```
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
```

```
RRRRRRRR     NN       NN    000000    MM       MM    AAAAAA      CCCCCCCC
RRRRRRRR     NN       NN    000000    MM       MM    AAAAAA      CCCCCCCC
RR       RR  NN       NN  00      00  MMMM   MMMM   AA      AA  CC
RR       RR  NN       NN  00      00  MMMM   MMMM   AA      AA  CC
RR       RR  NNNN     NN  00      00  MM  MM  MM   AA      AA  CC
RR       RR  NNNN     NN  00      00  MM  MM  MM   AA      AA  CC
RRRRRRRR     NN  NN   NN  00      00  MM      MM   AA      AA  CC
RRRRRRRR     NN  NN   NN  00      00  MM      MM   AA      AA  CC
RR  RR       NN    NNNN  00      00  MM      MM   AAAAAAAAAA  CC
RR  RR       NN    NNNN  00      00  MM      MM   AAAAAAAAAA  CC
RR    RR     NN      NN  00      00  MM      MM   AA      AA  CC              ....
RR    RR     NN      NN  00      00  MM      MM   AA      AA  CC              ....
RR      RR   NN      NN    000000    MM      MM   AA      AA    CCCCCCCC      ....
RR      RR   NN      NN    000000    MM      MM   AA      AA    CCCCCCCC      ....

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
```

```
RUN

LIT
LIT
LIT
LIT
LIT
LIT
LIT
LIT
LIT
LIT
LIT
LIT

%IF
LIT
%FI

LIT
LIT
LIT
LIT
LIT
LIT

%IF
LIT
LIT
LIT
%FI

LIT
LIT
LIT
LIT
LIT
LIT
LIT
LIT

%IF
LIT
%FI

LIT

%IF
LIT
%FI

LIT
LIT
LIT
LIT
LIT
LIT
LIT
LIT
```

```
Version:      'V04-000'
```



```
!++
! FACILITY:      DSR (Digital Standard RUNOFF) / DSRPLUS
!
! ABSTRACT:
!       RUNOFF Command Information.  Initialized by the command line interface
!       and picked up for distribution by DOOPTS.
!
! ENVIRONMENT:   Transportable BLISS
!
! AUTHOR:        Rich Friday
!
! CREATION DATE: 1978
!
! MODIFIED BY:
!
!       012     REM00012        Ray Marshall    5-December-1983
!               Added RNO$V_LN01_LOAD to indicate that one wishes to load
!                 fonts before printing this file on an LN01.
!
!       011     REM00011        Ray Marshall    13-Sep-1983
!               Added bit RNO$V_LN01_HEADER to support disabling the font
!                 control header information from being put into an .LN1 output
!                 file.
!
!       010     KAD00010        Keith Dawson    2-Jun-1983
!               Added bits RNO$V_S_RIGHT and RNO$V_S_DOWN to tell if the
!                 qualifiers were explicitly given.
!
!       009     KAD00009        Keith Dawson    22-Mar-1983
```

```
!               Renamed LN01 italic/underline and portrait/landscape bits,
!                 in order to reflect the default values.
!               Got rid of RNO$V_FLIP bit, in favor of (.gca_op_dev
!                 EQL op_dev_flip).
!
!       008     KAD00008        Keith Dawson    07-Mar-1983
!               Global edit of all modules. Updated module names, idents,
!               copyright dates. Changed require files to BLISS library.
!--

$FIELD  $rno$cmd_fields = SET

! File specification string descriptors.
!
    RNO$T_INPUT         = [$DESCRIPTOR(FIXED)] ,        ! Input
    RNO$T_OUTPUT        = [$DESCRIPTOR(FIXED)] ,        ! Output
    RNO$T_INTERMEDIATE  = [$DESCRIPTOR(FIXED)] ,        ! Binary intermediate .BRN type file
    RNO$T_DROP_REC      = [$DESCRIPTOR(FIXED)] ,        ! Drop-record file

! Miscellaneous other string decriptors.
