  90 00273         MOVB     #14, $IOB$OUTPUT+2                 :
                      03      AE              01  90 00277         MOVB     #1, $IOB$OUTPUT+3                  :
                      04      AE              67  D0 0027B         MOVL     (R7), $IOB$OUTPUT+4                :
                   00000000G  EF              6E  9E 0027F         MOVAB    $IOB$OUTPUT, IOB$+68               :
                   00000000G  EF              07  90 00286 30$:    MOVB     #7, IOB$+44                        :
                                 00000000G    EF  9F 0028D         PUSHAB   XPO$FAILURE                        :
                                              7E  D4 00293         CLRL     -(SP)                              :
                                 00000000G    EF  9F 00295         PUSHAB   IOB$                               :
```

```
                      00000000G  EF              03  FB 0029B          CALLS   #3, XPO$PUT                          ;
                                                 58  D4 002A2 31$:     CLRL    IDEAL                                ; 0937
       FDF7          53          01              5B  F1 002A4 32$:     ACBL    R11, #1, I, 4$                       ; 0771
                                 04 00000000G    EF  B1 002AA          CMPW    CMDBLK+6, #4                         ; 0941
                                                 03  12 002B1          BNEQ    33$                                  ;
                                               0165  31 002B3          BRW     44$                                  ;
                                 50 00000000G    EF  32 002B6 33$:     CVTWL   CMDBLK+4, R0                         ; 0947
                                 02              50  B1 002BD          CMPW    R0, #2                               ; 0950
                                                 03  18 002C0          BGEQ    35$                                  ;
                                               0085  31 002C2 34$:     BRW     39$                                  ;
                                 03              50  B1 002C5 35$:     CMPW    R0, #3                               ;
                                                 F8  14 002C8          BGTR    34$                                  ;
                                 09 00000000G    EF  E9 002CA          BLBC    PAGENO, 36$                          ; 0958
                                 53 00000000'    EF  9E 002D1          MOVAB   TMS_RIGHT, JUSTIFY                   ;
                                                 07  11 002D8          BRB     37$                                  ;
                                 53 00000000'    EF  9E 002DA 36$:     MOVAB   TMS_LEFT, JUSTIFY                    ;
                                                 7E  D4 002E1 37$:     CLRL    -(SP)                                ; 0960
                                    00000000G    EF  DD 002E3          PUSHL   PAGENO                               ;
                                 7E      0903    8F  3C 002E9          MOVZWL  #2307, -(SP)                         ;
                      00000000G  EF              03  FB 002EE          CALLS   #3, XST$ASCII                        ;
                                    00000000'    EF  9F 002F5          PUSHAB  $STR$STRING4                         ;
                                                 53  DD 002FB          PUSHL   JUSTIFY                              ;
                                    00000000'    EF  9F 002FD          PUSHAB  $STR$STRING2                         ;
                                                 50  DD 00303          PUSHL   R0                                   ;
                                    00000000'    EF  9F 00305          PUSHAB  $STR$STRING0                         ;
                      00000000G  EF              05  FB 0030B          CALLS   #5, XST$JOIN                         ;
                                    00000000G    EF  9F 00312          PUSHAB  STR$FAILURE                          ;
                                                 7E  D4 00318          CLRL    -(SP)                                ;
                                    00000000'    EF  9F 0031A          PUSHAB  $STR$TARGET                          ;
                                                 50  DD 00320          PUSHL   R0                                   ;
                                                 7E  D4 00322          CLRL    -(SP)                                ;
                      00000000G  EF              05  FB 00324          CALLS   #5, XST$COPY                         ;
                                                 7E  D4 0032B          CLRL    -(SP)                                ; 0962
                                    00000000G    EF  9F 0032D          PUSHAB  OUTIOB                               ;
                                    00000000'    EF  DD 00333          PUSHL   TMS_TMP+4                            ;
                                 7E 00000000'    EF  3C 00339          MOVZWL  TMS_TMP, -(SP)                       ;
                      00000000G  EF              04  FB 00340          CALLS   #4, TMSPUT                           ;
                                               00D1  31 00347 38$:     BRW     44$                                  ; 0947
                                 01              50  B1 0034A 39$:     CMPW    R0, #1                               ; 0965
                                                 12  12 0034D          BNEQ    40$                                  ;
                                 F4        04    AC  E8 0034F          BLBS    LAST, 38$                            ; 0969
                      00000000G  EF 00000000'    EF  9E 00353          MOVAB   $IOB$OUTPUT, IOB$+68                 ;
                                               009E  31 0035E          BRW     43$                                  ;
                                 04              50  B1 00361 40$:     CMPW    R0, #4                               ; 0971
                                                 E1  12 00364          BNEQ    38$                                  ;
                      00000000G  EF 00000000'    EF  9E 00366          MOVAB   $IOB$OUTPUT, IOB$+68                 ; 0976
                      00000000G  EF              07  90 00371          MOVB    #7, IOB$+44                          ;
                                    00000000G    EF  9F 00378          PUSHAB  XPO$FAILURE                          ;
                                                 7E  D4 0037E          CLRL    -(SP)                                ;
                                    00000000G    EF  9F 00380          PUSHAB  IOB$                                 ;
                      00000000G  EF              03  FB 00386          CALLS   #3, XPO$PUT                          ;
                                 54 00000000G    EF  D0 0038D          MOVL    ALLOWD, R4                           ; 0978
                                                 53  D4 00394          CLRL    I                                    ; 0982
                                                 33  11 00396          BRB     42$                                  ;
                                    00000000'    EF  9F 00398 41$:     PUSHAB  $STR$STRING1                         ;
                                               6943  7F 0039E          PUSHAQ  (R_COL_LINES)[I]                     ;
                      00000000G  EF              02  FB 003A1          CALLS   #2, XST$JOIN                         ;
```

```
              00000000G  EF              50  D0 003A8         MOVL    R0, IOB$+68                    ;
              00000000G  EF              07  90 003AF         MOVB    #7, IOB$+44                    ;
                         00000000G  EF   9F 003B6             PUSHAB  XPO$FAILURE                    ;
                                    7E   D4 003BC             CLRL    -(SP)                          ;
                         00000000G  EF   9F 003BE             PUSHAB  IOB$                           ;
              00000000G  EF              03  FB 003C4         CALLS   #3, XPO$PUT                    ;
          C9             53              54  F3 003CB 42$:    AOBLEQ  R4, I, 41$                     ;
                                    7E   D4 003CF             CLRL    -(SP)                          ; 0984
                         00000000G  EF   DD 003D1             PUSHL   PAGENO                         ;
                        7E     0903  8F  3C 003D7             MOVZWL  #2307, -(SP)                   ;
              00000000G  EF              03  FB 003DC         CALLS   #3, XST$ASCII                  ;
                         00000000'  EF   9F 003E3             PUSHAB  $STR$STRING2                   ;
                                    50   DD 003E9             PUSHL   R0                             ;
                         00000000'  EF   9F 003EB             PUSHAB  $STR$STRING0                   ;
              00000000G  EF              03  FB 003F1         CALLS   #3, XST$JOIN                   ;
              00000000G  EF              50  D0 003F8         MOVL    R0, IOB$+68                    ;
              00000000G  EF              07  90 003FF 43$:    MOVB    #7, IOB$+44                    ;
                         00000000G  EF   9F 00406             PUSHAB  XPO$FAILURE                    ;
                                    7E   D4 0040C             CLRL    -(SP)                          ;
                         00000000G  EF   9F 0040E             PUSHAB  IOB$                           ;
              00000000G  EF              03  FB 00414         CALLS   #3, XPO$PUT                    ;
                         00000000G  EF   D4 0041B 44$:        CLRL    LCOUNT                         ; 0991
                         00000000G  EF   D4 00421             CLRL    RCOUNT                         ; 0992
              00000000G  EF 00000000G  EF   D0 00427          MOVL    CMDBLK+20, ALLOWD              ; 0993
                                         04 00432             RET                                    ; 0994

; Routine Size:  1075 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 %SBTTL 'VJUST_COL -- Vertical justify column'
1 ROUTINE VJUST_COL (COUNT, COL_LINES, COL_TYPES) : NOVALUE =
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !      This routine is called by PUTPAG to vertical justify a column.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !      COUNT           - Address of column line counter
1 !      COL_LINES       - Address of column lines blockvector
1 !      COL_TYPES       - Address of column types vector
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !      None
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !      None
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !      None
1 !
1 ! SIDE EFFECTS:
1 !
1 !      None
1 !--
2    BEGIN
2
2    MAP
2        COL_LINES : REF BLOCKVECTOR [, STR$K_D_BLN],
2        COL_TYPES : REF VECTOR;
2
2    BIND
2        COL_COUNT = .COUNT;
2
2    LOCAL
2        N_LINES,
2        INSERT_POINTS,
2        INSERT_TYPE,
2        LINES_PER_INSERT,
2        LINES_LEFT,
2        TO_LINE,
2        FROM_LINE;
2
2     !
2     ! Initialization
2     !
2    N_LINES = 0;
2    INSERT_POINTS = 0;
2    INSERT_TYPE = BKT_E;
2    LINES_PER_INSERT = 0;
2    LINES_LEFT = 0;
```

```
2
2        !
2        ! Get number of blank lines at bottom of column
2        !
2        DECR I FROM .COL_COUNT TO 2 DO
3            IF (.COL_TYPES [.I] NEQ BKT_E) AND (.COL_TYPES [.I] NEQ FILL)
2            THEN
2                EXITLOOP
2            ELSE
2                N_LINES = .N_LINES + 1;
2
2        IF .N_LINES EQL 0 THEN RETURN;                 ! No justification to do
2
2        !
2        ! Get number of primary insertion points
2        !
2        DECR I FROM .COL_COUNT - .N_LINES TO 2 DO
2            IF .COL_TYPES [.I] EQL BKT_E THEN INSERT_POINTS = .INSERT_POINTS + 1;
2
2        IF .INSERT_POINTS EQL 0
2        THEN
3            BEGIN
3            !
3            ! Get number of secondary insertion points
3            !
3            INSERT_TYPE = GUIDE_FILL;
3            DECR I FROM .COL_COUNT - .N_LINES TO 2 DO
3                IF .COL_TYPES [.I] EQL GUIDE_FILL THEN INSERT_POINTS = .INSERT_POINTS + 1;
3
3            IF .INSERT_POINTS EQL 0 THEN RETURN;    ! No place to insert
2            END;
2
2        LINES_PER_INSERT = .N_LINES / .INSERT_POINTS;
2        LINES_LEFT = .N_LINES - (.LINES_PER_INSERT * .INSERT_POINTS);
2
2        TO_LINE = .COL_COUNT;
2        FROM_LINE = .COL_COUNT - .N_LINES;
2
2        WHILE (.INSERT_POINTS NEQ 0) DO
3            BEGIN
3            IF .COL_TYPES [.FROM_LINE] EQL .INSERT_TYPE
3            THEN
4                BEGIN
4                !
4                ! Insert fill line(s)
4                !
4                INCR I FROM 1 TO .LINES_PER_INSERT DO
5                    BEGIN
5                    PADLIN (4, .BLANKS, 4, COL_LINES [.TO_LINE, 0,0,0,0]);
5                    COL_TYPES [.TO_LINE] = FILL;
5                    TO_LINE = .TO_LINE - 1;
4                    END;
4
4                INSERT_POINTS = .INSERT_POINTS - 1;
4
4                IF .LINES_LEFT NEQ 0
4                THEN
```

```
;  630        1109  5                     BEGIN
;  631        1110  5                     !
;  632        1111  5                     ! Insert an extra line
;  633        1112  5                     !
;  634        1113  5                     PADLIN (4, .BLANKS, 4, COL_LINES [.TO_LINE, 0,0,0,0]);
;  635        1114  5                     COL_TYPES [.TO_LINE] = FILL;
;  636        1115  5                     TO_LINE = .TO_LINE - 1;
;  637        1116  5                     LINES_LEFT = .LINES_LEFT - 1;
;  638        1117  4                     END;
;  639        1118  4
;  640        1119  3                 END;
;  641        1120  3
;  642        1121  3             !
;  643        1122  3             ! Insert normal line
;  644        1123  3             !
;  645      P 1124  3             $STR_COPY (STRING = COL_LINES [.FROM_LINE, 0,0,0,0],
;  646        1125  3                 TARGET = COL_LINES [.TO_LINE, 0,0,0,0]);
;  647        1126  3
;  648        1127  3             COL_TYPES [.TO_LINE] = .COL_TYPES [.FROM_LINE];
;  649        1128  3
;  650        1129  3             TO_LINE = .TO_LINE - 1;
;  651        1130  3             FROM_LINE = .FROM_LINE - 1;
;  652        1131  2             END;
;  653        1132  2
;  654        1133  1     END;



                                        OFFC 00000 VJUST_COL:
                                                             .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11         ; 0996
                        5B 00000000'   EF   9E 00002         MOVAB    BLANKS, R11                                  ;
                        5A 00000000G   EF   9E 00009         MOVAB    PADLIN, R10                                  ;
                                       57   7C 00010         CLRQ     INSERT_POINTS                                ; 1048
                        59             01   D0 00012         MOVL     #1, INSERT_TYPE                              ; 1049
                                       52   7C 00015         CLRQ     LINES_PER_INSERT                             ; 1050
                        56       0C    AC   D0 00017         MOVL     COL_TYPES, R6                                ; 1057
                        50       04    BC   D0 0001B         MOVL     @COUNT, I                                    ;
                                       10   11 0001F         BRB      3$                                           ;
                        01           6640   D1 00021 1$:     CMPL     (R6)[I], #1                                  ;
                                       06   13 00025         BEQL     2$                                           ;
                        02           6640   D1 00027         CMPL     (R6)[I], #2                                  ;
                                       09   12 0002B         BNEQ     4$                                           ;
                                       53   D6 0002D 2$:     INCL     N_LINES                                      ; 1061
                                       50   D7 0002F         DECL     I                                            ; 1057
                        02             50   D1 00031 3$:     CMPL     I, #2                                        ;
                                       EB   18 00034         BGEQ     1$                                           ;
                                       53   D5 00036 4$:     TSTL     N_LINES                                      ; 1063
                                       4D   13 00038         BEQL     13$                                          ;
           51       04  BC             53   C3 0003A         SUBL3    N_LINES, @COUNT, R1                          ; 1068
                        50             51   D0 0003F         MOVL     R1, I                                        ;
                                       0A   11 00042         BRB      7$                                           ;
                        01           6640   D1 00044 5$:     CMPL     (R6)[I], #1                                  ; 1069
                                       02   12 00048         BNEQ     6$                                           ;
                                       57   D6 0004A         INCL     INSERT_POINTS                                ;
                                       50   D7 0004C 6$:     DECL     I                                            ;
```

```
                      02              50  D1 0004E 7$:     CMPL    I, #2                                    ;
                                      F1  18 00051         BGEQ    5$                                       ;
                                      57  D5 00053         TSTL    INSERT_POINTS                            ; 1071
                                      1B  12 00055         BNEQ    11$                                      ;
                      59              04  D0 00057         MOVL    #4, INSERT_TYPE                          ; 1077
                      50              51  D0 0005A         MOVL    R1, I                                    ; 1078
                                      0A  11 0005D         BRB     10$                                      ;
                      04            6640  D1 0005F 8$:     CMPL    (R6)[I], #4                              ; 1079
                                      02  12 00063         BNEQ    9$                                       ;
                                      57  D6 00065         INCL    INSERT_POINTS                            ;
                                      50  D7 00067 9$:     DECL    I                                        ;
                      02              50  D1 00069 10$:    CMPL    I, #2                                    ;
                                      F1  18 0006C         BGEQ    8$                                       ;
                                      57  D5 0006E         TSTL    INSERT_POINTS                            ; 1081
                                      77  13 00070         BEQL    17$                                      ;
             52       53              57  C7 00072 11$:    DIVL3   INSERT_POINTS, N_LINES, LINES_PER_INSERT ; 1084
             50       52              57  C5 00076         MULL3   INSERT_POINTS, LINES_PER_INSERT, R0      ; 1085
             58       53              50  C3 0007A         SUBL3   R0, N_LINES, LINES_LEFT                  ;
                      53      04      BC  D0 0007E         MOVL    @COUNT, TO_LINE                          ; 1087
                      54              51  D0 00082         MOVL    R1, FROM_LINE                            ; 1088
                                      57  D5 00085 12$:    TSTL    INSERT_POINTS                            ; 1090
                                      60  13 00087 13$:    BEQL    17$                                      ;
                      59            6644  D1 00089         CMPL    (R6)[FROM_LINE], INSERT_TYPE             ; 1092
                                      36  12 0008D         BNEQ    16$                                      ;
                                      55  D4 0008F         CLRL    I                                        ; 1100
                                      13  11 00091         BRB     15$                                      ;
                              08 BC43  7F 00093 14$:    PUSHAQ  @COL_LINES[TO_LINE]                      ;
                                      04  DD 00097         PUSHL   #4                                       ;
                                      6B  DD 00099         PUSHL   BLANKS                                   ;
                                      04  DD 0009B         PUSHL   #4                                       ;
                      6A              04  FB 0009D         CALLS   #4, PADLIN                               ;
                    6643              02  D0 000A0         MOVL    #2, (R6)[TO_LINE]                        ; 1101
                                      53  D7 000A4         DECL    TO_LINE                                  ; 1102
             E9       55              52  F3 000A6 15$:    AOBLEQ  LINES_PER_INSERT, I, 14$                 ; 1098
                                      57  D7 000AA         DECL    INSERT_POINTS                            ; 1105
                                      58  D5 000AC         TSTL    LINES_LEFT                               ; 1107
                                      15  13 000AE         BEQL    16$                                      ;
                              08 BC43  7F 000B0         PUSHAQ  @COL_LINES[TO_LINE]                      ; 1113
                                      04  DD 000B4         PUSHL   #4                                       ;
                                      6B  DD 000B6         PUSHL   BLANKS                                   ;
                                      04  DD 000B8         PUSHL   #4                                       ;
                      6A              04  FB 000BA         CALLS   #4, PADLIN                               ;
                    6643              02  D0 000BD         MOVL    #2, (R6)[TO_LINE]                        ; 1114
                                      53  D7 000C1         DECL    TO_LINE                                  ; 1115
                                      58  D7 000C3         DECL    LINES_LEFT                               ; 1116
                           00000000G  EF  9F 000C5 16$:    PUSHAB  STR$FAILURE                              ; 1125
                                      7E  D4 000CB         CLRL    -(SP)                                    ;
                              08 BC43  7F 000CD         PUSHAQ  @COL_LINES[TO_LINE]                      ;
                              08 BC44  7F 000D1         PUSHAQ  @COL_LINES[FROM_LINE]                    ;
                                      7E  D4 000D5         CLRL    -(SP)                                    ;
         00000000G  EF                05  FB 000D7         CALLS   #5, XST$COPY                             ;
                    6643            6644  D0 000DE         MOVL    (R6)[FROM_LINE], (R6)[TO_LINE]           ; 1127
                                      53  D7 000E3         DECL    TO_LINE                                  ; 1129
                                      54  D7 000E5         DECL    FROM_LINE                                ; 1130
                                      9C  11 000E7         BRB     12$                                      ; 1090
                                          04 000E9 17$:    RET                                              ; 1133
```

