```
RRRRRRRRRRR      UUU          UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF   FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF   FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF   FFFFFFFFFFFFFFF
RRR          RRR UUU          UUU  NNN         NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNN         NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNN         NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNNNNN      NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNNNNN      NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNNNNN      NNN  000          000 FFF               FFF
RRRRRRRRRRR      UUU          UUU  NNN   NNN   NNN  000          000 FFFFFFFFFFF       FFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN   NNN   NNN  000          000 FFFFFFFFFFF       FFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN   NNN   NNN  000          000 FFFFFFFFFFF       FFFFFFFFFFF
RRR   RRR        UUU          UUU  NNN       NNNNNN 000          000 FFF               FFF
RRR   RRR        UUU          UUU  NNN       NNNNNN 000          000 FFF               FFF
RRR   RRR        UUU          UUU  NNN       NNNNNN 000          000 FFF               FFF
RRR      RRR     UUU          UUU  NNN          NNN 000          000 FFF               FFF
RRR      RRR     UUU          UUU  NNN          NNN 000          000 FFF               FFF
RRR      RRR     UUU          UUU  NNN          NNN 000          000 FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
```

```
_$2

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

RLI
RNO
RNO
RTY
SAV
STR
STR
STR
STR
STR
STR

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

```
NN       NN  DDDDDDDD    XX       XX  PPPPPPPP      000000    LL
NN       NN  DDDDDDDD    XX       XX  PPPPPPPP      000000    LL
NN       NN  DD       DD  XX     XX   PP       PP  00      00  LL
NN       NN  DD       DD  XX     XX   PP       PP  00      00  LL
NNNN     NN  DD       DD    XX  XX     PP       PP  00      00  LL
NNNN     NN  DD       DD    XX  XX     PP       PP  00      00  LL
NN  NN   NN  DD       DD      XX       PPPPPPPP    00      00  LL
NN  NN   NN  DD       DD      XX       PPPPPPPP    00      00  LL
NN    NNNN  DD       DD    XX  XX     PP          00      00  LL
NN    NNNN  DD       DD    XX  XX     PP          00      00  LL
NN      NN  DD       DD  XX     XX   PP          00      00  LL                ....
NN      NN  DD       DD  XX     XX   PP          00      00  LL                ....
NN      NN  DDDDDDDD    XX       XX  PP            000000    LLLLLLLLLL        ....
NN      NN  DDDDDDDD    XX       XX  PP            000000    LLLLLLLLLL        ....

RRRRRRRR    EEEEEEEEEE    QQQQQQ
RRRRRRRR    EEEEEEEEEE    QQQQQQ
RR      RR  EE          QQ      QQ
RR      RR  EE          QQ      QQ
RR      RR  EE          QQ      QQ
RR      RR  EE          QQ      QQ
RRRRRRRR    EEEEEEEE    QQ      QQ
RRRRRRRR    EEEEEEEE    QQ      QQ
RR  RR      EE          QQ  QQ  QQ
RR  RR      EE          QQ  QQ  QQ
RR    RR    EE          QQ    QQ
RR    RR    EE          QQ    QQ
RR      RR  EEEEEEEEEE    QQQQ  QQ
RR      RR  EEEEEEEEEE    QQQQ  QQ
```

Version: 'V04-000'


```
!++
! FACILITY:
!   DSR (Digital Standard RUNOFF) /DSRPLUS DSRINDEX/INDEX Utility
!
! ABSTRACT:
!   This file contains literals and macros defining the data structures
!   found in the internal index pool
!
! ENVIRONMENT:   Transportable
!
! AUTHOR:        JPK
!
! CREATION DATE: January 1982
!
! MODIFIED BY:
!
!       003     JPK00015        04-Feb-1983
!               Cleaned up module names, modified revision history to
!               conform with established standards. Updated copyright dates.
!
!       002     JPK00009        24-Jan-1983
!               Modified to enhance performance. The sort buckets have each
!               been divided into 27 sub-buckets; 1 for each letter and 1
!               for non-alphas. Removed reference to BUCKET from INDEX.
!               Definition of the structure was added to NDXPOL. References
!               to BUCKET were changed in modules NDXOUT, NDXINI, NDXFMT
!               and NDXDAT.
!--
```

```
! Index entry

$FIELD XE_FIELDS =
    SET

    XE$A_PREV           = [$ADDRESS],       ! Link to previous item
    XE$A_NEXT           = [$ADDRESS],       ! Link to next item
    XE$A_SUBX           = [$ADDRESS],       ! Sub index pointer
    XE$A_REF            = [$ADDRESS],       ! Reference pointer
    XE$A_TEXT           = [$ADDRESS],       ! Pointer to text of index item
    XE$A_SORT_AS        = [$ADDRESS],       ! Pointer to SORT_AS string
    XE$H_SUBC           = [$SHORT_INTEGER], ! Sub index level

    XE$V_FLAGS          = [$SHORT_INTEGER], ! Entry flags

        $OVERLAY (XE$V_FLAGS)

        XE$V_BARS               = [$BIT],   ! Change bar flag

        $CONTINUE

    XE$A_BOOK_LIST      = [$ADDRESS]        ! Master index book name list

    $ALIGN (FULLWORD)

    TES;

