

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:
 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 RN\$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 RN\$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 RN\$V_S_RIGHT and RN\$V_S_DOWN to tell if the qualifiers were explicitly given.		
009	KAD00009	Keith Dawson	22-Mar-1983

RUN
 LIT
 LIT
 LIT
 LIT
 ZIF
 LIT
 ZFI
 LIT
 ZIF
 LIT
 ZFI
 LIT
 LIT
 LIT
 ZIF
 LIT
 ZFI
 LIT
 LIT
 LIT
 LIT
 ZIF
 LIT
 LIT
 LIT
 LIT
 LIT
 LIT
 LIT
 LIT
 LIT

Renamed LN01 italic/underline and portrait/landscape bits,
in order to reflect the default values.
Got rid of RNOSV_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.

```
RNOST_INPUT      = [$DESCRIPTOR(FIXED)] ,      ! Input
RNOST_OUTPUT     = [$DESCRIPTOR(FIXED)] ,      ! Output
RNOST_INTERMEDIATE = [$DESCRIPTOR(FIXED)] ,      ! Binary intermediate .BRN type file
RNOST_DROP_REC   = [$DESCRIPTOR(FIXED)] ,      ! Drop-record file
```

! Miscellaneous other string descriptors.

```
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$a_datetime     = [$BYTES(12)] ,             ! System date and time
  $OVERLAY( rno$a_datetime )
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$V_2_cross_ref  = [$BITS(2)]      ! Carries state of /CROSSREFERENCE to DOOPTS
  $OVERLAY ( rno$V_2_cross_ref )      !
rno$V_cross_ref    = [$BIT]          ! translate cross references
rno$V_s_cross_ref  = [$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$V_2_und_nosp  = [$BITS(2)] ,         ! underlining non-spacing inline
$OVERLAY ( rno$V_2_und_nosp )           !
rno$V_und_nosp    = [$BIT] ,              ! non-spacing inline
rno$V_s_und_nosp  = [$BIT] ,              ! /NONSPACING explicitly specified

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

```

```

(4 free bits in OPTION3)

```

```

$OVERLAY ( rno$V_option4 )

rno$V_ln01_ital_under = [$BIT] ,
rno$V_ln01_port_land  = [$BIT] ,
rno$V_ln01_header     = [$BIT] ,
rno$V_ln01_load       = [$BIT] ,

rno$V_s_down          = [$BIT] ,
rno$V_s_right         = [$BIT] ,

! Options for LN01 output:
! set = italics, clear = underlining
! set = portrait, clear = landscape
! set = if header wanted in O/P file
! set = loading of fonts is desired

! /DOWN explicitly specified
! /RIGHT explicitly specified

```



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

```
$CONTINUE
```

```
rno$c_change      = [$BYTE] ;           ! Change bar character (RNOSV_CHNG_CHAR set)
rno$c_underline   = [$BYTE] ;           ! Underline character (RNOSV_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
```