```
; Routine Size:  234 bytes,    Routine Base:  $CODE$ + 0433
```

```
1 %SBTTL 'LASTPG -- Write last page for RUNOFF output'
1 GLOBAL ROUTINE LASTPG : NOVALUE =
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is called to write the last page for RUNOFF
1 !       output. For two column output, the last page is balanced.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       None
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       ALLOWD          - Maximum number of lines allowed for page
1 !       LCOUNT          - Number of lines in left column
1 !       RCOUNT          - Number of lines in right column
1 !       LLINES          - Lines in left column
1 !       RLINES          - Lines in right column
1 !       CMDBLK          - Command line information block
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       None
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       None
1 !
1 ! SIDE EFFECTS:
1 !
1 !       None
1 !--
2    BEGIN
2
2    IF .CMDBLK [NDX$H_LAYOUT] EQL TWO_COLUMN
2    THEN
3       BEGIN
3       !
3       ! Balance last page for two column output
3       !
3       LOCAL
3           MIDPT,                          ! Used to check where left col will break
3           R_FIRST,                        ! First non-continuation line in right col
3           R_LAST,                         ! Last non-blank line in right col
3           R_TOTAL,                        ! Total lines in right column
3           R_BEGIN,                        ! First line in left col which will be in new right col
3           L_END,                          ! Last line in new left col
3           L_LAST;                         ! Last non-blank line in left col
3
3       R_FIRST = 1;
3       R_LAST = .RCOUNT;
3       R_TOTAL = .R_LAST - .R_FIRST + 1;
3       R_BEGIN = .LCOUNT + 1;
3       L_LAST = .LCOUNT;
```

```
3          L_END = .LCOUNT;
3
3          TCOUNT = 0;                              ! No lines in temp column yet
3
3          IF ((.LCOUNT + .RCOUNT + 1) / 2) LSS .ALLOWD - 1
3          THEN
4              BEGIN
4              !
4              ! Page is not full enough to write it out as is.
4              !
4
4              DECR I FROM .LCOUNT TO 2 DO
5                  BEGIN
5                  !
5                  ! Remove blank lines at end of left col
5                  !
5                  L_LAST = .I;
5                  IF (.LTYPE [.I] NEQ FILL) AND (.LTYPE [.I] NEQ BKT_E) THEN EXITLOOP;
4                  END;
4
5              IF (.RCOUNT NEQ 0)
4              THEN
5                  BEGIN
5                  !
5                  ! More than  ne column of output.
5                  !
5
6                  IF (.RTYPE [1] EQL GUIDE) AND (.LTYPE [.L_LAST] NEQ BKT_E)
5                  THEN
6                      BEGIN
6                      !
6                      ! Insert a bucket end line before the guide head
6                      ! that is, at end of current left column.
6                      !
6                      L_LAST = .L_LAST + 1;
6                      PADLIN (4, .BLANKS, 4, LLINES [.L_LAST, 0,0,0,0]);
6                      LTYPE [.L_LAST] = BKT_E;
5                      END;
5
5                  INCR I FROM 1 TO .RCOUNT DO
6                      BEGIN
6                      !
6                      ! Remove continuation head if any
6                      !
6                      R_FIRST = .I;
6                      IF .RTYPE [.I] NEQ CONT_HEAD THEN EXITLOOP;
5                      END;
5
5                  DECR I FROM .RCOUNT TO .R_FIRST DO
6                      BEGIN
6                      !
6                      ! Remove trailing blank lines
6                      !
6                      R_LAST = .I;
6                      IF (.RTYPE [.I] NEQ FILL) AND (.RTYPE [.I] NEQ BKT_E) THEN EXITLOOP;
5                      END;
5
```

```
5            R_TOTAL = .R_LAST - .R_FIRST + 1;
4            END;
4
4        MIDPT = (.L_LAST + .R_TOTAL + 1) / 2;
4
4        !
4        ! Find a good place to break the left column
4        !
4        CASE .LTYPE [.MIDPT] FROM BKT_E TO CONT_HEAD OF
4        SET
4
4        [BKT_E]:                            ! Bucket end
5            BEGIN
5            L_END = .MIDPT - 1;             ! Previous is last left col line
5            R_BEGIN = .MIDPT + 1;           ! Next line is first in right col
4            END;
4
4        [GUIDE]:                            ! Guide head
5            BEGIN
5            L_END = .MIDPT - 2;             ! Line before bucket end is last in left col
5            R_BEGIN = .MIDPT;               ! This line starts right col.
4            END;
4
4        [GUIDE_FILL]:                       ! Line after guide head
5            BEGIN
5            L_END = .MIDPT - 3;             ! Line before bucket end is last in left col
5            R_BEGIN = .MIDPT - 1;           ! Guide head starts right col.
4            END;
4
4        [ENTRY_B TO ENTRY_E]:               ! Somewhere in a top level entry
5            IF (.LTYPE [.MIDPT] EQL ENTRY_E)
5            AND (
6                ((.LTYPE [.MIDPT + 1] NEQ SUB_B) AND (.LTYPE [.MIDPT + 1] NEQ SUB_E))
5                OR NOT .CMDBLK [NDX$V_CONTINUATION]
5                )
4            THEN
5                BEGIN
5                !
5                ! Line is the last line in a top level entry.
5                ! and either the next line is not a subindex entry
5                ! or we aren't doing continuation headings.
5                !
5                ! Use this line as the last in the column.
5                !
5                L_END = .MIDPT;
5                R_BEGIN = .MIDPT + 1;
5                IF .R_BEGIN LEQ .L_LAST
5                THEN
5                    IF .LTYPE [.R_BEGIN] EQL BKT_E THEN R_BEGIN = .R_BEGIN + 1;
5                END
4            ELSE
5                BEGIN
5                IF .LTYPE [.MIDPT] EQL ENTRY_E THEN MIDPT = .MIDPT - 1;
5
5                DECR I FROM .MIDPT TO 1 DO
6                    BEGIN
6                    !
```

```
  6                 ! Find beginning of entry
  6                 !
  6                 L_END = .I;
  6                 R_BEGIN = .I + 1;
  6                 IF (.LTYPE [.I] NEQ ENTRY_B) AND (.LTYPE [.I] NEQ ENTRY_W) THEN EXITLOOP;
  5                 END;
  5
  5             SELECTONE .LTYPE [.L_END] OF
  5             SET
  5
  5             [GUIDE_FILL]:
  6                 BEGIN
  6                 R_BEGIN = .L_END - 1;
  6                 L_END = .L_END - 3;
  5                 END;
  5
  5             [BKT_E]:
  5                 L_END = .L_END - 1;
  5
  5             [OTHERWISE]:
  5                 ;
  5
  5             TES;
  5
  5             IF ((.L_LAST - .R_BEGIN + 1) + .R_TOTAL) GTR .ALLOWD
  5             THEN
  6                 BEGIN
  6                 !
  6                 ! Write page as is
  6                 !
L 6 %IF %BLISS (BLISS32)
  6 %THEN                                       ! Signal errors for BLISS32
  6
  6                 SIGNAL (INDEX$_CANTBAL);
  6
U 6 %ELSE                                       ! Use $XPO_PUT_MSG otherwise
U 6
U 6                 $XPO_PUT_MSG (SEVERITY = WARNING,
U 6                     STRING = 'can''t balance last page.');
U 6
  6 %FI
  6
  6                 R_FIRST = 1;
  6                 R_LAST = .RCOUNT;
  6                 R_TOTAL = .R_LAST - .R_FIRST + 1;
  6                 R_BEGIN = .LCOUNT + 1;
  6                 L_LAST = .LCOUNT;
  6                 L_END = .LCOUNT;
  5                 END;
  5
  4             END;
  4
  4       [SUB_B TO SUB_E]:                 ! Subindex entry
  5           BEGIN
  5           INCR I FROM .MIDPT TO .L_LAST DO
  6               BEGIN
  6               !
```

```
    6                       ! Find end of subindex entry
    6                       !
    6                       L_END = .I;
    6                       IF .LTYPE [.I] EQL SUB_E THEN EXITLOOP;
    5                       END;
    5
    5                   R_BEGIN = .L_END + 1;
    5
    5                   IF .R_BEGIN LEQ .L_LAST
    5                   THEN
    5                       CASE .LTYPE [.R_BEGIN] FROM BKT_E TO CONT_HEAD OF
    5                       SET
    5
    5                       [BKT_E]:
    5                           R_BEGIN = .R_BEGIN + 1;
    5
    5                       [SUB_B, SUB_E]:
    5                           IF .CMDBLK [NDX$V_CONTINUATION]
    5                           THEN
    6                               BEGIN
    6                               !
    6                               ! Must generate continuation head
    6                               !
    6                               IF NOT LAST_CONT (.R_BEGIN)
    7                               OR ((.TCOUNT + (.L_LAST - .R_BEGIN + 1) + .R_TOTAL) GTR .ALLOWD)
    6                               THEN
    7                                   BEGIN
    7                                   !
    7                                   ! Can't generate a continuation heading or
    7                                   ! continuation heading will make right column
    7                                   ! too long. - Can't balance last page
    7                                   !
L   7 %IF %BLISS (BLISS32)
    7 %THEN                                                     ! Signal errors for BLISS32
    7
    7                                   SIGNAL (INDEX$_CANTBAL);
    7
U   7 %ELSE                                                     ! Use $XPO_PUT_MSG otherwise
U   7
U   7                                   $XPO_PUT_MSG (SEVERITY = WARNING, STRING = 'can''t balance last page');
U   7
    7 %FI
    7
    7                                   TCOUNT = 0;
    7                                   R_FIRST = 1;
    7                                   R_LAST = .RCOUNT;
    7                                   R_TOTAL = .R_LAST - .R_FIRST + 1;
    7                                   R_BEGIN = .LCOUNT + 1;
    7                                   L_LAST = .LCOUNT;
    7                                   L_END = .LCOUNT;
    6                                   END;
    5                               END;
    5
    5                       [INRANGE]:
    5                           ;
    5
    5                       TES;
```

```
                END;

            [INRANGE]:                              ! FILL and CONT_HEAD
                ;

            TES;
            END;

        IF .R_BEGIN EQL 1
        THEN
            BEGIN
            !
            ! All of left column is to be copied to the new right column.
            ! This means that the left column is very short so write out
            ! everything in the left column.
            !
            R_BEGIN = .LCOUNT + 1,
            L_END = .LCOUNT;
            L_LAST = .LCOUNT;
            END;

        LCOUNT = .L_END;

        IF .R_BEGIN LEQ .L_LAST
        THEN
            INCR I FROM .R_BEGIN TO .L_LAST DO
                BEGIN
                !
                ! Copy remainder of left column to temp column
                !
                TCOUNT = .TCOUNT + 1;
                TTYPE [.TCOUNT] = .LTYPE [.I];
                $STR_COPY (STRING = LLINES [.I, 0,0,0,0],
                    TARGET = TLINES [.TCOUNT, 0,0,0,0]);
                END;

        IF .R_TOTAL NEQ 0
        THEN
            INCR I FROM .R_FIRST TO .R_LAST DO
                BEGIN
                !
                ! Copy remainder of right column to temp column
                !
                TCOUNT = .TCOUNT + 1;
                TTYPE [.TCOUNT] = .RTYPE [.I];
                $STR_COPY (STRING = RLINES [.I, 0,0,0,0],
                    TARGET = TLINES [.TCOUNT, 0,0,0,0]);
                END;

        IF .LCOUNT LSS .TCOUNT
        THEN
            BEGIN
            !
            ! Insert enough blank lines in left column to make columns even
            !
            INCR I FROM .LCOUNT + 1 TO .TCOUNT DO
                BEGIN
```

```
;  998          1476  5                     LTYPE [.I] = FILL;
;  999          1477  5                     PADLIN (4, .BLANKS, 4, LLINES [.I, 0,0,0,0]);
; 1000          1478  4                     END;
; 1001          1479  4
; 1002          1480  4                 LCOUNT = .TCOUNT;
; 1003          1481  4                 END
; 1004          1482  3             ELSE
; 1005          1483  3                 IF .TCOUNT LSS .LCOUNT
; 1006          1484  3                 THEN
; 1007          1485  4                     BEGIN
; 1008          1486  4                     !
; 1009          1487  4                     ! Insert enough blank lines in right column to make columns even
; 1010          1488  4                     !
; 1011          1489  4                     INCR I FROM .TCOUNT + 1 TO .LCOUNT DO
; 1012          1490  5                         BEGIN
; 1013          1491  5                         TTYPE [.I] = FILL;
; 1014          1492  5                         PADLIN (4, .BLANKS, 4, TLINES [.I, 0,0,0,0]);
; 1015          1493  4                         END;
; 1016          1494  4
; 1017          1495  4                     TCOUNT = .LCOUNT;
; 1018          1496  3                     END;
; 1019          1497  3
; 1020          1498  2             END;
; 1021          1499  2
; 1022          1500  2         ALLOWD = .LCOUNT;
; 1023          1501  2         PUTPAG (TRUE);
; 1024          1502  2
; 1025          1503  2         IF .CMDBLK [NDX$H_FORMAT] EQL DSR THEN PUT_LINE ('.RESTORE');
; 1026          1504  1         END;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                  45  52  4F  54  53  45  52  2E  00057 P.AAZ:  .ASCII  \.RESTORE\                                 ;

                                                                .PSECT  $OWN$,NOEXE,2

                                    0008  0008C $IOB$OUTPUT:
                                                                .WORD   8                                           ;
                                  01  0E  0008E                 .BYTE   14, 1                                       ;
                               00000000'  00090                 .ADDRESS P.AAZ                                      ;



                                                                .PSECT  $CODE$,NOWRT,2

                                     0FFC 00000                 .ENTRY  LASTPG, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-  ; 1135
                                                                        R11                                         ;
                      5B 00000000G  EF  9E 00002                MOVAB   TCOUNT, R11                                 ;
                      5A 00000000G  EF  9E 00009                MOVAB   LTYPE, R10                                  ;
                      01 00000000G  EF  B1 00010                CMPW    CMDBLK+6, #1                                ; 1171
                                    03  13 00017                BEQL    1$                                          ;
                                  0337  31 00019                BRW     53$                                         ;
                      56            01  D0 0001C 1$:            MOVL    #1, R_FIRST                                 ; 1186
                      51 00000000G  EF  D0 0001F                MOVL    RCOUNT, R1                                  ; 1187
                      57            51  D0 00026                MOVL    R1, R_LAST                                  ;
```