!
    rno$t_program_name  = [$DESCRIPTOR(FIXED)] ,        ! Name of program
    rno$t_program_filename = [$DESCRIPTOR(FIXED)] ,     ! Filename of program

    rno$t_pages         = [$DESCRIPTOR(FIXED)] ,        ! Page range string descriptor:
        $OVERLAY( $SUB_FIELD( rno$t_pages, STR$H_LENGTH ) )
    rno$h_pages         = [$BYTES(2)] ,                 !  length of page range string
        $OVERLAY( $SUB_FIELD( rno$t_pages, STR$A_POINTER ) )
    rno$a_pages         = [$POINTER] ,                  !  pointer to page range string
        $CONTINUE

    rno$t_variant       = [$DESCRIPTOR(FIXED)] ,        ! Variant string descriptor:
        $OVERLAY( $SUB_FIELD( rno$t_variant, STR$H_LENGTH ) )
    rno$h_variant       = [$BYTES(2)] ,                 !  length of variant string
        $OVERLAY( $SUB_FIELD( rno$t_variant, STR$A_POINTER ) )
    rno$a_variant       = [$POINTER] ,                  !  pointer to variant string
        $CONTINUE

    rno$a_datime        = [$BYTES(12)],                 !  System date and time
        $OVERLAY( rno$a_datime )
    rno$h_year          = [$BYTES(2)] ,
    rno$h_month         = [$BYTES(2)] ,
    rno$h_monthday      = [$BYTES(2)] ,
    rno$h_hours         = [$BYTES(2)] ,
    rno$h_minutes       = [$BYTES(2)] ,
    rno$h_seconds       = [$BYTES(2)] ,
        $CONTINUE

! Three words of switch option bits:
!
    rno$v_options       = [$BYTES(8)] ,                 ! Command option indicators:
        $OVERLAY ( rno$v_options )
    rno$v_option1       = [$BYTES(2)] ,
    rno$v_option2       = [$BYTES(2)] ,
    rno$v_option3       = [$BYTES(2)] ,
```

```
    rno$v_option4       = [$BYTES(2)] ,

        $OVERLAY ( rno$v_option1 )

    rno$v_2_automatic   = [$BITS(2)] ,              !  Carries state of /AUTOMATIC to DOOPTS
        $OVERLAY ( rno$v_2_automatic )
    rno$v_automatic     = [$BIT] ,                  !  execute multiple passes and utilities as needed
    rno$v_s_automatic   = [$BIT] ,                  !  /AUTOMATIC explicitly specified

    rno$v_2_backspace   = [$BITS(2)] ,              !  backspace on same line
        $OVERLAY ( rno$v_2_backspace )
    rno$v_backspace     = [$BIT] ,                  !  backspace on same line
    rno$v_s_backspace   = [$BIT] ,                  !  /BACKSPACE explicitly specified

    rno$v_2_change      = [$BITS(2)] ,              !  change bars requested
        $OVERLAY ( rno$v_2_change )
    rno$v_change        = [$BIT] ,                  !  change bars allowed
    rno$v_s_change      = [$BIT] ,                  !  /CHANGE explicitly specified

    rno$v_chng_char     = [$BIT] ,                  !  change bar character specified (see below)

    rno$v_2_cross_reference = [$BITS(2)] ,          !  Carries state of /CROSSREFERENCE to DOOPTS
        $OVERLAY ( rno$v_2_cross_reference )
    rno$v_cross_reference   = [$BIT] ,              !  translate cross references
    rno$v_s_cross_reference = [$BIT] ,              !  /CROSS_REFERENCE explicitly specified

    rno$v_deb_cond      = [$BIT] ,                  !  go through all paths of .IFs and output draft flags
    rno$v_deb_cont      = [$BIT] ,                  !  echo toc entries in output file
    rno$v_deb_cros      = [$BIT] ,                  !  echo .REFs & $FOOs in output file
    rno$v_deb_files     = [$BIT] ,                  !  put file names in the output.
    rno$v_deb_index     = [$BIT] ,                  !  echo index entries in output file
    rno$v_deb_save      = [$BIT] ,                  !  echo .SAVEs & .RESs in output file
!
! (1 free bit in OPTION1)

        $OVERLAY ( rno$v_option2 )

    rno$v_2_intermediate = [$BITS(2)] ,             !  Carries state of /INTERMEDIATE to DOOPTS
        $OVERLAY ( rno$v_2_intermediate )
    rno$v_intermediate  = [$BIT] ,                  !  a .BRN type file will be generated
    rno$v_s_intermediate = [$BIT] ,                 !  /INTERMEDIATE explicitly specified

    rno$v_2_log         = [$BITS(2)] ,              !  print final statistics
        $OVERLAY ( rno$v_2_log )
    rno$v_log           = [$BIT] ,                  !  print final statistics
    rno$v_s_log         = [$BIT] ,                  !  /LOG explicitly specified

    rno$v_msg_out       = [$BIT] ,                  !  put DSR messages in the output file
    rno$v_msg_user      = [$BIT] ,                  !  send DSR messages to the user

    rno$v_4_out_format  = [$BITS(4)] ,              !  defines output formats

    rno$v_2_output      = [$BITS(2)] ,              !  output file requested
        $OVERLAY ( rno$v_2_output )
    rno$v_output        = [$BIT] ,                  !  output file requested
    rno$v_s_output      = [$BIT] ,                  !  /OUTPUT explicitly specified
```

```
    rno$v_2_overprint   = [$BITS(2)] ,          .  overprint on same line
        $OVERLAY ( rno$v_2_overprint )
    rno$v_overprint     = [$BIT] ,              !  overprint on same line
    rno$v_s_overprint   = [$BIT] ,              !  /OVERPRINT  explicitly specified

    rno$v_2_pause       = [$BITS(2)] ,          !  pause at page boundary requested
        $OVERLAY ( rno$v_2_pause )
    rno$v_pause         = [$BIT] ,              !  pause at page boundary requested
    rno$v_s_pause       = [$BIT] ,              !  /PAUSE explicitly specified
!