LITERAL
    XE$K_LENGTH = $FIELD_SET_SIZE;

MACRO
    $XE_BLOCK = BLOCK [XE$K_LENGTH] FIELD (XE_FIELDS) %;

! End of Index entry


! Reference entry

$FIELD XX_FIELDS =
    SET

    XX$A_LINK           = [$ADDRESS],       ! Link to additional entries
    XX$A_APPEND         = [$ADDRESS],       ! APPEND text pointer
    XX$H_PAGE           = [$SHORT_INTEGER], ! Transaction number

    XX$V_FLAGS          = [$SHORT_INTEGER], ! Display attributes

        $OVERLAY (XX$V_FLAGS)

        XX$V_BOLD       = [$BIT],           ! Bold page reference
        XX$V_UNDERLINE  = [$BIT],           ! Underline page reference
        XX$V_BEGIN      = [$BIT],           ! Begin page range
        XX$V_END        = [$BIT],           ! End page range

        $CONTINUE
```

```
    XX$A_BOOK           = [$ADDRESS]            ! Master index book name
    $ALIGN (FULLWORD)
    TES;

LITERAL
    XX$K_LENGTH = $FIELD_SET_SIZE;

MACRO
    $XX_BLOCK = BLOCK [XX$K_LENGTH] FIELD (XX_FIELDS) %;

! End of Reference entry

! Master index book reference entry

$FIELD XM_FIELDS =
    SET
    XM$A_LINK           = [$ADDRESS],       ! Link to additional entries
    XM$A_BOOK           = [$ADDRESS]        ! Pointer to book name
    TES;

LITERAL
    XM$K_LENGTH = $FIELD_SET_SIZE;

MACRO
    $XM_BLOCK = BLOCK [XM$K_LENGTH] FIELD (XM_FIELDS) %;

! End of Master index book reference entry

! Current Entry

$FIELD C_FIELDS =
    SET
    C$A_CURR            = [$ADDRESS],       ! Pointer to current cell
    C$A_PREV            = [$ADDRESS],       ! Pointer to previous cell
    C$A_HEAD            = [$ADDRESS],       ! Pointer to head of chain
    $ALIGN (FULLWORD)
    C$V_FLAGS           = [$INTEGER],       ! Current cell flags
        $OVERLAY (C$V_FLAGS)
        C$V_IDNS        = [$BIT]            ! Identical string flag
        $CONTINUE
    TES;
```

```
LITERAL
    C$K_LENGTH = $FIELD_SET_SIZE;

MACRO
    $C_BLOCK = BLOCK [C$K_LENGTH] FIELD (C_FIELDS) %;

! End of current entry


!
! Dummy datasets
!
LITERAL
    DS_X_ENTRY  = XE$K_LENGTH,
    DS_XX_ENTRY = XX$K_LENGTH,
    DS_XM_ENTRY = XM$K_LENGTH,
    DS_X_STRING = 0;

!
! Structure definition for bucket array.
!
!       Buckets are arranged so that each row represents the first letter of
!       the string and each column represents the second letter of the string.
!
!       This approach is used only for master indexes as no performance
!       improvement is realised until about 10 input files have been processed.
!
!       Indexes which are not master indexes use only the first element of
!       each row, i.e., [0, 0] ... [26, 0].
!
!       The only exception is for nonalphabetic characters which use only
!       element [0, 0]. Elements [0, 1] ... [0, 26] are not used since mapping
!       all nonalphabetics into one row loses the sort order of the first
!       character in the string. For nonalphabetics to work correctly in a two
!       dimensional bucket scheme, the array would have to be at least 127 x 127
!
!            0   1   .   .   .  26
!        0   *   not used   . . .
!        1   A.  AA           AZ
!        .                    .
!        .                    .
!        .                    .
!       26   Z?  ZA  .   .   .  ZZ
!
STRUCTURE
    $BUCKET_ARRAY [ROW_IDX, COL_IDX; M, N] =
        [M * N * %UPVAL] ($BUCKET_ARRAY + (ROW_IDX * N + COL_IDX) * %UPVAL);

!--     End of NDXPOL.REQ
```

CONVRT REQ
ECC REQ
FLIPRECS REQ
FNCT REQ
FRMSTK REQ
GETQSC REQ
GNCC REQ
IFSTK REQ
KWITEM REQ
LSTOPS REQ
NDXCLI REQ
NDXRTY REQ
OUTOPT REQ
PHDEF REQ
RUNTAB REQ
DIGIT REQ
FSPACK REQ
NDXXPL REQ
MAXIMA REQ
POOL REQ
RUNHAN REQ
DMDEFS REQ
FFDEFS REQ
GSLUCC REQ
INDEX REQ
NDXLIN REQ
PAGEN REQ
LETTER REQ
NMLST REQ
FOOFIL REQ
GCA REQ
MSG REQ
RNODEF REQ
DSRLIB REQ
FLGT REQ
HCT REQ
IRAC REQ
NDXPOL REQ
PASS REQ
LODEFS REQ
NUMPRM REQ
FOOREC REQ
MSGTXT REQ
NBITS REQ
KC REQ
PDT REQ
RNOMAC REQ
FLIPCHARS REQ
HLC REQ
LSTBTS REQ
OPDEV REQ