```
                     50           57           56  C3 00029          SUBL3   R_FIRST, R_LAST, R0                                       : 1188
                                  55      01   A0  9E 0002D          MOVAB   1(R0), R_TOTAL                                            :
                                  50 00000000G EF  D0 00031          MOVL    LCOUNT, R0                                                : 1189
                                  52      01   A0  9E 00038          MOVAB   1(R0), R_BEGIN                                            :
                                  53           50  D0 0003C          MOVL    R0, L_LAST                                                : 1190
                                  54           50  D0 0003F          MOVL    R0, L_END                                                 : 1191
                                               6B  D4 00042          CLRL    TCOUNT                                                    : 1193
                                  58      01 A140  9E 00044          MOVAB   1(R1)[R0], R8                                             : 1195
                                  58           02  C6 00049          DIVL2   #2, R8                                                    :
                     59 00000000G EF           01  C3 0004C          SUBL3   #1, ALLOWD, R9                                            :
                                  59           58  D1 00054          CMPL    R8, R9                                                    :
                                               14  19 00057          BLSS    4$                                                        :
                                             01F2  31 00059          BRW     40$                                                       :
                                  53           50  D0 0005C 2$:      MOVL    I, L_LAST                                                 : 1207
                                  02         6A40  D1 0005F          CMPL    LTYPE[I], #2                                              : 1208
                                               06  13 00063          BEQL    3$                                                        :
                                  01         6A40  D1 00065          CMPL    LTYPE[I], #1                                              :
                                               07  12 00069          BNEQ    5$                                                        :
                                               50  D7 0006B 3$:      DECL    I                                                         : 1202
                                  02           50  D1 0006D 4$:      CMPL    I, #2                                                     :
                                               EA  18 00070          BGEQ    2$                                                        :
                                               51  D5 00072 5$:      TSTL    R1                                                        : 1211
                                               75  13 00074          BEQL    14$                                                       :
                                  03 00000000G EF  D1 00076          CMPL    RTYPE+4, #3                                               : 1218
                                               24  12 0007D          BNEQ    6$                                                        :
                                  01         6A43  D1 0007F          CMPL    LTYPE[L_LAST], #1                                         :
                                               1E  13 00083          BEQL    6$                                                        :
                                               53  D6 00085          INCL    L_LAST                                                    : 1225
                                     00000000GEF43  7F 00087          PUSHAQ  LINES[L_LAST]                                             : 1226
                                               04  DD 0008E          PUSHL   #4                                                        :
                                     00000000' EF  DD 00090          PUSHL   BLANKS                                                    :
                                               04  DD 00096          PUSHL   #4                                                        :
                        00000000G EF           04  FB 00098          CALLS   #4, PADLIN                                                :
                               6A43            01  D0 0009F          MOVL    #1, LTYPE[L_LAST]                                         : 1227
                                               50  D4 000A3 6$:      CLRL    I                                                         : 1230
                                               0D  11 000A5          BRB     8$                                                        :
                                  56           50  D0 000A7 7$:      MOVL    I, R_FIRST                                                : 1235
                                  0B 00000000GEF40  D1 000AA          CMPL    RTYPE[I], #11                                             : 1236
                                               08  12 000B2          BNEQ    9$                                                        :
                     EB           50 00000000G EF  F3 000B4 8$:      AOBLEQ  RCOUNT, I, 7$                                             : 1230
                                  50 00000000G EF  D0 000BC 9$:      MOVL    RCOUNT, I                                                 : 1239
                                               19  11 000C3          BRB     12$                                                       :
                                  57           50  D0 000C5 10$:     MOVL    I, R_LAST                                                 : 1244
                                  02 00000000GEF40  D1 000C8          CMPL    RTYPE[I], #2                                              : 1245
                                               0A  13 000D0          BEQL    11$                                                       :
                                  01 00000000GEF40  D1 000D2          CMPL    RTYPE[I], #1                                              :
                                               07  12 000DA          BNEQ    13$                                                       :
                                               50  D7 000DC 11$:     DECL    I                                                         : 1239
                                  56           50  D1 000DE 12$:     CMPL    I, R_FIRST                                                :
                                               E2  18 000E1          BGEQ    10$                                                       :
                     50           57           56  C3 000E3 13$:     SUBL3   R_FIRST, R_LAST, R0                                       : 1248
                                  55      01   A0  9E 000E7          MOVAB   1(R0), R_TOTAL                                            :
                                  50      01 A543  9E 000EB 14$:     MOVAB   1(R_TOTAL)[L_LAST], R0                                    : 1251
                                  50           02  C6 000F0          DIVL2   #2, MIDPT                                                 :
                     0A           01         6A40  CF 000F3          CASEL   LTYPE[MIDPT], #1, #10                                     : 1256
       0029        0020         0156         0016     000F8 15$:     .WORD   16$-15$,-                                                 :
       00C2        0034         0034         0034     00100                  40$-15$,-                                                 :
           0156           00C2          00C2      00108                   17$-15$,-                                        ;
                                                                          18$-15$,-                                        ;
                                                                          20$-15$,-                                        ;
                                                                          20$-15$,-                                        ;
                                                                          20$-15$,-                                        ;
                                                                          30$-15$,-                                        ;
                                                                          30$-15$,-                                        ;
                                                                          30$-15$,-                                        ;
                                                                          40$-15$                                          ;
                          54          FF    A0  9E 0010E 16$:    MOVAB    -1(R0), L_END                                    ; 1261
                          52          01    A0  9E 00112         MOVAB    1(R0), R_BEGIN                                   ; 1262
                                            11  11 00116         BRB      19$                                              ; 1256
                          54          FE    A0  9E 00118 17$:    MOVAB    -2(R0), L_END                                    ; 1267
                          52                50  D0 0011C         MOVL     MIDPT, R_BEGIN                                   ; 1268
                                            08  11 0011F         BRB      19$                                              ; 1256
                          54          FD    A0  9E 00121 18$:    MOVAB    -3(R0), L_END                                    ; 1273
                          52          FF    A0  9E 00125         MOVAB    -1(R0), R_BEGIN                                  ; 1274
                                          0122  31 00129 19$:    BRW      40$                                              ; 1256
                                            51  D4 0012C 20$:    CLRL     R1                                               ; 1278
                          07              6A40  D1 0012F         CMPL     LTYPE[MIDPT], #7                                 ;
                                            2D  12 00132         BNEQ     23$                                              ;
                                            51  D6 00134         INCL     R1                                               ;
                          08          04  AA40  D1 00136         CMPL     LTYPE+4[MIDPT], #8                               ; 1280
                                            07  13 0013B         BEQL     21$                                              ;
                          0A          04  AA40  D1 0013D         CMPL     LTYPE+4[MIDPT], #10                              ;
                                            08  12 00142         BNEQ     22$                                              ;
             15 00000000G EF                06  E0 00144 21$:    BBS      #6, CMDBLK, 23$                                  ; 1281
                          54                50  D0 0014C 22$:    MOVL     MIDPT, L_END                                     ; 1292
                          52          01    A0  9E 0014F         MOVAB    1(R0), R_BEGIN                                   ; 1293
                          53                52  D1 00153         CMPL     R_BEGIN, L_LAST                                  ; 1294
                                            D1  14 00156         BGTR     19$                                              ;
                          01              6A42  D1 00158         CMPL     LTYPE[R_BEGIN], #1                               ; 1296
                                            CB  12 0015C         BNEQ     19$                                              ;
                                          008E  31 0015E         BRW      35$                                              ;
                          02                51  E9 00161 23$:    BLBC     R1, 24$                                          ; 1300
                                            50  D7 00164         DECL     MIDPT                                            ;
                                            50  D6 00166 24$:    INCL     I                                                ; 1302
                                            13  11 00168         BRB      26$                                              ;
                          54                50  D0 0016A 25$:    MOVL     I, L_END                                         ; 1307
                          52          01    A0  9E 0016D         MOVAB    1(R0), R_BEGIN                                   ; 1308
                          05              6A40  D1 00171         CMPL     LTYPE[I], #5                                     ; 1309
                                            06  13 00175         BEQL     26$                                              ;
                          06              6A40  D1 00177         CMPL     LTYPE[I], #6                                     ;
                                            03  12 0017B         BNEQ     27$                                              ;
                          EA                50  F5 0017D 26$:    SOBGTR   I, 25$                                           ; 1302
                          50              6A44  D0 00180 27$:    MOVL     LTYPE[L_END], R0                                 ; 1312
                          04                50  D1 00184         CMPL     R0, #4                                           ; 1315
                                            09  12 00187         BNEQ     28$                                              ;
                          52          FF    A4  9E 00189         MOVAB    -1(R4), R_BEGIN                                  ; 1317
                          54                03  C2 0018D         SUBL2    #3, L_END                                        ; 1318
                                            07  11 00190         BRB      29$                                              ; 1312
                          01                50  D1 00192 28$:    CMPL     R0, #1                                           ; 1321
                                            02  12 00195         BNEQ     29$                                              ;
                                            54  D7 00197         DECL     L_END                                            ; 1322
           50             53                52  C3 00199 29$:    SUBL3    R_BEGIN, L_LAST, R0                              ; 1329
                          50          01  A540  9E 0019D         MOVAB    1(R_TOTAL)[R0], R0                               ;
                00000000G EF                50  D1 001A2         CMPL     R0, ALLOWD                                       ;
```

```
                                               6F  15 001A9       BLEQ    37$                                    ;
                                  00000000G    8F  DD 001AB       PUSHL   #DSRINDEX$_CANTBAL                     ; 1338
                      00000000G  00            01  FB 001B1       CALLS   #1, LIB$SIGNAL                         ;
                                               71  11 001B8       BRB     39$                                    ; 1347
                                               50  D7 001BA  30$: DECL    I                                      ; 1359
                                               09  11 001BC       BRB     32$                                    ;
                                 54            50  D0 001BE  31$: MOVL    I, L_END                               ; 1364
                                 0A          6A40  D1 001C1       CMPL    LTYPE[I], #10                          ; 1365
                                               04  13 001C5       BEQL    33$                                    ;
                     F3          50            53  F3 001C7  32$: AOBLEQ  L_LAST, I, 31$                         ; 1359
                                 52      01    A4  9E 001CB  33$: MOVAB   1(R4), R_BEGIN                         ; 1368
                                 53            52  D1 001CF       CMPL    R_BEGIN, L_LAST                        ; 1370
                                               7A  14 001D2       BGTR    40$                                    ;
                     0A          01          6A42  CF 001D4       CASEL   LTYPE[R_BEGIN], #1, #10                ; 1372
        0075       0075        0075          0016     001D9  34$: .WORD   35$-34$,-                              ;
        001A       0075        0075          0075     001E1               40$-34$,-                              ;
                   0075        001A          0075     001E9               40$-34$,-                              ;
                                                                          40$-34$,-                              ;
                                                                          40$-34$,-                              ;
                                                                          40$-34$,-                              ;
                                                                          40$-34$,-                              ;
                                                                          36$-34$,-                              ;
                                                                          40$-34$,-                              ;
                                                                          36$-34$,-                              ;
                                                                          40$-34$                                ;
                                               52  D6 001EF  35$: INCL    R_BEGIN                                ; 1376
                                               5B  11 001F1       BRB     40$                                    ;
                   53 00000000G  EF            06  E1 001F3  36$: BBC     #6, CMDBLK, 40$                        ; 1379
                                               52  DD 001FB       PUSHL   R_BEGIN                                ; 1385
                      00000000V  EF            01  FB 001FD       CALLS   #1, LAST_CONT                          ;
                                 15            50  E9 00204       BLBC    R0, 38$                                ;
                     50          53            52  C3 00207       SUBL3   R_BEGIN, L_LAST, R0                    ; 1386
                                 50            6B  C0 0020B       ADDL2   TCOUNT, R0                             ;
                                 50      01  A540  9E 0020E       MOVAB   1(R_TOTAL)[R0], R0                     ;
                      00000000G  EF            50  D1 00213       CMPL    R0, ALLOWD                             ;
                                               32  15 0021A  37$: BLEQ    40$                                    ;
                                  00000000G    8F  DD 0021C  38$: PUSHL   #DSRINDEX$_CANTBAL                     ; 1397
                      00000000G  00            01  FB 00222       CALLS   #1, LIB$SIGNAL                         ;
                                               6B  D4 00229       CLRL    TCOUNT                                 ; 1405
                                 56            01  D0 0022B  39$: MOVL    #1, R_FIRST                            ; 1406
                                 57 00000000G  EF  D0 0022E       MOVL    RCOUNT, R_LAST                         ; 1407
                     50          57            56  C3 00235       SUBL3   R_FIRST, R_LAST, R0                    ; 1408
                                 55      01    A0  9E 00239       MOVAB   1(R0), R_TOTAL                         ;
                                 50 00000000G  EF  D0 0023D       MOVL    LCOUNT, R0                             ; 1409
                                 52      01    A0  9E 00244       MOVAB   1(R0), R_BEGIN                         ;
                                 53            50  D0 00248       MOVL    R0, L_LAST                             ; 1410
                                 54            50  D0 0024B       MOVL    R0, L_END                              ; 1411
                                 01            52  D1 0024E  40$: CMPL    R_BEGIN, #1                            ; 1427
                                               11  12 00251       BNEQ    41$                                    ;
                                 50 00000000G  EF  D0 00253       MOVL    LCOUNT, R0                             ; 1435
                                 52      01    A0  9E 0025A       MOVAB   1(R0), R_BEGIN                         ;
                                 54            50  D0 0025E       MOVL    R0, L_END                              ; 1436
                                 53            50  D0 00261       MOVL    R0, L_LAST                             ; 1437
                      00000000G  EF            54  D0 00264  41$: MOVL    L_END, LCOUNT                          ; 1440
                                 53            52  D1 0026B       CMPL    R_BEGIN, L_LAST                        ; 1442
                                               35  14 0026E       BGTR    44$                                    ;
                                               52  D7 00270       DECL    I                                      ; 1444
```

```
                                   2D  11 00272         BRB     43$                               :
                                   6B  D6 00274 42$:    INCL    TCOUNT                            : 1449
                      50           6B  D0 00276         MOVL    TCOUNT, R0                        : 1450
         00000000GEF40           6A42  D0 00279         MOVL    LTYPE[I], TTYPE[R0]               :
                      00000000G    EF  9F 00282         PUSHAB  STR$FAILURE                       : 1452
                                   7E  D4 00288         CLRL    -(SP)                             :
                      00000000GEF40    7F 0028A         PUSHAQ  TLINES[R0]                        :
                      00000000GEF42    7F 00291         PUSHAQ  LLINES[I]                         :
                                   7E  D4 00298         CLRL    -(SP)                             :
         00000000G   EF            05  FB 0029A         CALLS   #5, XST$COPY                      :
      CF             52            53  F3 002A1 43$:    AOBLEQ  L_LAST, I, 42$                    : 1444
                                   55  D5 002A5 44$:    TSTL    R_TOTAL                           : 1455
                                   3B  13 002A7         BEQL    47$                               :
                     52       FF   A6  9E 002A9         MOVAB   -1(R6), I                         : 1457
                                   31  11 002AD         BRB     46$                               :
                                   6B  D6 002AF 45$:    INCL    TCOUNT                            : 1462
                     50            6B  D0 002B1         MOVL    TCOUNT, R0                        : 1463
         00000000GEF40 00000000GEF42   D0 002B4         MOVL    RTYPE[I], TTYPE[R0]               :
                       00000000G   EF  9F 002C1         PUSHAB  STR$FAILURE                       : 1465
                                   7E  D4 002C7         CLRL    -(SP)                             :
                       00000000GEF40   7F 002C9         PUSHAQ  TLINES[R0]                        :
                       00000000GEF42   7F 002D0         PUSHAQ  RLINES[I]                         :
                                   7E  D4 002D7         CLRL    -(SP)                             :
         00000000G   EF            05  FB 002D9         CALLS   #5, XST$COPY                      :
      CB             52            57  F3 002E0 46$:    AOBLEQ  R_LAST, I, 45$                    : 1457
                     54 00000000G  EF  D0 002E4 47$:    MOVL    LCOUNT, R4                        : 1468
                     52            6B  D0 002EB         MOVL    TCOUNT, R2                        :
                     52            54  D1 002EE         CMPL    R4, R2                            :
                                   2E  18 002F1         BGEQ    50$                               :
                     53            54  D0 002F3         MOVL    R4, I                             : 1474
                                   1C  11 002F6         BRB     49$                               :
                  6A43             02  D0 002F8 48$:    MOVL    #2, LTYPE[I]                      : 1476
                       00000000GEF43   7F 002FC         PUSHAQ  LLINES[I]                         : 1477
                                   04  DD 00303         PUSHL   #4                                :
                       00000000'   EF  DD 00305         PUSHL   BLANKS                            :
                                   04  DD 0030B         PUSHL   #4                                :
         00000000G   EF            04  FB 0030D         CALLS   #4, PADLIN                        :
      E0             53            52  F3 00314 49$:    AOBLEQ  R2, I, 48$                        : 1474
         00000000G   EF            6B  D0 00318         MOVL    TCOUNT, LCOUNT                    : 1480
                                   32  11 0031F         BRB     53$                               : 1468
                     54            52  D1 00321 50$:    CMPL    R2, R4                            : 1483
                                   2D  18 00324         BGEQ    53$                               :
                                   20  11 00326         BRB     52$                               : 1489
         00000000GEF42             02  D0 00328 51$:    MOVL    #2, TTYPE[I]                      : 1491
                       00000000GEF42   7F 00330         PUSHAQ  TLINES[I]                         : 1492
                                   04  DD 00337         PUSHL   #4                                :
                       00000000'   EF  DD 00339         PUSHL   BLANKS                            :
                                   04  DD 0033F         PUSHL   #4                                :
         00000000G   EF            04  FB 00341         CALLS   #4, PADLIN                        :
      DC             52            54  F3 00348 52$:    AOBLEQ  R4, I, 51$                        : 1489
                     6B 00000000G  EF  D0 0034C         MOVL    LCOUNT, TCOUNT                    : 1495
         00000000G   EF 00000000G  EF  D0 00353 53$:    MOVL    LCOUNT, ALLOWD                    : 1500
                                   01  DD 0035E         PUSHL   #1                                : 1501
             F77E    CF            01  FB 00360         CALLS   #1, PUTPAG                        :
                     01 00000000G  EF  B1 00365         CMPW    CMDBLK+4, #1                      : 1503
                                   27  12 0036C         BNEQ    54$                               :
         00000000G   EF 00000000'  EF  9E 0036E         MOVAB   $IOB$OUTPUT, IOB$+68              :
```

```
                    C0000000G  EF             07  90 00379          MOVB    #7, IOB$+44                               ;
                               00000000G  EF  9F 00380          PUSHAB  XPO$FAILURE                               ;
                                          7E  D4 00386          CLRL    -(SP)                                     ;
                               00000000G  EF  9F 00388          PUSHAB  IOB$                                      ;
                    00000000G  EF             03  FB 0038E          CALLS   #3, XPO$PUT                               ;
                                              04 00395 54$:     RET                                               ; 1504

; Routine Size:  918 bytes,    Routine Base:  $CODE$ + 051D
```

```
%SBTTL 'LAST_CONT -- Generate continuation heading for last page'
ROUTINE LAST_CONT (L_NUM) =
!++
!
! FUNCTIONAL DESCRIPTION:
!
!       This routine generates a continuation heading for the last page
!       in the temp column.
!
!       It first builds a stack of line numbers which have the index
!       levels (current - 1), (current - 2), ..., 0 as these lines are
!       the predecessors of the current line.
!
!       It then generates the continuation heading.
!
!       For each index level, the line on the stack is copied.
!
!       If the line is not a continuation head, tne line is copied
!       up to either the NULL inserted by INS_LINE to delimit the
!       start of page references or the whole string is copied.
!
!       If the line is a continuation head, it is copied up to the
!       string '(Cont.)' or the whole string is copied.
!
!       The following lines will also be copied if the delimiting
!       string was not found and they are wrap lines (i.e., they
!       have an indent level equal to the current level + 2)
!
! FORMAL PARAMETERS:
!
!       L_NUM   - Index into left column pointing to line which
!                 will be the first in the new right column (i.e.,
!                 the line for which the continuation heading is generated.
!
! IMPLICIT INPUTS:
!
!       LTYPE   - vector of line types for left column
!       LLINES  - left column lines
!
! IMPLICIT OUTPUTS:
!
!       TTYPE   - vector of line types for temp column
!       TCOUNT  - number of lines in temp column
!       TLINES - temp column lines
!
! ROUTINE VALUE:
! COMPLETION CODES:
!
!       Returns TRUE if it was possible to generate a continuation heading
!       Returns FALSE if it was impossible to generate a continuation heading
!
! SIDE EFFECTS:
!
!       None
!--
    BEGIN
    LOCAL
```

```
  2             CONT,
  2             CONT_INT_LEN,
  2             CONT_EXT_LEN,
  2             MAX_EXT,
  2             CUR_INDENT,
  2             CUR_LINE,
  2             INDENT;
  2
  2         INDENT = INDENT_LEVEL (LLINES [.L_NUM, 0,0,0,0]) - 1;
  2
  2         IF .INDENT LSS 0
  2         THEN
  3             BEGIN
  3             !
  3             ! The input line was blank.
  3             ! This can happen if the column width was too narrow to fit
  3             ! the entry into. Usually this is caused by a combination of
  3             ! a narrow column width and a deep subindex level.
  3             !
L 3 %IF %BLISS (BLISS32)
  3 %THEN
  3
  3             SIGNAL (INDEX$_LASTCONT);
  3
U 3 %ELSE
U 3
U 3             $XPO_PUT_MSG (SEVERITY = WARNING,
U 3                 STRING = 'can''t generate continuation heading on last page');
U 3
  3 %FI
  3             RETURN FALSE;
  2             END;
  2
  2         CONT = CH$PTR (UPLIT ('(Cont_.)'));
  2         CONT_INT_LEN = 8;                           ! Internal length of '(Cont_.)'
  2
  2         IF .CMDBLK [NDX$H_FORMAT] NEQ DSR
  2         THEN
  3             BEGIN
  3             !
  3             ! TMS11 output
  3             ! Compute length of '(Cont.)' in TMS units
  3             ! Set maximum line length in TMS units
  3             !
  3             LOCAL
  3                 PTR;
  3
  3             CONT_EXT_LEN = 0;
  3             PTR = .CONT;
  3
  3             INCR I FROM 1 TO .CONT_INT_LEN DO
  4                 BEGIN
  4
  4                 LOCAL
  4                     CH;
  4
  4                 CH = CH$RCHAR_A (PTR);
  4
  4                     IF .CH NEQ %C'_' THEN CONT_EXT_LEN = .CONT_EXT_LEN + .CHRSIZ [.CH];
  3                     END;
  3
  3                 MAX_EXT = .CMDBLK [NDX$G_COLUMN_WID] * TMSSTD;
  3                 END
  2             ELSE
  3                 BEGIN
  3                 !
  3                 ! RUNOFF output
  3                 ! Length of '(Cont.)' and maximum line length are in characters
  3                 !
  3                 CONT_EXT_LEN = 7;
  3                 MAX_EXT = .CMDBLK [NDX$G_COLUMN_WID];
  2                 END;
  2
  2             CUR_INDENT = .INDENT;
  2             CUR_LINE = .L_NUM - 1;
  2
  2             WHILE (.CUR_INDENT GEQ 0) AND (.CUR_LINE GTR 0) DO
  3                 BEGIN
  3                 !
  3                 ! Build a stack of entry lines at the correct indent level
  3                 !
  4                 IF (.CUR_INDENT EQL INDENT_LEVEL (LLINES [.CUR_LINE, 0,0,0,0]))
  4                 AND (.LTYPE [.CUR_LINE] GEQ ENTRY_B)
  3                 THEN
  4                     BEGIN
  4                     !
  4                     ! Found preceeding subentry
  4                     !
  4                     LSTSTK [.CUR_INDENT] = .CUR_LINE;
  4                     CUR_INDENT = .CUR_INDENT - 1;
  3                     END;
  3
  3                 CUR_LINE = .CUR_LINE - 1;
  2                 END;
  2
  3             IF (.CUR_LINE EQL 0) AND (.CUR_INDENT GEQ 0)
  2             THEN
  3                 BEG N
  3                 !
  3                 ! An internal inconsistancy prevented finding the predecessors
  3                 ! of the current line. This error is non-fatal: the last page
  3                 ! will be output as is.
  3                 !
  3
L 3 %IF %BLISS (BLISS32)
  3 %THEN                                                   ! Signal errors for BLISS32
  3
  3                 SIGNAL (INDEX$_LASTCONT, 0, INDEX$_BADLOGIC);
  3
U 3 %ELSE                                                   ! Use $XPO_PUT_MSG otherwise
U 3
U 3                 $XPO_PUT_MSG (SEVERITY = WARNING,
U 3                     STRING = 'internal error - cannot generate continuation heading on last page');
U 3
```