! (NO free bits in OPTION2)
!
        $OVERLAY ( rno$v_option3 )

    rno$v_quick         = [$BIT] ,              !  do quick processing (for contents, index)

    rno$v_2_sequence    = [$BITS(2)] ,          !  output sequence numbers requested
        $OVERLAY ( rno$v_2_sequence )
    rno$v_sequence      = [$BIT] ,              !  output sequence numbers requested
    rno$v_s_sequence    = [$BIT] ,              !  /SEQUENCE explicitly specified

    rno$v_2_simulate    = [$BITS(2)] ,          !  simulate printer form feeds
        $OVERLAY ( rno$v_2_simulate )
    rno$v_simulate      = [$BIT] ,              !  simulate printer form feeds
    rno$v_s_simulate    = [$BIT] ,              !  /SIMULATE explicitly specified

    rno$v_2_underline   = [$BITS(2)] ,          !  underlining requested
        $OVERLAY ( rno$v_2_underline )
    rno$v_underline     = [$BIT] ,              !  underlining requested: (default)
    rno$v_s_underline   = [$BIT] ,              !  /UNDERLINE explicitly specified

    rno$v_2_und_separ   = [$BITS(2)] ,          !    underlining on separate line
        $OVERLAY ( rno$v_2_und_separ )
    rno$v_und_separ     = [$BIT] ,              !    underlining on separate line
    rno$v_s_und_separ   = [$BIT] ,              !    /SEPARATE_UNDERLINE explicitly specified

    rno$v_2_und_nonsp   = [$BITS(2)] ,          !    underlining non-spacing inline
        $OVERLAY ( rno$v_2_und_nonsp )
    rno$v_und_nonsp     = [$BIT] ,              !    non-spacing inline
    rno$v_s_und_nonsp   = [$BIT] ,              !    /NONSPACING explicitly specified

    rno$v_und_char      = [$BIT] ,              !    character specified (see below)

!
! (4 free bits in OPTION3)
!
        $OVERLAY ( rno$v_option4 )
                                        ! Options for LN01 output:
    rno$v_ln01_ital_under = [$BIT] ,    !    set = italics, clear = underlining
    rno$v_ln01_port_land = [$BIT] ,     !    set = portrait, clear = landscape
    rno$v_ln01_header = [$BIT] ,        !    set = if header wanted in O/P file
    rno$v_ln01_load = [$BIT] ,          !    set = loading of fonts is desired

    rno$v_s_down        = [$BIT] ,      !    /DOWN explicitly specified
    rno$v_s_right       = [$BIT] ,      !    /RIGHT explicitly specified
```

```
!
! (11 free bits in OPTION4)
!
        $CONTINUE

    rno$c_change        = [$BYTE] ,                 ! Change bar character (RNO$V_CHNG_CHAR set)
    rno$c_underline     = [$BYTE] ;                 ! Underline character (RNO$V_UND_CHAR set)

    rno$h_bold          = [$SHORT_INTEGER] ,        ! Bolding overprint count
    rno$h_dbg1          = [$SHORT_INTEGER] ,        ! Debug flags, 1st word
    rno$h_dbg2          = [$SHORT_INTEGER] ,        ! Debug flags, 2nd word
    rno$h_down          = [$SHORT_INTEGER] ,        ! Down shift count (number of lines)
    rno$h_right           [$SHORT_INTEGER] ,        ! Right shift count (number of characters)
    rno$h_form_size       [$SHORT_INTEGER]          ! Form size (number of lines)

    TES;

LITERAL rno$k_cmd_len   = $FIELD_SET_SIZE ;         ! Length of RUNOFF command block

MACRO
    $rno_cmd =
        BLOCK [ rno$k_cmd_len ]  FIELD ( $rno$cmd_fields )
    %;

!                       End of RNOMAC.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