```
3 %FI
3         RETURN FALSE;
2         END;
2
2     INCR I FROM 0 TO .INDENT DO
3         BEGIN
3         LOCAL
3             S : REF $STR_DESCRIPTOR (),
3             LINE_NO;
3
3         LINE_NO = .LSTSTK [.I];
3
3         WHILE .LINE_NO NEQ 0 DO
4             BEGIN
4             LOCAL
4                 PTR,
4                 LEN;
4
4             S = LLINES [.LINE_NO, 0,0,0,0];
4
4             IF .LTYPE [.LINE_NO] EQL CONT_HEAD
4             THEN
4                 !
4                 ! Picked up a continuation heading
4                 !
4                 PTR = CH$FIND_SUB (.S [STR$H_LENGTH], .S [STR$A_POINTER], 8, .CONT)
4             ELSE
4                 !
4                 ! Line not a continuation head
4                 !
4                 PTR = CH$FIND_CH (.S [STR$H_LENGTH], .S [STR$A_POINTER], 0);
4
4             IF CH$FAIL (.PTR)
4             THEN
5                 BEGIN
5                 !
5                 ! Delimiter not found
5                 !
5                 LEN = .S [STR$H_LENGTH];
5                 IF INDENT_LEVEL (LLINES [.LINE_NO + 1, 0,0,0,0]) EQL .I + 2
5                 THEN
5                     !
5                     ! Line is wrapped to next
5                     !
5                     LINE_NO = .LINE_NO + 1
5                 ELSE
5                     !
5                     ! Line not wrapped to next
5                     !
5                     LINE_NO = 0;
5                 END
4             ELSE
5                 BEGIN
5                 !
5                 ! Delimiter found
5                 !
5                 LEN = CH$DIFF (.PTR, .S [STR$A_POINTER]);
```

```
5
5                 !
5                 ! Signal no more lines at this level
5                 !
5                 LINE_NO = 0;
4                 END;
4
4             !
4             ! Point to beginning of string
4             !
4             PTR = .S [STR$A_POINTER];
4
4             IF CH$NEQ (1, .BLANKS, .LEN, .PTR, %C' ')
4             THEN
5                 BEGIN
5                 !
5                 ! Line is non-blank.
5                 !
5                 DECR I FROM .LEN - 1 TO 0 DO
5                     IF CH$RCHAR (CH$PLUS (.PTR, .I)) NEQ %C' '
5                     THEN
5                         EXITLOOP
5                     ELSE
5                         !
5                         ! Remove a trailing blank
5                         !
5                         LEN = .LEN - 1;
5
5                 !
5                 ! Bump line count, set line type and copy line
5                 !
5                 TCOUNT = .TCOUNT + 1;
5                 $STR_COPY (STRING = (.LEN, .PTR), TARGET = TLINES [.TCOUNT, 0,0,0,0]);
5                 TTYPE [.TCOUNT] = CONT_HEAD;
4                 END;
3             END;
2         END;
2
2
2     IF (GET_EXT_LEN (TLINES [.TCOUNT, 0,0,0,0]) + .CONT_EXT_LEN) GEQ .MAX_EXT
2     THEN
3         BEGIN
3         !
3         ! '(Cont.)' doesn't fit on current line.
3         ! See if it will fit on next line.
3         !
4         IF  (
5             (.CMDBLK [NDX$H_FORMAT] NEQ DSR)
5             AND ((.INDENT + 2) * MSPACE + .CONT_EXT_LEN GEQ .MAX_EXT)
4             )
4         OR  (
5             (.CMDBLK [NDX$H_FORMAT] EQL DSR)
5             AND ((.INDENT + 2) * 2 + .CONT_EXT_LEN GEQ .MAX_EXT)
4             )
3         THEN
4             BEGIN
4             !
```

```
         1790  4                ! Can't put '(Cont.)' on new line - hence can't generate a
         1791  4                ! continuation heading.
         1792  4                !
       L 1793  4 %IF %BLISS (BLISS32)
         1794  4 %THEN                                          ! Signal errors in BLISS32
         1795  4
         1796  4                SIGNAL (INDEX$_LASTCONT, 0, INDEX$_DOESNTFIT, 2, .CONT_INT_LEN, .CONT);
         1797  4
       U 1798  4 %ELSE                                          ! Use $XPO_PUT_MSG otherwise
       U 1799  4
       U 1800  4                $XPO_PUT_MSG (SEVERITY = WARNING,
       U 1801  4                    STRING = 'can''t generate continuation heading on last page');
       U 1802  4
         1803  4 %FI
         1804  4                RETURN FALSE;
         1805  4                END
         1806  3            ELSE
         1807  4                BEGIN
         1808  4                !
         1809  4                ! Indent a new line and append '(Cont.)' to it.
         1810  4                !
         1811  4                LOCAL
         1812  4                    LINE : VECTOR [CH$ALLOCATION (200)],
         1813  4                    PTR,
         1814  4                    LEN;
         1815  4
         1816  4                PTR = CH$PTR (LINE);
         1817  4                LEN = (.INDENT + 2) * 2 - 1;
         1818  4                CH$FILL (%C' ', .LEN, .PTR);
         1819  4
         1820  4                TCOUNT = .TCOUNT + 1;
         1821  4                $STR_COPY (STRING = (.LEN, .PTR), TARGET = TLINES [.TCOUNT, 0,0,0,0]);
         1822  4                TTYPE [.TCOUNT] = CONT_HEAD;
         1823  3                END;
         1824  2            END;
         1825  2
       P 1826  2        $STR_APPEND (STRING = $STR_CONCAT ((1, .BLANKS), (.CONT_INT_LEN, .CONT)),
         1827  2            TARGET = TLINES [.TCOUNT, 0,0,0,0]);
         1828  2
         1829  2        RETURN TRUE;
         1830  1        END;


                                                                   .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                          0005F    .BLKB   1
                  29  2E  5F  74  6E  6F  43  28   00060 P.ABA:    .ASCII  \(Cont_.)\                              ;


                                                                   .PSECT  $CODE$,NOWRT,2

                                                     0FFC 00000 LAST_CONT:
                                                                   .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11    ; 1506
                        5E      FF24    CE  9E 00002               MOVAB   -220(SP), SP                            ;
                        52        04    AC  D0 00007               MOVL    L_NUM, R2                               ; 1570
                        00000000GEF42  7F 0000B          PUSHAQ  LLINES[R2]                          ;
         00000000V  EF            01  FB 00012          CALLS   #1, INDENT_LEVEL                    ;
                    5A        FF  A0  9E 00019          MOVAB   -1(R0), INDENT                      ;
                                  10  18 0001D          BGEQ    1$                                  ; 1572
                        00000000G  8F  DD 0001F          PUSHL   #DSRINDEX$_LASTCONT                 ; 1584
         00000000G  00            01  FB 00025          CALLS   #1, LIB$SIGNAL                      ;
                                02A5  31 0002C          BRW     28$                                 ; 1592
              04    AE 00000000'  EF  9E 0002F 1$:      MOVAB   P.ABA, CONT                         ; 1595
              08    AE            08  D0 00037          MOVL    #8, CONT_INT_LEN                    ; 1596
                    01 00000000G  EF  B1 0003B          CMPW    CMDBLK+4, #1                        ; 1598
                                  31  13 00042          BEQL    4$                                  ;
                                  5B  D4 00044          CLRL    CONT_EXT_LEN                        ; 1609
                    53        04  AE  D0 00046          MOVL    CONT, PTR                           ; 1610
                                  51  D4 0004A          CLRL    I                                   ; 1612
                                  14  11 0004C          BRB     3$                                  ;
                    50            83  9A 0004E 2$:      MOVZBL  (PTR)+, CH                          ; 1618
         0000005F   8F            50  D1 00051          CMPL    CH, #95                             ; 1620
                                  08  13 00058          BEQL    3$                                  ;
                    5B 00000000GFF40  C0 0005A          ADDL2   @CHRSIZ[CH], CONT_EXT_LEN           ;
     E7             51        08  AE  F3 00062 3$:      AOBLEQ  CONT_INT_LEN, I, 2$                 ; 1612
     6E  00000000G  EF 00000000G  8F  C5 00067          MULL3   #TMSSTD, CMDBLK+12, MAX_EXT         ; 1623
                                  0A  11 00073          BRB     5$                                  ; 1598
                    5B            07  D0 00075 4$:      MOVL    #7, CONT_EXT_LEN                    ; 1631
                    6E 00000000G  EF  D0 00078          MOVL    CMDBLK+12, MAX_EXT                  ; 1632
                    53            5A  D0 0007F 5$:      MOVL    INDENT, CUR_INDENT                  ; 1635
                                  52  D7 00082 6$:      DECL    CUR_LINE                            ; 1636
                                  53  D5 00084          TSTL    CUR_INDENT                          ; 1638
                                  2D  19 00086          BLSS    7$                                  ;
                                  52  D5 00088          TSTL    CUR_LINE                            ;
                                  29  15 0008A          BLEQ    7$                                  ;
                        00000000GEF42  7F 0008C          PUSHAQ  LLINES[CUR_LINE]                    ; 1643
         00000000V  EF            01  FB 00093          CALLS   #1, INDENT_LEVEL                    ;
                    50            53  D1 0009A          CMPL    CUR_INDENT, R0                      ;
                                  E3  12 0009D          BNEQ    6$                                  ;
                    05 00000000GEF42  D1 0009F          CMPL    LTYPE[CUR_LINE], #5                 ; 1644
                                  D9  19 000A7          BLSS    6$                                  ;
         00000000GEF43            52  D0 000A9          MOVL    CUR_LINE, LSTSTK[CUR_INDENT]        ; 1650
                                  53  D7 000B1          DECL    CUR_INDENT                          ; 1651
                                  CD  11 000B3          BRB     6$                                  ; 1654
                                  52  D5 000B5 7$:      TSTL    CUR_LINE                            ; 1657
                                  1C  12 000B7          BNEQ    8$                                  ;
                                  53  D5 000B9          TSTL    CUR_INDENT                          ;
                                  18  19 000BB          BLSS    8$                                  ;
                        00000000G  8F  DD 000BD          PUSHL   #DSRINDEX$_BADLOGIC                 ; 1669
                                  7E  D4 000C3          CLRL    -(SP)                               ;
                        00000000G  8F  DD 000C5          PUSHL   #DSRINDEX$_LASTCONT                 ;
         00000000G  00            03  FB 000CB          CALLS   #3, LIB$SIGNAL                      ;
                                01FF  31 000D2          BRW     28$                                 ; 1677
                    54            01  CE 000D5 8$:      MNEGL   #1, I                               ; 1680
                                00D3  31 000D8 9$:      BRW     22$                                 ;
                    56 00000000GEF44  D0 000DB 10$:     MOVL    LSTSTK[I], LINE_NO                  ; 1686
                                  56  D5 000E3 11$:     TSTL    LINE_NO                             ; 1688
                                  F1  13 000E5          BEQL    9$                                  ;
                    58 00000000GEF46  7E 000E7          MOVAQ   LLINES[LINE_NO], S                  ; 1694
                    0B 00000000GEF46  D1 000EF          CMPL    LTYPE[LINE_NO], #11                 ; 1696
                                  17  12 000F7          BNEQ    13$                                 ;
                    55            68  3C 000F9          MOVZWL  (S), R5                             ; 1701
```

```
                                                                      D  6
NDXPAG            NDXPAG -- Output page formatting routines      16-Sep-1984 01:06:39    VAX-11 Bliss-32 V4.0-742                  Page  58
V04-000           LAST_CONT -- Generate continuation heading for  14-Sep-1984 13:07:15    [RUNOFF.SRC]NDXPAG.BLI;1                        (6)

                                        59       04   A8 D0 000FC              MOVL    4(S), R9                        ;
            69            55       04   BE            08 39 00100              MATCHC  #8, @CONT, R5, (R9)             ;
                                                      03 13 00106              BEQL    12$                             ;
                                        53            08 D0 00108              MOVL    #8, R3                          ;
                                        57            73 7E 0010B 12$:         MOVAQ   -(R3), PTR                      ;
                                                      12 11 0010E              BRB     15$                             ;
                                        55            68 3C 00110 13$:         MOVZWL  (S), R5                         ; 1706
                                        59       04   A8 D0 00113              MOVL    4(S), R9                        ;
                          69            55            00 3A 00117              LOCC    #0, R5, (R9)                    ;
                                                      02 12 0011B              BNEQ    14$                             ;
                                                      51 D4 0011D              CLRL    R1                              ;
                                        57            51 D0 0011F 14$:         MOVL    R1, PTR                         ;
                                                      1B 12 00122 15$:         BNEQ    16$                             ; 1708
                                           00000000GEF46 7F 00124              PUSHAQ  LLINES+8[LINE_NO]               ; 1715
                              00000000V EF            01 FB 0012B              CALLS   #1, INDENT_LEVEL                ;
                                        51       02   A4 9E 00132              MOVAB   2(R4), R1                       ;
                                        51            50 D1 00136              CMPL    R0, R1                          ;
                                                      08 12 00139              BNEQ    17$                             ;
                                                      56 D6 0013B              INCL    LINE_NO                         ; 1720
                                                      06 11 0013D              BRB     18$                             ;
                          55            57            59 C3 0013F 16$:         SUBL3   R9, PTR, LEN                    ; 1732
                                                      56 D4 00143 17$:         CLRL    LINE_NO                         ; 1737
                                        57            59 D0 00145 18$:         MOVL    R9, PTR                         ; 1743
            55            20 00000000'  FF            01 2D 00148              CMPC5   #1, @BLANKS, #32, LEN, (PTR)    ; 1745
                                                      67    00151                                                      ;
                                                      8F 13 00152              BEQL    11$                             ;
                                        50            55 D0 00154              MOVL    LEN, I                          ; 1751
                                                      08 11 00157              BRB     20$                             ;
                                        20          6047 91 00159 19$:         CMPB    (I)[PTR], #32                   ; 1752
                                                      05 12 0015D              BNEQ    21$                             ;
                                                      55 D7 0015F              DECL    LEN                             ; 1759
                                        F5            50 F4 00161 20$:         SOBGEQ  I, 19$                          ; 1752
                                           00000000G  EF D6 00164 21$:         INCL    TCOUNT                          ; 1764
                                        50 00000000G  EF D0 0016A              MOVL    TCOUNT, R0                      ; 1765
                                   F8   AD            55 B0 00171              MOVW    LEN, $STR$STRING                ;
                                   FA   AD            0E 90 00175              MOVB    #14, $STR$STRING+2              ;
                                   FB   AD            01 90 00179              MOVB    #1, $STR$STRING+3               ;
                                   FC   AD            57 D0 0017D              MOVL    PTR, $STR$STRING+4              ;
                                           00000000G  EF 9F 00181              PUSHAB  STR$FAILURE                     ;
                                                      7E D4 00187              CLRL    -(SP)                           ;
                                           00000000GEF40 7F 00189              PUSHAQ  TLINES[R0]                      ;
                                                 F8   AD 9F 00190              PUSHAB  $STR$STRING                     ;
                                                      7E D4 00193              CLRL    -(SP)                           ;
                              00000000G EF            05 FB 00195              CALLS   #5, XST$COPY                    ;
                                        50 00000000G  EF D0 0019C              MOVL    TCOUNT, R0                      ; 1766
                              00000000GEF40           0B D0 001A3              MOVL    #11, TTYPE[R0]                  ;
                                                    FF35 31 001AB              BRW     11$                             ; 1688
         FF27             54            01            5A F1 001AE 22$:         ACBL    INDENT, #1, I, 10$              ; 1680
                                        50 00000000G  EF D0 001B4              MOVL    TCOUNT, R0                      ; 1772
                                           00000000GEF40 7F 001BB              PUSHAQ  TLINES[R0]                      ;
                              00000000V EF            01 FB 001C2              CALLS   #1, GET_EXT_LEN                 ;
                                        50            5B C0 001C9              ADDL2   CONT_EXT_LEN, R0                ;
                                        6E            50 D1 001CC              CMPL    R0, MAX_EXT                     ;
                                                      03 18 001CF              BGEQ    23$                             ;
                                                    00A8 31 001D1              BRW     27$                             ;
                                        01 00000000G  EF B1 001D4 23$:         CMPW    CMDBLK+4, #1                    ; 1780
                                                      15 13 001DB              BEQL    24$                             ;
```

```
                  50              5A 00000000G  8F   C5 001DD          MULL3   #MSPACE, INDENT, R0                          : 1781
                                  50 000000C0*EB40   9E 001E5          MOVAB   <MSPACE*2>(CONT_EXT_LEN)[R0], R0             :
                                  6E            50   D1 001ED          CMPL    R0, MAX_EXT                                  :
                                                13   18 001F0          BGEQ    25$                                          :
                                  01 00000000G  EF   B1 001F2  24$:    CMPW    CMDBLK+4, #1                                 : 1784
                                                2A   12 001F9          BNEQ    26$                                          :
                                  50       04 AB4A   3E 001FB          MOVAW   4(CONT_EXT_LEN)[INDENT], R0                  : 1785
                                  6E            50   D1 00200          CMPL    R0, MAX_EXT                                  :
                                                20   19 00203          BLSS    26$                                          :
                                           04   AE   DD 00205  25$:    PUSHL   CONT                                         : 1796
                                           0C   AE   DD 00208          PUSHL   CONT_INT_LEN                                 :
                                                02   DD 0020B          PUSHL   #2                                           :
                                     00000000G  8F   DD 0020D          PUSHL   #DSRINDEX$_DOESNTFIT                         :
                                                7E   D4 00213          CLRL    -(SP)                                        :
                                     00000000G  8F   DD 00215          PUSHL   #DSRINDEX$_LASTCONT                          :
                        00000000G 00            06   FB 0021B          CALLS   #6, LIB$SIGNAL                               :
                                              00AF   31 00222          BRW     28$                                          : 1804
                                  56       14   AE   9E 00225  26$:    MOVAB   LINE, PTR                                    : 1816
                                  5A            02   C4 00229          MULL2   #2, LEN                                      : 1817
                                  5A            03   C0 0022C          ADDL2   #3, LEN                                      :
       5A         20              6E            00   2C 0022F          MOVC5   #0, (SP), #32, LEN, (PTR)                    : 1818
                                                66      00234                                                               :
                                     00000000G  EF   D6 00235          INCL    TCOUNT                                       : 1820
                                  50 00000000G  EF   D0 0023B          MOVL    TCOUNT, R0                                   : 1821
                           0C     AE            5A   B0 00242          MOVW    LEN, $STR$STRING                             :
                           0E     AE            0E   90 00246          MOVB    #14, $STR$STRING+2                           :
                           0F     AE            01   90 0024A          MOVB    #1, $STR$STRING+3                            :
                           10     AE            56   D0 0024E          MOVL    PTR, $STR$STRING+4                           :
                                     00000000G  EF   9F 00252          PUSHAB  STR$FAILURE                                  :
                                                7E   D4 00258          CLRL    -(SP)                                        :
                                     00000000GEF40   7F 0025A          PUSHAQ  TLINES[R0]                                   :
                                           18   AE   9F 00261          PUSHAB  $STR$STRING                                  :
                                                7E   D4 00264          CLRL    -(SP)                                        :
                        00000000G EF            05   FB 00266          CALLS   #5, XST$COPY                                 :
                                  50 00000000G  EF   D0 0026D          MOVL    TCOUNT, R0                                   : 1822
                        00000000GEF40           0B   D0 00274          MOVL    #11, TTYPE[R0]                               :
                           F8     AD            01   B0 0027C  27$:    MOVW    #1, $STR$STRING0                             : 1827
                           FA     AD            0E   90 00280          MOVB    #14, $STR$STRING0+2                          :
                           FB     AD            01   90 00284          MOVB    #1, $STR$STRING0+3                           :
                           FC     AD 00000000'  EF   D0 00288          MOVL    BLANKS, $STR$STRING0+4                       :
                           F0     AD       08   AE   B0 00290          MOVW    CONT_INT_LEN, $STR$STRING1                   :
                           F2     AD            0E   90 00295          MOVB    #14, $STR$STRING1+2                          :
                           F3     AD            01   90 00299          MOVB    #1, $STR$STRING1+3                           :
                           F4     AD       04   AE   D0 0029D          MOVL    CONT, $STR$STRING1+4                         :
                                           F0   AD   9F 002A2          PUSHAB  $STR$STRING1                                 :
                                           F8   AD   9F 002A5          PUSHAB  $STR$STRING0                                 :
                        00000000G EF            02   FB 002A8          CALLS   #2, XST$JOIN                                 :
                                     00000000G  EF   9F 002AF          PUSHAB  STR$FAILURE                                  :
                                                7E   D4 002B5          CLRL    -(SP)                                        :
                                  51 00000000G  EF   D0 002B7          MOVL    TCOUNT, R1                                   :
                                     00000000GEF41   7F 002BE          PUSHAQ  TLINES[R1]                                   :
                                                50   DD 002C5          PUSHL   R0                                           :
                                                7E   D4 002C7          CLRL    -(SP)                                        :
                        00000000G EF            05   FB 002C9          CALLS   #5, XST$APPEND                               :
                                  50            01   D0 002D0          MOVL    #1, R0                                       : 1829
                                                     04 002D3          RET                                                  :
                                                50   D4 002D4  28$:    CLRL    R0                                           : 1830
```

```
                                                   04 002D6         RET                                                      ;

; Routine Size:  727 bytes,    Routine Base:  $CODE$ + 08B3
```

```
1 %SBTTL 'GET_EXT_LEN - Get external length of line'
1 ROUTINE GET_EXT_LEN (DSC) =
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine returns the external length of a line in either
1 !       number of characters or in TMS relative units
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       DSC     - Address of string descriptor of line
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       CHRSIZ  - TMS character size vector
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       None
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Returns external length of line
1 !
1 ! SIDE EFFECTS:
1 !
1 !       None
1 !--
2     BEGIN
2
2     MAP
2         DSC : REF $STR_DESCRIPTOR ();
2
2     LOCAL
2         PTR,
2         EXT_LEN;
2
2     PTR = .DSC [STR$A_POINTER];
2     EXT_LEN = 0;
2
2     INCR I FROM 1 TO .DSC [STR$H_LENGTH] DO
3         BEGIN
3         LOCAL
3             CH;
3
3         CH = CH$RCHAR_A (PTR);
3
3         SELECTONE .CH OF
3         SET
3
3         [%C'*']:
3             !
3             ! Bold sequence. Doesn't add to external length.
3             !
3             ;
```

```
; 1412      1888  3
; 1413      1889  3
; 1414      1890  3             [%C'&']:
; 1415      1891  3                 !
; 1416      1892  3                 ! Underline sequence. Doesn't add to external length
; 1417      1893  3                 ;
; 1418      1894  3
; 1419      1895  3             [%C'%']:
; 1420      1896  4                 BEGIN
; 1421      1897  4                 !
; 1422      1898  4                 ! Overstrike sequence.
; 1423      1899  4                 ! Next character doesn't count either.
; 1424      1900  4                 !
; 1425      1901  4                 CH$RCHAR_A (PTR);
; 1426      1902  4                 I = .I + 1;
; 1427      1903  3                 END;
; 1428      1904  3
; 1429      1905  3             [OTHERWISE]:
; 1430      1906  4                 BEGIN
; 1431      1907  4                 !
; 1432      1908  4                 ! An ordinary character.
; 1433      1909  4                 !
; 1434      1910  4                 IF .CH EQL %C'_'
; 1435      1911  4                 THEN
; 1436      1912  5                     BEGIN
; 1437      1913  5                     !
; 1438      1914  5                     ! Quote sequence doesn't count
; 1439      1915  5                     !
; 1440      1916  5                     CH$RCHAR_A (PTR);
; 1441      1917  5                     I = .I + 1;
; 1442      1918  4                     END;
; 1443      1919  4
; 1444      1920  4                 IF .CMDBLK [NDX$H_FORMAT] NEQ DSR
; 1445      1921  4                 THEN
; 1446      1922  4                     !
; 1447      1923  4                     ! For TMS, use relative character size table
; 1448      1924  4                     !
; 1449      1925  4                     EXT_LEN = .EXT_LEN + .CHRSIZ [.CH]
; 1450      1926  4                 ELSE
; 1451      1927  4                     !
; 1452      1928  4                     ! For RUNOFF, just count a character
; 1453      1929  4                     !
; 1454      1930  4                     EXT_LEN = .EXT_LEN + 1;
; 1455      1931  3                 END;
; 1456      1932  3
; 1457      1933  3             TES;
; 1458      1934  2             END;
; 1459      1935  2
; 1460      1936  2         RETURN .EXT_LEN;
; 1461      1937  1         END;



                                                  001C 00000 GET_EXT_LEN:
                                                                     .WORD   Save R2,R3,R4                          ; 1832
```

```
                    50        04   AC  D0 00002          MOVL    DSC, R0                               ; 1870
                    54        C4   A0  D0 00006          MOVL    4(R0), PTR                            ;
                    53             60  3C 0000A          MOVZWL  (R0), R3                              ; 1873
                                   51  7C 0000D          CLRQ    I                                     ;
                                   3A  11 0000F          BRB     5$                                    ;
                    50             84  9A 00011 1$:      MOVZBL  (PTR)+, CH                            ; 1878
                    2A             50  D1 00014          CMPL    CH, #42                               ; 1883
                                   32  13 00017          BEQL    5$                                    ;
                    26             50  D1 00019          CMPL    CH, #38                               ; 1890
                                   2D  13 0001C          BEQL    5$                                    ;
                    25             50  D1 0001E          CMPL    CH, #37                               ; 1895
                                   06  12 00021          BNEQ    2$                                    ;
                                   54  D6 00023          INCL    PTR                                   ; 1901
                                   51  D6 00025          INCL    I                                     ; 1902
                                   22  11 00027          BRB     5$                                    ; 1880
        0000005F    8F             50  D1 00029 2$:      CMPL    CH, #95                               ; 1910
                                   04  12 00030          BNEQ    3$                                    ;
                                   54  D6 00032          INCL    PTR                                   ; 1916
                                   51  D6 00034          INCL    I                                     ; 1917
                    01 00000000G   EF  B1 00036 3$:      CMPW    CMDBLK+4, #1                          ; 1920
                                   0A  13 0003D          BEQL    4$                                    ;
                    52 00000000GFF 40  C0 0003F          ADDL2   @CHRSIZ[CH], EXT_LEN                  ; 1925
                                   02  11 00047          BRB     5$                                    ;
                                   52  D6 00049 4$:      INCL    EXT_LEN                               ; 1930
               C2   51             53  F3 0004B 5$:      AOBLEQ  R3, I, 1$                             ; 1873
                    50             52  D0 0004F          MOVL    EXT_LEN, R0                           ; 1936
                                   04     00052          RET                                           ; 1937

; Routine Size:  83 bytes,    Routine Base:  $CODE$ + 0B8A


; 1462          1938  1
```

```
1 %SBTTL 'INDENT_LEVEL - Get indent level of string'
1 ROUTINE INDENT_LEVEL (DSC) =
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine computes the indent level of a line
1 !
1 !       The indent level is equal to 1/2 the number of
1 !       leading spaces on the line.
1 !
1 !       If the line is blank or begins with a tab, the indent level is -1.
1 !
1 !       The number of leading spaces is found by taking the
1 !       difference of a pointer to the first non-blank character
1 !       and a pointer to the beginning of the line.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       DSC - Address of string descriptor describing the line
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       None
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       None
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Returns the indent level of the input line
1 !       Returns -1 if the line is blank or if the line begins with a tab.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       None
1 !--
2     BEGIN
2     MAP
2       DSC : REF $STR_DESCRIPTOR ();
2
2     LOCAL
2       LEN,
2       PTR;
2
2     LEN = .DSC [STR$H_LENGTH];
2     PTR = .DSC [STR$A_POINTER];
2
2     IF CH$EQL (1, CH$PTR (UPLIT (' ')), .LEN, .PTR, %C' ')
2     THEN
2       !
2       ! Blank line.
2       !
2       RETURN -1
2     ELSE
```

```
; 1521          1996  2         !
; 1522          1997  2         ! Non-blank line
; 1523          1998  2         !
; 1524          1999  2         IF CH$EQL (1, CH$PTR (UPLIT ('  ')), 1, .PTR)
; 1525          2000  2         THEN
; 1526          2001  2             !
; 1527          2002  2             ! Line begins with a tab
; 1528          2003  2             !
; 1529          2004  2             RETURN -1
; 1530          2005  2         ELSE
; 1531          2006  2             !
; 1532          2007  2             ! Compute indent level
; 1533          2008  2             !
; 1534          2009  2             RETURN CH$DIFF (CH$FIND_NOT_CH (.LEN, .PTR, %C' '), .PTR) / 2;
; 1535          2010  2
; 1536          2011  1     END;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                    00  00  00  20  00068 P.ABB:  .ASCII  \ \<0><0><0>                             ;
                                    00  00  00  09  0006C P.ABC:  .ASCII  <9><0><0><0>                              ;


                                                                .PSECT  $CODE$,NOWRT,2

                                                003C 00000 INDENT_LEVEL:
                                                                .WORD   Save R2,R3,R4,R5                          ; 1940
                                 50       04  AC  D0 00002      MOVL    DSC, R0                                   ; 1986
                                 55           60  3C 00006      MOVZWL  (R0), LEN                                 ;
                                 54       04  A0  D0 00009      MOVL    4(R0), PTR                                ; 1987
         55          20 00000000'  EF          01  2D 0000D      CMPC5   #1, P.ABB, #32, LEN, (PTR)                ; 1989
                                               64     00016                                                       ;
                                               09  13 00017      BEQL    1$                                        ;
                                 64 00000000'  EF  91 00019      CMPB    P.ABC, (PTR)                              ; 1999
                                               05  12 00020      BNEQ    2$                                        ;
                                 51           01  CE 00022 1$:  MNEGL   #1, R1                                    ; 2004
                                               0E  11 00025      BRB     4$                                        ;
                     64          55           20  3B 00027 2$:  SKPC    #32, LEN, (PTR)                           ; 2009
                                               02  12 0002B      BNEQ    3$                                        ;
                                               51  D4 0002D      CLRL    R1                                        ;
                                 51           54  C2 0002F 3$:  SUBL2   PTR, R1                                   ;
                                 51           02  C6 00032      DIVL2   #2, R1                                    ;
                                 50           51  D0 00035 4$:  MOVL    R1, R0                                    ; 1999
                                               04     00038      RET                                               ; 2011

; Routine Size:  57 bytes,    Routine Base:  $CODE$ + 0BDD
```

```
1 %SBTTL 'GUIDE_HEAD -- Build a guide head for TMS11 or TEX'
1 ROUTINE GUIDE_HEAD (DSC) : NOVALUE =
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine inserts the format strings into a guide heading
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       DSC - Address of guide head string
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       None
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       None
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       None
1 !
1 ! SIDE EFFECTS:
1 !
1 !       None
1 !--
2     BEGIN
2
2     IF .CMDBLK [NDX$H_FORMAT] NEQ TEX
2     THEN
2         !
2         ! TMS11 output
2         !
3         $STR_COPY (STRING = $STR_CONCAT (TMS_GUIDE, .DSC, TMS_TXT_FMT), TARGET = TMS_TMP)
2     ELSE
2         !
2         ! TEX output
2         !
2         $STR_COPY (STRING = $STR_CONCAT ('{\gh ', .DSC, '}'), TARGET = TMS_TMP);
2
2     $STR_COPY (STRING = TMS_TMP, TARGET = .DSC);
1     END;


                                                     .PSECT  $PLIT$,NOWRT,NOEXE,2

                    20  68  67  5C  7B  00070 P.ABF:  .ASCII  \{\<92>\gh \                        :
                                    7D  00075 P.ABG:  .ASCII  \}\                                 :

                                                     .PSECT  $OWN$,NOEXE,2

                                0005  00094 $STR$STRING0:
                                                     .WORD   5                                    :
```

```
                                  01  0E  00096           .BYTE   14, 1                                ;
                               00000000'  00098           .ADDRESS P.ABF                               ;
                                    0001  0009C $STR$STRING2:
                                                          .WORD   1                                    ;
                                  01  0E  0009E           .BYTE   14, 1                                ;
                               00000000'  000A0           .ADDRESS P.ABG                               ;

                                                $STR$STRING0=       TMS_GUIDE
                                                $STR$STRING2=       TMS_TXT_FMT
                                                $STR$TARGET=        TMS_TMP
                                                $STR$TARGET=        TMS_TMP
                                                $STR$STRING=        TMS_TMP

                                                          .PSECT  $CODE$,NOWRT,2

                                    001C  00000 GUIDE_HEAD:
                                                          .WORD   Save R2,R3,R4                        ; 2013
                54 00000000G  EF   9E  00002          MOVAB   XST$COPY, R4                         ;
                53 00000000G  EF   9E  00009          MOVAB   STR$FAILURE, R3                      ;
                52 00000000'  EF   9E  00010          MOVAB   $STR$TARGET, R2                      ;
                04 00000000G  EF   B1  00017          CMPW    CMDBLK+4, #4                         ; 2043
                              0B   13  0001E          BEQL    1$                                   ;
                      28      A2   9F  00020          PUSHAB  $STR$STRING2                         ; 2048
                      04      AC   DD  00023          PUSHL   DSC                                  ;
                      10      A2   9F  00026          PUSHAB  $STR$STRING0                         ;
                              0B   11  00029          BRB     2$                                   ;
                    0098      C2   9F  0002B 1$:      PUSHAB  $STR$STRING2                         ; 2053
                      04      AC   DD  0002F          PUSHL   DSC                                  ;
                    0090      C2   9F  00032          PUSHAB  $STR$STRING0                         ;
        00000000G  EF              03   FB  00036 2$:      CALLS   #3, XST$JOIN                         ;
                              53   DD  0003D          PUSHL   R3                                   ;
                              7E   D4  0003F          CLRL    -(SP)                                ;
                              05   BB  00041          PUSHR   #^M<R0,R2>                           ;
                              7E   D4  00043          CLRL    -(SP)                                ;
                64            05   FB  00045          CALLS   #5, XST$COPY                         ;
                              53   DD  00048          PUSHL   R3                                   ; 2055
                              7E   D4  0004A          CLRL    -(SP)                                ;
                      04      AC   DD  0004C          PUSHL   DSC                                  ;
                              52   DD  0004F          PUSHL   R2                                   ;
                              7E   D4  00051          CLRL    -(SP)                                ;
                64            05   FB  00053          CALLS   #5, XST$COPY                         ;
                                   04  00056          RET                                          ; 2056

; Routine Size:  87 bytes,    Routine Base:  $CODE$ + 0C16
```

```
%SBTTL 'TMSINI -- Generate TMS11 top of file string'
GLOBAL ROUTINE TMSINI : NOVALUE =
!++
!
! FUNCTIONAL DESCRIPTION:
!
!       This routine generates and outputs the top of file sequence for
!       TMS11 output files
!
! FORMAL PARAMETERS:
!
!       None
!
! IMPLICIT INPUTS:
!
!       CMDBLK          - Command line information block
!       TMSCOL          - Default TMS column width
!
! IMPLICIT OUTPUTS:
!
!       TMS_TMP         - Modified
!       TMSTOF          - Contains top-of-file string
!       TMSSIZ          - Ideal file size in blocks
!       TMS_TXT_FMT     - Text format markup (different for /TMS=A and /TMS=E)
!       TMS_FOOT        - Page footer markup (different for /TMS=A and /TMS=E)
!       TMS_PAGE        - New page markup (different for /TMS=A and /TMS=E)
!
! ROUTINE VALUE:
! COMPLETION CODES:
!
!       None
!
! SIDE EFFECTS:
!
!       None
!--
    BEGIN
    LOCAL
        C,
        TS;

    IF .CMDBLK [NDX$H_FORMAT] EQL TMS11_A
    THEN
        BEGIN
        !
        ! User specified /TMS11=A
        !
        $STR_COPY (TARGET = TMSTOF, STRING = '*start**text*');
        $STR_COPY (TARGET = TMS_TXT_FMT, STRING = '[f1p10]');
        $STR_COPY (TARGET = TMS_FOOT, STRING = '[va36][fb]Index+n');

        IF .CMDBLK [NDX$V_TELLTALE]
        THEN
            $STR_COPY (TARGET = TMS_PAGE, STRING = '[va50]/_/l')
        ELSE
            $STR_COPY (TARGET = TMS_PAGE, STRING = '[va50]/_/l[va50]');

            END
        ELSE
            BEGIN
            !
            ! Initialize for /TMS11=E
            !
            $STR_COPY (TARGET = TMSTOF, STRING = '*start2**etext*');
            $STR_COPY (TARGET = TMS_TXT_FMT, STRING = '[f13p10]');
            $STR_COPY (TARGET = TMS_FOOT, STRING = '[va36][p11][fb]Index+n');

            IF .CMDBLK [NDX$V_TELLTALE]
            THEN
                $STR_COPY (TARGET = TMS_PAGE, STRING = '[p10][va50]/_/l')
            ELSE
                $STR_COPY (TARGET = TMS_PAGE, STRING = '[p10][va50]/_/l[va50]');

            END;

        !
        ! Write the version number
        !
        $STR_COPY (TARGET = TMS_TMP,
            STRING = $STR_CONCAT ('< INDEX version ', (.NDXVRL, .NDXVRP), ' >'));
        TMSPUT (.TMS_TMP [STR$H_LENGTH], .TMS_TMP [STR$A_POINTER], OUTIOB, FALSE);

        !
        ! Write the command line
        !
        $STR_COPY (STRING = $STR_CONCAT ('< ', CMDBLK [NDX$T_COMMAND_LINE], ' >'), TARGET = TMS_TMP);
        TMSPUT (.TMS_TMP [STR$H_LENGTH], .TMS_TMP [STR$A_POINTER], OUTIOB, FALSE);

        !
        ! Compute tab stop
        !
        TS = .CMDBLK [NDX$G_COLUMN_WID] * 18;
        TS = (IF (.TS MOD TMSCOL) NEQ 0 THEN 1 ELSE 0) + (.TS / TMSCOL);

        !
        ! Compute column width and ideal file size
        !
        SELECTONE .CMDBLK [NDX$H_LAYOUT] OF
        SET

        [TWO_COLUMN]:
            BEGIN
            C = .CMDBLK [NDX$G_COLUMN_WID] * 2;

            TMSSIZ = 40;                                ! Ideal size is 40 blocks for TWO_COLUMN output
            END;

        [SEPARATE]:
            BEGIN
            C = .CMDBLK [NDX$G_COLUMN_WID] + .CMDBLK [NDX$G_SEPARATE_WIDTH];

            TMSSIZ = 25;                                ! Ideal size is 25 blocks for SEPARATE MASTER format
            END;
```

```
; 1698          2171  2       [OTHERWISE]:
; 1699          2172  3           BEGIN
; 1700          2173  3           C = .CMDBLK [NDX$G_COLUMN_WID];
; 1701          2174  3
; 1702          2175  3           TMSSIZ = 20;                                 ! Ideal size is 20 blocks for ONE_COLUMN or GALLEY formats
; 1703          2176  2           END;
; 1704          2177  2
; 1705          2178  2       TES;
; 1706          2179  2
; 1707          2180  3       C = (IF (.C MOD TMSCOL) NEQ 0 THEN 1 ELSE 0)
; 1708          2181  2           + (.C * 18 / TMSCOL) + .CMDBLK [NDX$G_GUTTER_WID];
; 1709          2182  2
; 1710          2183  2       !
; 1711          2184  2       ! Build top of file string and write it out
; 1712          2185  2       !
; 1713        P 2186  2       $STR_APPEND (TARGET = TMSTOF,
; 1714        P 2187  2           STRING = $STR_CONCAT ('[v12c', $STR_ASCII (.C), 'ts',
; 1715        P 2188  2               $STR_ASCII (.TS), ',', $STR_ASCII (.TS + .CMDBLK [NDX$G_GUTTER_WID]),
; 1716          2189  2               ']', TMS_TXT_FMT));
; 1717          2190  2
; 1718          2191  2       TMSPUT (.TMSTOF [STR$H_LENGTH], .TMSTOF [STR$A_POINTER], OUTIOB, FALSE);
; 1719          2192  2       TMSPUT (1, .BLANKS, OUTIOB, FALSE);
; 1720          2193  2
; 1721          2194  2       !
; 1722          2195  2       ! Put out title
; 1723          2196  2       !
; 1724          2197  2       $STR_COPY (STRING = $STR_CONCAT (TMS_TITLE, TMS_TXT_FMT), TARGET = TMS_TMP);
; 1725          2198  2       TMSPUT (.TMS_TMP [STR$H_LENGTH], .TMS_TMP [STR$A_POINTER], OUTIOB, FALSE);
; 1726          2199  2
; 1727          2200  1       END;



                                                                                         .PSECT  $PLIT$,NOWRT,NOEXE,2

       2A  74  78  65  74  2A  2A  74  72  61  74  73  2A  00076 P.ABJ:    .ASCII  \*start**text*\                         :
                               5D  30  31  70  31  66  5B  00083 P.ABK:    .ASCII  \[f1p10]\                               :
   78  65  64  6E  49  5D  62  66  5B  5D  36  33  61  76  5B  0008A P.ABL:    .ASCII  \[va36][fb]Index+n\                     :
                                                       6E  2B  00099                                                       :
                           6C  2F  5F  2F  5D  30  35  61  76  5B  0009B P.ABM:    .ASCII  \[va50]/_/l\                     :
   30  35  61  76  5B  6C  2F  5F  2F  5D  30  35  61  76  5B  000A5 P.ABN:    .ASCII  \[va50]/_/l[va50]\                     :
                                                           5D  000B4                                                       :
   2A  74  78  65  74  65  2A  2A  32  74  72  61  74  73  2A  000B5 P.ABO:    .ASCII  \*start2**etext*\                      :
                               5D  30  31  70  33  31  66  5B  000C4 P.ABP:    .ASCII  \[f13p10]\                             :
   5D  62  66  5B  5D  31  31  70  5B  5D  36  33  61  76  5B  000CC P.ABQ:    .ASCII  \[va36][p11][fb]Index+n\               :
                                   6E  2B  78  65  64  6E  49  000DB                                                       :
   6C  2F  5F  2F  5D  30  35  61  76  5B  5D  30  31  70  5B  000E2 P.ABR:    .ASCII  \[p10][va50]/_/l\                      :
   6C  2F  5F  2F  5D  30  35  61  76  5B  5D  30  31  70  5B  000F1 P.ABS:    .ASCII  \[p10][va50]/_/l[va50]\                :
                                       5D  30  35  61  76  5B  00100                                                       :
   6E  6F  69  73  72  65  76  20  58  45  44  4E  49  20  3C  00106 P.ABV:    .ASCII  \< INDEX version \                     :
                                                           20  00115                                                       :
                                                       3E  20  00116 P.ABW:    .ASCII  \ >\                                   :
                                                       20  3C  00118 P.ACB:    .ASCII  \< \                                   :
                                                       3E  20  0011A P.ACC:    .ASCII  \ >\                                   :
                                           63  32  31  76  5B  0011C P.ACJ:    .ASCII  \[v12c\                                :
                                                       73  74  00121 P.ACK:    .ASCII  \ts\                                   :
                                                           2C  00123 P.ACL:    .ASCII  \,\                                    :
```

```
             5D  00124 P.ACM:   .ASCII  \]\                                          ;

                                .PSECT  $OWN$,NOEXE,2

           000D  000A4 $STR$STRING:
                                .WORD   13                                           ;
          01 0E  000A6          .BYTE   14, 1                                        ;
      00000000'  000A8          .ADDRESS P.ABJ                                       ;
           0007  000AC $STR$STRING:
                                .WORD   7                                            ;
          01 0E  000AE          .BYTE   14, 1                                        ;
      00000000'  000B0          .ADDRESS P.ABK                                       ;
           0011  000B4 $STR$STRING:
                                .WORD   17                                           ;
          01 0E  000B6          .BYTE   14, 1                                        ;
      00000000'  000B8          .ADDRESS P.ABL                                       ;
           000A  000BC $STR$STRING:
                                .WORD   10                                           ;
          01 0E  000BE          .BYTE   14, 1                                        ;
      00000000'  000C0          .ADDRESS P.ABM                                       ;
           0010  000C4 $STR$STRING:
                                .WORD   16                                           ;
          01 0E  000C6          .BYTE   14, 1                                        ;
      00000000'  000C8          .ADDRESS P.ABN                                       ;
           000F  000CC $STR$STRING:
                                .WORD   15                                           ;
          01 0E  000CE          .BYTE   14, 1                                        ;
      00000000'  000D0          .ADDRESS P.ABO                                       ;
           0008  000D4 $STR$STRING:
                                .WORD   8                                            ;
          01 0E  000D6          .BYTE   14, 1                                        ;
      00000000'  000D8          .ADDRESS P.ABP                                       ;
           0016  000DC $STR$STRING:
                                .WORD   22                                           ;
          01 0E  000DE          .BYTE   14, 1                                        ;
      00000000'  000E0          .ADDRESS P.ABQ                                       ;
           000F  000E4 $STR$STRING:
                                .WORD   15                                           ;
          01 0E  000E6          .BYTE   14, 1                                        ;
      00000000'  000E8          .ADDRESS P.ABR                                       ;
           0015  000EC $STR$STRING:
                                .WORD   21                                           ;
          01 0E  000EE          .BYTE   14, 1                                        ;
      00000000'  000F0          .ADDRESS P.ABS                                       ;
           0010  000F4 $STR$STRING0:
                                .WORD   16                                           ;
          01 0E  000F6          .BYTE   14, 1                                        ;
      00000000'  000F8          .ADDRESS P.ABV                                       ;
           0002  000FC $STR$STRING2:
                                .WORD   2                                            ;
          01 0E  000FE          .BYTE   14, 1                                        ;
      00000000'  00100          .ADDRESS P.ABW                                       ;
           0002  00104 $STR$STRING0:
                                .WORD   2                                            ;
          01 0E  00106          .BYTE   14, 1                                        ;
      00000000'  00108          .ADDRESS P.ACB                                       ;
           0002  0010C $STR$STRING2:
```

```
                                                  .WORD    2                                          :
                     01  0E  0010E                .BYTE    14, 1                                      :
               00000000'     00110                .ADDRESS P.ACC                                      :
                     0005    00114 $STR$STRING0:                                                      
                                                  .WORD    5                                          :
                     01  0E  00116                .BYTE    14, 1                                      :
               00000000'     00118                .ADDRESS P.ACJ                                      :
                     0002    0011C $STR$STRING2:                                                      
                                                  .WORD    2                                          :
                     01  0E  0011E                .BYTE    14, 1                                      :
               00000000'     00120                .ADDRESS P.ACK                                      :
                     0001    00124 $STR$STRING4:                                                      
                                                  .WORD    1                                          :
                     01  0E  00126                .BYTE    14, 1                                      :
               00000000'     00128                .ADDRESS P.ACL                                      :
                     0001    0012C $STR$STRING6:                                                      
                                                  .WORD    1                                          :
                     01  0E  0012E                .BYTE    14, 1                                      :
               00000000'     00130                .ADDRESS P.ACM                                      :

                                   $STR$TARGET=            TMS_TXT_FMT
                                   $STR$TARGET=            TMS_FOOT
                                   $STR$TARGET=            TMS_PAGE
                                   $STR$TARGET=            TMS_PAGE
                                   $STR$TARGET=            TMS_TXT_FMT
                                   $STR$TARGET=            TMS_FOOT
                                   $STR$TARGET=            TMS_PAGE
                                   $STR$TARGET=            TMS_PAGE
                                   $STR$TARGET=            TMS_TMP
                                   $STR$TARGET=            TMS_TMP
                                   $STR$STRING7=           TMS_TXT_FMT
                                   $STR$STRING0=           TMS_TITLE
                                   $STR$STRING1=           TMS_TXT_FMT
                                   $STR$TARGET=            TMS_TMP


                                                  .PSECT   $CODE$,NOWRT,2

                            0FFC 00000            .ENTRY   TMSINI, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-   ; 2058
                                                           R11
  5B 00000000G   EF   9E 00002                    MOVAB    TMSPUT, R11                                  :
  5A 00000000G   EF   9E 00009                    MOVAB    OUTIOB, R10                                  :
  59 00000000G   EF   9E 00010                    MOVAB    $STR$TARGET, R9                              :
  58 00000000G   EF   9E 00017                    MOVAB    CMDBLK+4, R8                                 :
  57 00000000G   EF   9E 0001E                    MOVAB    XST$COPY, R7                                 :
  56 00000000G   EF   9E 00025                    MOVAB    STR$FAILURE, R6                              :
  55 00000000'   EF   9E 0002C                    MOVAB    $STR$TARGET, R5                              :
  5E             08   C2 00033                    SUBL2    #8, SP                                       :
  02             68   B1 00036                    CMPW     CMDBLK+4, #2                                 ; 2098
                 4E   12 00039                    BNEQ     2$                                           :
                 56   DD 0003B                    PUSHL    R6                                           ; 2104
                 7E   D4 0003D                    CLRL     -(SP)                                        :
                 59   DD 0003F                    PUSHL    R9                                           :
          00A0   C5   9F 00041                    PUSHAB   $STR$STRING                                  :
                 7E   D4 00045                    CLRL     -(SP)                                        :
  67             05   FB 00047                    CALLS    #5, XST$COPY                                 :
                 56   DD 0004A                    PUSHL    R6                                           ; 2105
```

```
                                 7E  D4 0004C          CLRL    -(SP)                                  :
                           28    A5  9F 0004E          PUSHAB  $STR$TARGET                            :
                         00A8    C5  9F 00051          PUSHAB  $STR$STRING                            :
                                 7E  D4 00055          CLRL    -(SP)                                  :
                   67            05  FB 00057          CALLS   #5, XST$COPY                           :
                                 56  DD 0005A          PUSHL   R6                                     : 2106
                                 7E  D4 0005C          CLRL    -(SP)                                  :
                           38    A5  9F 0005E          PUSHAB  $STR$TARGET                            :
                         00B0    C5  9F 00061          PUSHAB  $STR$STRING                            :
                                 7E  D4 00065          CLRL    -(SP)                                  :
                   67            05  FB 00067          CALLS   #5, XST$COPY                           :
        0D         FD      A8    04  E1 0006A          BBC     #4, CMDBLK+1, 1$                       : 2108
                                 56  DD 0006F          PUSHL   R6                                     : 2110
                                 7E  D4 00071          CLRL    -(SP)                                  :
                           40    A5  9F 00073          PUSHAB  $STR$TARGET                            :
                         00B8    C5  9F 00076          PUSHAB  $STR$STRING                            :
                                 59  11 0007A          BRB     4$                                     :
                                 56  DD 0007C 1$:      PUSHL   R6                                     : 2112
                                 7E  D4 0007E          CLRL    -(SP)                                  :
                           40    A5  9F 00080          PUSHAB  $STR$TARGET                            :
                         00C0    C5  9F 00083          PUSHAB  $STR$STRING                            :
                                 4C  11 00087          BRB     4$                                     :
                                 56  DD 00089 2$:      PUSHL   R6                                     : 2120
                                 7E  D4 0008B          CLRL    -(SP)                                  :
                                 59  DD 0008D          PUSHL   R9                                     :
                         00C8    C5  9F 0008F          PUSHAB  $STR$STRING                            :
                                 7E  D4 00093          CLRL    -(SP)                                  :
                   67            05  FB 00095          CALLS   #5, XST$COPY                           :
                                 56  DD 00098          PUSHL   R6                                     : 2121
                                 7E  D4 0009A          CLRL    -(SP)                                  :
                           28    A5  9F 0009C          PUSHAB  $STR$TARGET                            :
                         00D0    C5  9F 0009F          PUSHAB  $STR$STRING                            :
                                 7E  D4 000A3          CLRL    -(SP)                                  :
                   67            05  FB 000A5          CALLS   #5, XST$COPY                           :
                                 56  DD 000A8          PUSHL   R6                                     : 2122
                                 7E  D4 000AA          CLRL    -(SP)                                  :
                           38    A5  9F 000AC          PUSHAB  $STR$TARGET                            :
                         00D8    C5  9F 000AF          PUSHAB  $STR$STRING                            :
                                 7E  D4 000B3          CLRL    -(SP)                                  :
                   67            05  FB 000B5          CALLS   #5, XST$COPY                           :
        0D         FD      A8    04  E1 000B8          BBC     #4, CMDBLK+1, 3$                       : 2124
                                 56  DD 000BD          PUSHL   R6                                     : 2126
                                 7E  D4 000BF          CLRL    -(SP)                                  :
                           40    A5  9F 000C1          PUSHAB  $STR$TARGET                            :
                         00E0    C5  9F 000C4          PUSHAB  $STR$STRING                            :
                                 0B  11 000C8          BRB     4$                                     :
                                 56  DD 000CA 3$:      PUSHL   R6                                     : 2128
                                 7E  D4 000CC          CLRL    -(SP)                                  :
                           40    A5  9F 000CE          PUSHAB  $STR$TARGET                            :
                         00E8    C5  9F 000D1          PUSHAB  $STR$STRING                            :
                                 7E  D4 000D5 4$:      CLRL    -(SP)                                  :
                   67            05  FB 000D7          CALLS   #5, XST$COPY                           :
                   6E 00000000G  EF  B0 000DA          MOVW    NDXVRL, $STR$STRING1                   : 2136
             02    AE            0E  90 000E1          MOVB    #14, $STR$STRING1+2                    :
             03    AE            01  90 000E5          MOVB    #1, $STR$STRING1+3                     :
             04    AE 00000000G  EF  D0 000E9          MOVL    NDXVRP, $STR$STRING1+4                 :
                         00F8    C5  9F 000F1          PUSHAB  $STR$STRING2                           :
```

```
                                                                        G  7
NDXPAG              NDXPAG -- Output page formatting routines        16-Sep-1984 01:06:39    VAX-11 Bliss-32 V4.0-742          Page  74
V04-000             TMSINI -- Generate TMS11 top of file string      14-Sep-1984 13:07:15    [RUNOFF.SRC]NDXPAG.BLI;1           (10)

                                            04  AE  9F  000F5        PUSHAB  $STR$STRING1                        :
                                          00F0  C5  9F  000F8        PUSHAB  $STR$STRING0                        :
                        00000000G  EF           03  FB  000FC        CALLS   #3, XST$JOIN                        :
                                                56  DD  00103        PUSHL   R6                                  :
                                                7E  D4  00105        CLPL    -(SP)                               :
                                                21  BB  00107        PUSHR   #^M<R0,R5>                          :
                                                7E  D4  00109        CLRL    -(SP)                               :
                                   67           05  FB  0010B        CALLS   #5, XST$COPY                        :
                                                7E  D4  0010E        CLRL    -(SP)                               : 2137
                                                5A  DD  00110        PUSHL   R10                                 :
                                            04  A5  DD  00112        PUSHL   TMS_TMP+4                           :
                                   7E           65  3C  00115        MOVZWL  TMS_TMP, -(SP)                      :
                                   6B           04  FB  00118        CALLS   #4, TMSPUT                          :
                                          0108  C5  9F  0011B        PUSHAB  $STR$STRING2                        : 2142
                                            44  A8  9F  0011F        PUSHAB  $STR$STRING1                        :
                                          0100  C5  9F  00122        PUSHAB  $STR$STRING0                        :
                        00000000G  EF           03  FB  00126        CALLS   #3, XST$JOIN                        :
                                                56  DD  0012D        PUSHL   R6                                  :
                                                7E  D4  0012F        CLRL    -(SP)                               :
                                                21  BB  00131        PUSHR   #^M<R0,R5>                          :
                                                7E  D4  00133        CLRL    -(SP)                               :
                                   67           05  FB  00135        CALLS   #5, XST$COPY                        :
                                                7E  D4  00138        CLRL    -(SP)                               : 2143
                                                5A  DD  0013A        PUSHL   R10                                 :
                                            04  A5  DD  0013C        PUSHL   TMS_TMP+4                           :
                                   7E           65  3C  0013F        MOVZWL  TMS_TMP, -(SP)                      :
                                   6B           04  FB  00142        CALLS   #4, TMSPUT                          :
                                   53       08  A8  D0  00145        MOVL    CMDBLK+12, R3                       : 2148
                    52             53           12  C5  00149        MULL3   #18, R3, TS                         :
        7E          00             52           01  7A  0014D        EMUL    #1, TS, #0, -(SP)                   : 2149
        50          50             8E  00000000G    8F  7B  00152        EDIV    #TMSCOL, (SP)+, R0, R0              :
                                                50  D5  0015B        TSTL    R0                                  :
                                                05  13  0015D        BEQL    5$                                  :
                                   51           01  D0  0015F        MOVL    #1, R1                              :
                                                02  11  00162        BRB     6$                                  :
                                                51  D4  00164  5$:   CLRL    R1                                  :
                    50             52  00000000G    8F  C7  00166  6$:   DIVL3   #TMSCOL, TS, R0                     :
                    52             51           50  C1  0016E        ADDL3   R0, R1, TS                          :
                                   50       02  A8  32  00172        CVTWL   CMDBLK+6, R0                        : 2154
                                   01           50  B1  00176        CMPW    R0, #1                              : 2157
                                                0D  12  00179        BNEQ    7$                                  :
                    50             53           01  78  0017B        ASHL    #1, R3, C                           : 2159
                        00000000G  EF           28  D0  0017F        MOVL    #40, TMSSIZ                         : 2161
                                                1D  11  00186        BRB     9$                                  : 2154
                                   03           50  B1  00188  7$:   CMPW    R0, #3                              : 2164
                                                0E  12  0018B        BNEQ    8$                                  :
                    50             53       18  A8  C1  0018D        ADDL3   CMDBLK+28, R3, C                    : 2166
                        00000000G  EF           19  D0  00192        MOVL    #25, TMSSIZ                         : 2168
                                                0A  11  00199        BRB     9$                                  : 2154
                                   50           53  D0  0019B  8$:   MOVL    R3, C                               : 2173
                        00000000G  EF           14  D0  0019E        MOVL    #20, TMSSIZ                         : 2175
        7E          00             50           01  7A  001A5  9$:   EMUL    #1, C, #0, -(SP)                    : 2180
        51          51             8E  00000000G    8F  7B  001AA        EDIV    #TMSCOL, (SP)+, R1, R1              :
                                                51  D5  001B3        TSTL    R1                                  :
                                                05  13  001B5        BEQL    10$                                 :
                                   51           01  D0  001B7        MOVL    #1, R1                              :
                                                02  11  001BA        BRB     11$                                 :
```

```
                                      51  D4 001BC 10$:    CLRL     R1                                      ;
             53           50          12  C5 001BE 11$:    MULL3    #18, C, R3                              ; 2181
                          53 00000000G    8F  C6 001C2     DIVL2    #TMSCOL, R3                             ;
                          51              53  C0 001C9     ADDL2    R3, R1                                  ;
             50           51          0C  A8  C1 001CC     ADDL3    CMDBLK+16, R1, C                        ;
                                      7E  D4 001D1         CLRL     -(SP)                                   ; 2189
                                      50  DD 001D3         PUSHL    C                                       ;
                          7E      0903    8F  3C 001D5     MOVZWL   #2307, -(SP)                            ;
               00000000G  EF              03  FB 001DA     CALLS    #3, XST$ASCII                           ;
                          54              50  D0 001E1     MOVL     R0, R4                                  ;
                                      7E  D4 001E4         CLRL     -(SP)                                   ;
                                      52  DD 001E6         PUSHL    TS                                      ;
                          7E      0903    8F  3C 001E8     MOVZWL   #2307, -(SP)                            ;
               00000000G  EF              03  FB 001ED     CALLS    #3, XST$ASCII                           ;
                          53              50  D0 001F4     MOVL     R0, R3                                  ;
                                      7E  D4 001F7         CLRL     -(SP)                                   ;
                                0C B842   9F 001F9         PUSHAB   @CMDBLK+16[TS]                          ;
                          7E      0903    8F  3C 001FD     MOVZWL   #2307, -(SP)                            ;
               00000000G  EF              03  FB 00202     CALLS    #3, XST$ASCII                           ;
                                      28  A5  9F 00209     PUSHAB   $STR$STRING7                            ;
                                    0128  C5  9F 0020C     PUSHAB   $STR$STRING6                            ;
                                      50  DD 00210         PUSHL    R0                                      ;
                                    0120  C5  9F 00212     PUSHAB   $STR$STRING4                            ;
                                      53  DD 00216         PUSHL    R3                                      ;
                                    0118  C5  9F 00218     PUSHAB   $STR$STRING2                            ;
                                      54  DD 0021C         PUSHL    R4                                      ;
                                    0110  C5  9F 0021E     PUSHAB   $STR$STRING0                            ;
               00000000G  EF              08  FB 00222     CALLS    #8, XST$JOIN                            ;
                                      56  DD 00229         PUSHL    R6                                      ;
                                      7E  D4 0022B         CLRL     -(SP)                                   ;
                                    0201  8F  BB 0022D     PUSHR    #^M<R0,R9>                              ;
                                      7E  D4 00231         CLRL     -(SP)                                   ;
               00000000G  EF              05  FB 00233     CALLS    #5, XST$APPEND                          ;
                                      7E  D4 0023A         CLRL     -(SP)                                   ; 2191
                                      5A  DD 0023C         PUSHL    R10                                     ;
                                      04  A9  DD 0023E     PUSHL    TMSTOF+4                                ;
                          7E              69  3C 00241     MOVZWL   TMSTOF, -(SP)                           ;
                          6B              04  FB 00244     CALLS    #4, TMSPUT                              ;
                                      7E  D4 00247         CLRL     -(SP)                                   ; 2192
                                      5A  DD 00249         PUSHL    R10                                     ;
                                      FC  A5  DD 0024B     PUSHL    BLANKS                                  ;
                                      01  DD 0024E         PUSHL    #1                                      ;
                          6B              04  FB 00250     CALLS    #4, TMSPUT                              ;
                                      28  A5  9F 00253     PUSHAB   $STR$STRING1                            ; 2197
                                      08  A5  9F 00256     PUSHAB   $STR$STRING0                            ;
               00000000G  EF              02  FB 00259     CALLS    #2, XST$JOIN                            ;
                                      56  DD 00260         PUSHL    R6                                      ;
                                      7E  D4 00262         CLRL     -(SP)                                   ;
                                      21  BB 00264         PUSHR    #^M<R0,R5>                              ;
                                      7E  D4 00266         CLRL     -(SP)                                   ;
                          67              05  FB 00268     CALLS    #5, XST$COPY                            ;
                                      7E  D4 0026B         CLRL     -(SP)                                   ; 2198
                                      5A  DD 0026D         PUSHL    R10                                     ;
                                      04  A5  DD 0026F     PUSHL    TMS_TMP+4                               ;
                          7E              65  3C 00272     MOVZWL   TMS_TMP, -(SP)                          ;
                          6B              04  FB 00275     CALLS    #4, TMSPUT                              ;
                                          04 00278         RET                                              ; 2200
```

```
; Routine Size:  633 bytes,    Routine Base:  $CODE$ + 0C6D
```

```
1 %SBTTL 'TELLTALE_HEAD -- Generate and output a telltale heading'
1 ROUTINE TELLTALE_HEAD : NOVALUE =
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine generates and outputs a telltale heading.
1 !
1 !       No heading is generated for the first page.
1 !       All emphasis except overstriking is removed from the heading string.
1 !       If generating a heading for RUNOFF, each character is bolded.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       None
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       PAGENO          - Page number
1 !       RLINES [0, ...] - Right telltale
1 !       RTYPE [0]       - Right telltale line type
1 !       LLINES [0, ...] - Left telltale
1 !       LTYPE [0]       - Left telltale line type
1 !       CMDBLK          - Command line information block
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       The telltale heading is written to the output file if not page 1.
1 !
1 !       LLINES [0, ...] - Set to right telltale string if generating a right
1 !                         telltale heading.
1 !       LTYPE [0]       - Set to right telltale line type if generating a
1 !                         right telltale heading.
1 !       RLINES [0, ...] - Right telltale is set to the null string if a right
1 !                         telltale was generated.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       None
1 !
1 ! SIDE EFFECTS:
1 !
1 !       None
1 !--
2    BEGIN
2
2    LOCAL
2       T_PTR,
2       I_PTR,
2       I_LEN,
2       O_PTR,
2       O_LEN,
2       O_BUF : VECTOR [CH$ALLOCATION (1024)];
2
2    BIND
2       STR = LLINES [0, 0,0,0,0] : $STR_DESCRIPTOR ();
```

```
2
2       IF .PAGENO
2       THEN
3           BEGIN
3           !
3           ! Odd numbered page.
3           ! Save the right telltale as the left telltale and
3           ! set the right telltale to the null string.
3           !
3           $STR_COPY (STRING = RLINES [0, 0,0,0,0], TARGET = LLINES [0, 0,0,0,0]);
3           LTYPE [0] = .RTYPE [0];
3           $STR_COPY (STRING = '', TARGET = RLINES [0, 0,0,0,0]);
2           END;
2
2       !
2       ! If this is page 1, do nothing; just return.
2       !
2       IF .PAGENO EQL 1 THEN RETURN;
2
2       !
2       ! Initialize pointers to input and output strings and string lengths.
2       !
2       I_PTR = .STR [STR$A_POINTER];
2       I_LEN = .STR [STR$H_LENGTH];
2       O_LEN = 0;
2       O_PTR = CH$PTR (O_BUF);
2
2       IF .LTYPE [0] EQL CONT_HEAD
2       THEN
3           BEGIN
3           !
3           ! Line was a continuation heading. Search for '(Cont_.)'
3           !
3           T_PTR = CH$FIND_SUB (.I_LEN, .I_PTR, 8, CH$PTR (UPLIT ('(Cont_.)')));
3           END
2       ELSE
3           BEGIN
3           !
3           ! Line was an index entry. Search for the NULL which delimits the
3           ! start of the page references (if any).
3           !
3           T_PTR = CH$FIND_CH (.I_LEN, .I_PTR, 0);
2           END;
2
2       IF NOT CH$FAIL (.T_PTR)
2       THEN
3           BEGIN
3           !
3           ! There are either page references or '(Cont_.)' on the line which
3           ! should be ignored.
3           !
3           I_LEN = CH$DIFF (.T_PTR, .I_PTR);
2           END;
2
2       WHILE .I_LEN GTR 0 DO
3           BEGIN
3           !
```

```
        ! Copy the line removing all emphasis except overstriking.
        ! If generating RUNOFF output, bold each character.
        !
        LOCAL
            CH;

        CH = CH$RCHAR_A (I_PTR);
        I_LEN = .I_LEN - 1;

        SELECTONE .CH OF
            SET

            [%C'*', %C'&']:
                !
                ! Bold or underline. Ignore it.
                !
                ;

            [%C'_']:
                BEGIN
                !
                ! Accept flag. Write it and the next character out.
                !
                IF .CMDBLK [NDX$H_FORMAT] EQL DSR
                THEN
                    BEGIN
                    !
                    ! Bold the character
                    !
                    CH$WCHAR_A (%C'*', O_PTR);
                    O_LEN = .O_LEN + 1;
                    END;

                CH$WCHAR_A (.CH, O_PTR);
                CH$WCHAR_A (CH$RCHAR_A (I_PTR), O_PTR);
                I_LEN = .I_LEN - 1;
                O_LEN = .O_LEN + 2;
                END;

            [%C'%']:
                BEGIN
                !
                ! Overstrike flag. Just write it out.
                !
                CH$WCHAR_A (.CH, O_PTR);                ! Write overstrike flag
                O_LEN = .O_LEN + 1;
                END;

            [OTHERWISE]:
                BEGIN
                !
                ! A normal character.
                !
                IF .CMDBLK [NDX$H_FORMAT] EQL DSR
                THEN
                    BEGIN
                    !
```

```
5                               ! Bold the character
5                               !
5                               CH$WCHAR_A (%C'*', O_PTR);
5                               O_LEN = .O_LEN + 1;
4                               END;
4
4                           CH$WCHAR_A (.CH, O_PTR);
4                           O_LEN = .O_LEN + 1;
3                           END;
3
3                   TES;
3
2               END;
2
2       O_PTR = CH$PTR (O_BUF);
2
2       SELECTONE .CMDBLK [NDX$H_FORMAT] OF
2           SET
2
2           [DSR]:
2               PUT_LINE ($STR_CONCAT ('.SUBTITLE ', (.O_LEN, .O_PTR)));
2
2           [TMS11_A, TMS11_E]:
3               BEGIN
3               !
3               ! Write telltale for /TMS
3               !
3               RNOTMS (.O_LEN, .O_PTR, TMS_TMP);   ! Convert special characters
3
3               IF .PAGENO
3               THEN                                ! Odd page - right telltale
4                   $STR_APPEND (STRING = TMS_RIGHT, TARGET = TMS_TMP)
3               ELSE                                ! Even page - left telltale
3                   $STR_APPEND (STRING = TMS_LEFT, TARGET = TMS_TMP);
3
3               IF .CMDBLK [NDX$H_FORMAT] EQL TMS11_A
3               THEN
4                   $STR_COPY (STRING = '*telltale*', TARGET = TMS_TELLTALE)
3               ELSE
3                   $STR_COPY (STRING = '*etelltale*', TARGET = TMS_TELLTALE);
3
3               $STR_APPEND (STRING = $STR_CONCAT (TMS_TMP, '@'), TARGET = TMS_TELLTALE);
3               TMSPUT (.TMS_TELLTALE [STR$H_LENGTH], .TMS_TELLTALE [STR$A_POINTER], OUTIOB, TRUE);
2               END;
2
2           [TEX]:
3               BEGIN
3               !
3               ! Write telltale for TEX
3               !
3               RNOTEX (.O_LEN, .O_PTR, TMS_TMP);
3
3               IF .PAGENO
3               THEN
4                   BEGIN
4                   !
4                   ! Odd page, telltale is flush right
```

```
; 1957          2429  4                    !
; 1958        P 2430  4                    $STR_COPY (TARGET = TMS_TELLTALE,
; 1959          2431  4                        STRING = $STR_CONCAT ('\telltale{\hfill ', TMS_TMP, '}'));
; 1960          2432  4
; 1961          2433  4                    END
; 1962          2434  3                ELSE
; 1963          2435  4                    BEGIN
; 1964          2436  4                    !
; 1965          2437  4                    ! Even page, telltale is flush left
; 1966          2438  4                    !
; 1967        P 2439  4                    $STR_COPY (TARGET = TMS_TELLTALE,
; 1968          2440  4                        STRING = $STR_CONCAT ('\telltale{', TMS_TMP, '\hfill }'));
; 1969          2441  4
; 1970          2442  3                    END;
; 1971          2443  3
; 1972          2444  3                PUT_LINE (TMS_TELLTALE);
; 1973          2445  2                END;
; 1974          2446  2
; 1975          2447  2            TES;
; 1976          2448  2
; 1977          2449  1        END;



                                                                          .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                     00125 P.ACR:   .BLKB   0
                                                     00125          .BLKB   3
                          29  2E  5F  74  6E  6F  43  28  00128 P.ACS:   .ASCII  \(Cont.)\
                  20  45  4C  54  49  54  42  55  53  2F  00130 P.ACU:   .ASCII  \.SUBTITLE \
                  2A  65  6C  61  74  6C  6C  65  74  2A  0013A P.ACW:   .ASCII  \*telltale*\
              2A  65  6C  61  74  6C  6C  65  74  65  2A  00144 P.ACX:   .ASCII  \*etelltale*\
                                                      40  0014F P.ACZ:   .ASCII  \@\
6C  69  66  68  5C  7B  65  6C  61  74  6C  6C  65  74  5C  00150 P.ADD:   .ASCII  <92>\telltale{\<92>\hfill \
                                                  20  6C  0015F
                                                      7D  00161 P.ADE:   .ASCII  \}\
                  7B  65  6C  61  74  6C  6C  65  74  5C  00162 P.ADJ:   .ASCII  <92>\telltale{\
                          7D  20  6C  6C  69  66  68  5C  0016C P.ADK:   .ASCII  <92>\hfill }\

                                                                          .PSECT  $OWN$,NOEXE,2

                                               0000  00134 $STR$STRING:
                                                                          .WORD   0
                                             01  0E  00136                .BYTE   14, 1
                                          00000000'  00138                .ADDRESS P.ACR
                                               000A  0013C $STR$STRING0:
                                                                          .WORD   10
                                             01  0E  0013E                .BYTE   14, 1
                                          00000000'  00140                .ADDRESS P.ACU
                                               000A  00144 $STR$STRING:
                                                                          .WORD   10
                                             01  0E  00146                .BYTE   14, 1
                                          00000000'  00148                .ADDRESS P.ACW
                                               000B  0014C $STR$STRING:
                                                                          .WORD   11
                                             01  0E  0014E                .BYTE   14, 1
                                          00000000'  00150                .ADDRESS P.ACX
```

```
                                      0001  00154 $STR$STRING1:
                                                          .WORD     1                                          :
                                    01  0E  00156         .BYTE     14, 1                                      :
                              00000000'  00158            .ADDRESS  P.ACZ                                      :
                                      0011  0015C $STR$STRING0:
                                                          .WORD     17                                         :
                                    01  0E  0015E         .BYTE     14, 1                                      :
                              00000000'  00160            .ADDRESS  P.ADD                                      :
                                      0001  00164 $STR$STRING2:
                                                          .WORD     1                                          :
                                    01  0E  00166         .BYTE     14, 1                                      :
                              00000000'  00168            .ADDRESS  P.ADE                                      :
                                      000A  0016C $STR$STRING0:
                                                          .WORD     10                                         :
                                    01  0E  0016E         .BYTE     14, 1                                      :
                              00000000'  00170            .ADDRESS  P.ADJ                                      :
                                      0008  00174 $STR$STRING2:
                                                          .WORD     8                                          :
                                    01  0E  00176         .BYTE     14, 1                                      :
                              00000000'  00178            .ADDRESS  P.ADK                                      :

                                                  $STR$STRING=          TMS_RIGHT
                                                  $STR$STRING=          TMS_LEFT
                                                  $STR$TARGET=          TMS_TELLTALE
                                                  $STR$TARGET=          TMS_TELLTALE
                                                  $STR$STRING0=         TMS_TMP
                                                  $STR$STRING1=         TMS_TMP
                                                  $STR$TARGET=          TMS_TELLTALE
                                                  $STR$STRING1=         TMS_TMP
                                                  $STR$TARGET=          TMS_TELLTALE
                                                  $IOB$OUTPUT=          TMS_TELLTALE


                                                          .PSECT    $CODE$,NOWRT,2

                                      0FFC  00000 TELLTALE_HEAD:
                                                          .WORD     Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11      : 2202
                  5B 00000000G  EF   9E  00002            MOVAB     XST$COPY, R11                              :
                  5A 00000000G  EF   9E  00009            MOVAB     IOB$+68, R10                               :
                  59 00000000G  EF   9E  00010            MOVAB     STR$FAILURE, R9                            :
                  58 00000000'  EF   9E  00017            MOVAB     TMS_TMP, R8                                :
                  5E     FBF8   CE   9E  0001E            MOVAB     -1032(SP), SP                              :
                  33 00000000G  EF   E9  00023            BLBC      PAGENO, 1$                                 : 2259
                               59    DD  0002A            PUSHL     R9                                         : 2267
                               7E    D4  0002C            CLRL      -(SP)                                      :
                     00000000G  EF   9F  0002E            PUSHAB    $STR$TARGET                                :
                     00000000G  EF   9F  00034            PUSHAB    $STR$STRING                                :
                               7E    D4  0003A            CLRL      -(SP)                                      :
                  6B           05    FB  0003C            CALLS     #5, XST$COPY                               :
      00000000G   EF 00000000G  EF   D0  0003F            MOVL      RTYPE, LTYPE                               : 2268
                               59    DD  0004A            PUSHL     R9                                         : 2269
                               7E    D4  0004C            CLRL      -(SP)                                      :
                     00000000G  EF   9F  0004E            PUSHAB    $STR$TARGET                                :
                          0130  C8   9F  00054            PUSHAB    $STR$STRING                                :
                               7E    D4  00058            CLRL      -(SP)                                      :
                  6B           05    FB  0005A            CALLS     #5, XST$COPY                               :
                  01 00000000G  EF   D1  0005D 1$:        CMPL      PAGENO, #1                                 : 2275
```

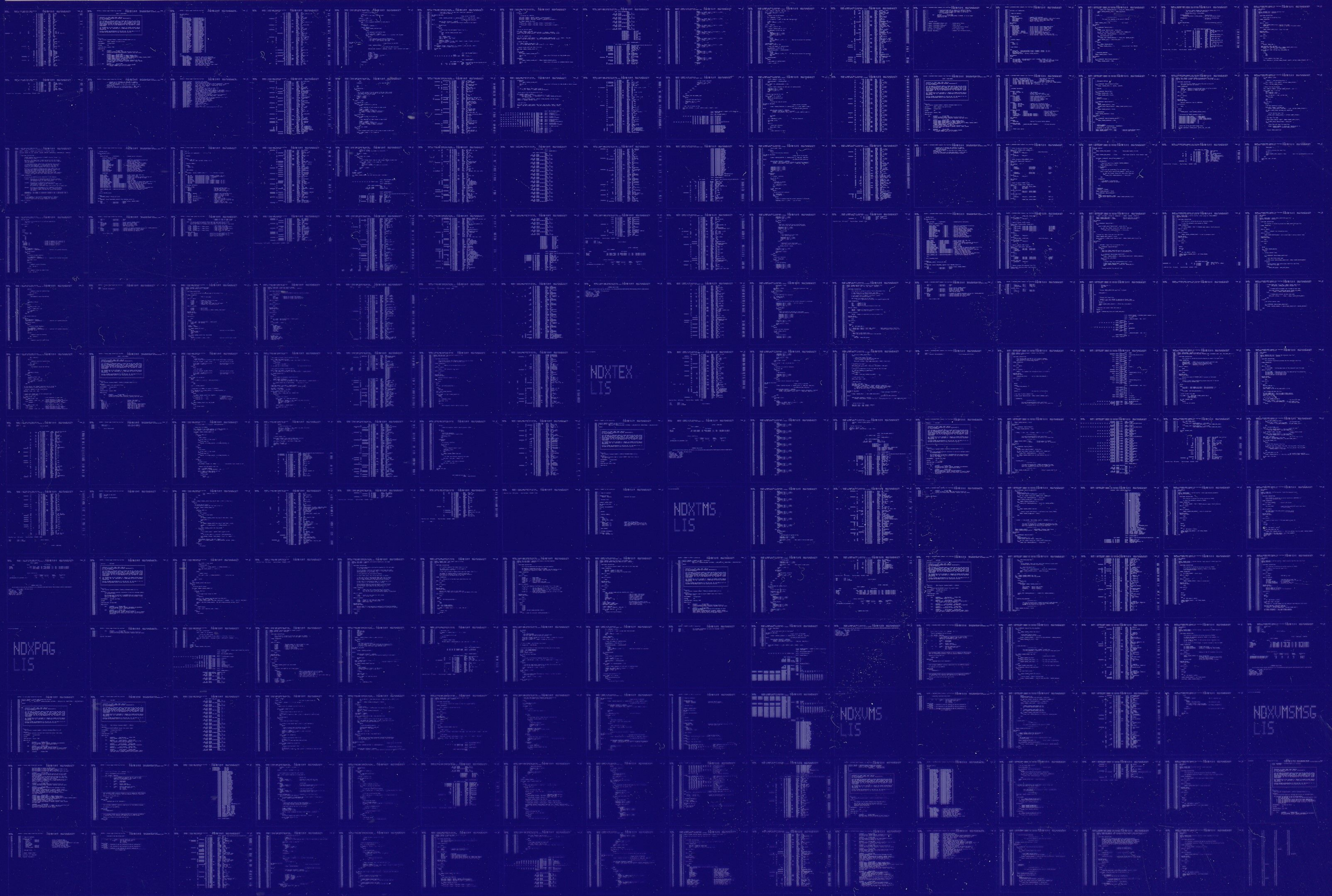
```
                                         01  12 00064          BNEQ    2$                                          ;
                                             04 00066          RET                                                 ;
                           57 00000000G  EF  D0 00067 2$:      MOVL    STR+4, I_PTR                                ; 2280
                           56 00000000G  EF  3C 0006E          MOVZWL  STR, I_LEN                                  ; 2281
                                         55  D4 00075          CLRL    O_LEN                                       ; 2282
                           54        08  AE  9E 00077          MOVAB   O_BUF, O_PTR                                ; 2283
                           0B 00000000G  EF  D1 0007B          CMPL    LTYPE, #T1                                  ; 2285
                                         13  12 00082          BNEQ    4$                                          ;
        67           56 00000000'   EF   08  39 00084          MATCHC  #8, P.ACS, I_LEN, (I_PTR)                   ; 2291
                                         03  13 0008D          BEQL    3$                                          ;
                           53            08  D0 0008F          MOVL    #8, R3                                      ;
                           53            08  C2 00092 3$:      SUBL2   #8, R3                                      ;
                                         0B  11 00095          BRB     6$                                          ; 2285
                 67        56            00  3A 00097 4$:      LOCC    #0, I_LEN, (I_PTR)                          ; 2299
                                         02  12 0009B          BNEQ    5$                                          ;
                                         51  D4 0009D          CLRL    R1                                          ;
                           53            51  D0 0009F 5$:      MOVL    R1, T_PTR                                   ;
                                         04  13 000A2 6$:      BEQL    7$                                          ; 2302
                 56        53            57  C3 000A4          SUBL3   I_PTR, T_PTR, I_LEN                         ; 2309
                                         56  D5 000A8 7$:      TSTL    I_LEN                                       ; 2312
                                         4D  15 000AA          BLEQ    11$                                         ;
                           50            87  9A 000AC          MOVZBL  (I_PTR)+, CH                                ; 2321
                                         56  D7 000AF          DECL    I_LEN                                       ; 2322
                           26            50  D1 000B1          CMPL    CH, #38                                     ; 2327
                                         F2  13 000B4          BEQL    7$                                          ;
                           2A            50  D1 000B6          CMPL    CH, #42                                     ;
                                         ED  13 000B9          BEQL    7$                                          ;
                   0000005F   8F         50  D1 000BB          CMPL    CH, #95                                     ; 2333
                                         1B  12 000C2          BNEQ    9$                                          ;
                           01 00000000G  EF  B1 000C4          CMPW    CMDBLK+4, #1                                ; 2338
                                         05  12 000CB          BNEQ    8$                                          ;
                           84            2A  90 000CD          MOVB    #42, (O_PTR)+                               ; 2344
                                         55  D6 000D0          INCL    O_LEN                                       ; 2345
                           84            50  90 000D2 8$:      MOVB    CH, (O_PTR)+                                ; 2348
                           84            87  90 000D5          MOVB    (I_PTR)+, (O_PTR)+                          ; 2349
                                         56  D7 000D8          DECL    I_LEN                                       ; 2350
                           55            02  C0 000DA          ADDL2   #2, O_LEN                                   ; 2351
                                         C9  11 000DD          BRB     7$                                          ; 2324
                           25            50  D1 000DF 9$:      CMPL    CH, #37                                     ; 2354
                                         0E  13 000E2          BEQL    10$                                         ;
                           01 00000000G  EF  B1 000E4          CMPW    CMDBLK+4, #1                                ; 2368
                                         05  12 000EB          BNEQ    10$                                         ;
                           84            2A  90 000ED          MOVB    #42, (O_PTR)+                               ; 2374
                                         55  D6 000F0          INCL    O_LEN                                       ; 2375
                           84            50  90 000F2 10$:     MOVB    CH, (O_PTR)+                                ; 2378
                                         55  D6 000F5          INCL    O_LEN                                       ; 2379
                                         AF  11 000F7          BRB     7$                                          ; 2312
                           54        08  AE  9E 000F9 11$:     MOVAB   O_BUF, O_PTR                                ; 2386
                           50 00000000G  EF  32 000FD          CVTWL   CMDBLK+4, R0                                ; 2388
                           01            50  B1 00104          CMPW    R0, #1                                      ; 2391
                                         22  12 00107          BNEQ    12$                                         ;
                           6E            55  B0 00109          MOVW    O_LEN, $STR$STRING1                         ; 2392
                     02    AE            0E  90 0010C          MOVB    #14, $STR$STRING1+2                         ;
                     03    AE            01  90 00110          MOVB    #1, $STR$STRING1+3                          ;
                     04    AE            54  D0 00114          MOVL    O_PTR, $STR$STRING1+4                       ;
                                         5E  DD 00118          PUSHL   SP                                          ;
                                 0138    C8  9F 0011A          PUSHAB  $STR$STRINGO                                ;
```

```
                    00000000G  EF           02  FB 0011E        CALLS   #2, XST$JOIN                  ;
                               6A           50  D0 00125        MOVL    R0, IOB$+68                   ;
                                          00DF  31 00128        BRW     22$                           ;
                               02           50  B1 0012B 12$:   CMPW    R0, #2                        ; 2394
                               03           18  00 0012E        BGEQ    14$                           ;
                                          008F  31 00130 13$:   BRW     19$                           ;
                               03           50  B1 00133 14$:   CMPW    R0, #3                        ;
                                            F8  14 00136        BGTR    13$                           ;
                                     0110   8F  BB 00138        PUSHR   #^M<R4,R8>                    ; 2399
                                            55  DD 0013C        PUSHL   O_LEN                         ;
                    00000000G  EF           03  FB 0013E        CALLS   #3, RNOTMS                    ;
                               0B 00000000G EF  E9 00145        BLBC    PAGENO, 15$                   ; 2401
                                            59  DD 0014C        PUSHL   R9                            ; 2403
                                            7E  D4 0014E        CLRL    -(SP)                         ;
                                            58  DD 00150        PUSHL   R8                            ;
                                       20   A8  9F 00152        PUSHAB  $STR$STRING                   ;
                                            09  11 00155        BRB     16$                           ;
                                            59  DD 00157 15$:   PUSHL   R9                            ; 2405
                                            7E  D4 00159        CLRL    -(SP)                         ;
                                            58  DD 0015B        PUSHL   R8                            ;
                                       18   A8  9F 0015D        PUSHAB  $STR$STRING                   ;
                                            7E  D4 00160 16$:   CLRL    -(SP)                         ;
                    00000000G  EF           05  FB 00162        CALLS   #5, XST$APPEND                ;
                               02 00000000G EF  B1 00169        CMPW    CMDBLK+4, #2                  ; 2407
                                            0D  12 00170        BNEQ    17$                           ;
                                            59  DD 00172        PUSHL   R9                            ; 2409
                                            7E  D4 00174        CLRL    -(SP)                         ;
                                       30   A8  9F 00176        PUSHAB  $STR$TARGET                   ;
                                     0140   C8  9F 00179        PUSHAB  $STR$STRING                   ;
                                            0B  11 0017D        BRB     18$                           ;
                                            59  DD 0017F 17$:   PUSHL   R9                            ; 2411
                                            7E  D4 00181        CLRL    -(SP)                         ;
                                       30   A8  9F 00183        PUSHAB  $STR$TARGET                   ;
                                     0148   C8  9F 00186        PUSHAB  $STR$STRING                   ;
                                            7E  D4 0018A 18$:   CLRL    -(SP)                         ;
                               6B           05  FB 0018C        CALLS   #5, XST$COPY                  ;
                                     0150   C8  9F 0018F        PUSHAB  $STR$STRING1                  ; 2413
                                            58  DD 00193        PUSHL   R8                            ;
                    00000000G  EF           02  FB 00195        CALLS   #2, XST$JOIN                  ;
                                            59  DD 0019C        PUSHL   R9                            ;
                                            7E  D4 0019E        CLRL    -(SP)                         ;
                                       30   A8  9F 001A0        PUSHAB  TMS_TELLTALE                  ;
                                            50  DD 001A3        PUSHL   R0                            ;
                                            7E  D4 001A5        CLRL    -(SP)                         ;
                    00000000G  EF           05  FB 001A7        CALLS   #5, XST$APPEND                ;
                                            01  DD 001AE        PUSHL   #1                            ; 2414
                                       BC   AA  9F 001B0        PUSHAB  OUTIOB                        ;
                                       34   A8  DD 001B3        PUSHL   TMS_TELLTALE+4                ;
                               7E      30   A8  3C 001B6        MOVZWL  TMS_TELLTALE, -(SP)           ;
                    00000000G  EF           04  FB 001BA        CALLS   #4, TMSPUT                    ;
                                                04 001C1        RET                                   ; 2388
                               04           50  B1 001C2 19$:   CMPW    R0, #4                        ; 2417
                                            59  12 001C5        BNEQ    23$                           ;
                                     0110   8F  BB 001C7        PUSHR   #^M<R4,R8>                    ; 2422
                                            55  DD 001CB        PUSHL   O_LEN                         ;
                    00000000G  EF           03  FB 001CD        CALLS   #3, RNOTEX                    ;
                               0C 00000000G EF  E9 001D4        BLBC    PAGENO, 20$                   ; 2424
```

```
                                          0160  C8  9F 001DB          PUSHAB  $STR$STRING2                              ; 2431
                                                58  DD 001DF          PUSHL   R8                                        ;
                                          0158  C8  9F 001E1          PUSHAB  $STR$STRING0                              ;
                                                0A  11 001E5          BRB     21$                                       ;
                                          0170  C8  9F 001E7 20$:     PUSHAB  $STR$STRING2                              ; 2440
                                                58  DD 001EB          PUSHL   R8                                        ;
                                          0168  C8  9F 001ED          PUSHAB  $STR$STRING0                              ;
                          00000000G  EF         03  FB 001F1 21$:     CALLS   #3, XST$JOIN                              ;
                                                59  DD 001F8          PUSHL   R9                                        ;
                                                7E  D4 001FA          CLRL    -(SP)                                     ;
                                            30  A8  9F 001FC          PUSHAB  $STR$TARGET                               ;
                                                50  DD 001FF          PUSHL   R0                                        ;
                                                7E  D4 00201          CLRL    -(SP)                                     ;
                                 6B             05  FB 00203          CALLS   #5, XST$COPY                              ;
                                 6A         30  A8  9E 00206          MOVAB   $IOB$OUTPUT, IOB$+68                      ; 2444
                             E8  AA             07  90 0020A 22$:     MOVB    #7, IOB$+44                               ;
                                     00000000G  EF  9F 0020E          PUSHAB  XPO$FAILURE                               ;
                                                7E  D4 00214          CLRL    -(SP)                                     ;
                                            BC  AA  9F 00216          PUSHAB  IOB$                                      ;
                          00000000G  EF         03  FB 00219          CALLS   #3, XPO$PUT                               ;
                                                04 00220 23$:         RET                                               ; 2449

; Routine Size:  545 bytes,    Routine Base:  $CODE$ + 0EE6


; 1978          2450  1
; 1979          2451  1 END                                   ! End of module
; 1980          2452  0 ELUDOM



                                                                          .EXTRN  LIB$SIGNAL

;                                              PSECT SUMMARY
;
;      Name                              Bytes                           Attributes
;
;  $PLIT$                                    372  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                                     380  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                                   4359  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       150        25         252         00:00.2
;                                       COMMAND QUALIFIERS

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

; Size:         4359 code + 752 data bytes
; Run Time:        03:02.4
; Elapsed Time:    04:59.0
; Lines/CPU Min:      806
; Lexemes/CPU-Min:101836
; Memory Used:  492 pages
; Compilation Complete
```

0345 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



NDXTEX
LIS

NDXTMS
LIS

NDXPAG
LIS

NDXVMS
LIS

NDXVMSMSG
LIS