

RRRRRRRRRRRR		UUU		UUU	NNN		NNN	00000000		FFFFFFFFFFFFFF		FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU		UUU	NNN		NNN	00000000		FFFFFFFFFFFFFF		FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU		UUU	NNN		NNN	00000000		FFFFFFFFFFFFFF		FFFFFFFFFFFFFF
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	FFFFFFFFFFFFFF		FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU		UUU	NNN	NNN	NNN	000	000	FFFFFFFFFFFFFF		FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU		UUU	NNN	NNN	NNN	000	000	FFFFFFFFFFFFFF		FFFFFFFFFFFFFF
RRR	RRR	UUU		UUU	NNN	NNNNNN	NNN	000	000	FFF		FFF
RRR	RRR	UUU		UUU	NNN	NNNNNN	NNN	000	000	FFF		FFF
RRR	RRR	UUU		UUU	NNN	NNNNNN	NNN	000	000	FFF		FFF
RRR	RRR	UUU		UUU	NNN	NNN	NNN	000	000	FFF		FFF
RRR	RRR	UUU		UUU	NNN	NNN	NNN	000	000	FFF		FFF
RRR	RRR	UUU		UUU	NNN	NNN	NNN	000	000	FFF		FFF
RRR	RRR	UUUUUUUUUUUUUUUU		UUUUUUUUUUUUUUUU	NNN	NNN	NNN	00000000		FFF		FFF
RRR	RRR	UUUUUUUUUUUUUUUU		UUUUUUUUUUUUUUUU	NNN	NNN	NNN	00000000		FFF		FFF
RRR	RRR	UUUUUUUUUUUUUUUU		UUUUUUUUUUUUUUUU	NNN	NNN	NNN	00000000		FFF		FFF

```

FFFFFFFFF LL      GGGGGGGG TTTTTTTTTT
FFFFFFFFF LL      GGGGGGGG TTTTTTTTTT
FF        LL      GG        TT
FF        LL      GG        TT
FF        LL      GG        TT
FF        LL      GG        TT
FFFFFFFFF LL      GG        TT
FFFFFFFFF LL      GG        TT
FF        LL      GG  GGGGGG TT
FF        LL      GG  GGGGGG TT
FF        LL      GG    GG    TT
FF        LL      GG    GG    TT
FF        LLLLLLLLL GGGGGG TT
FF        LLLLLLLLL GGGGGG TT

```

```

....
....
....
....

```

```

RRRRRRRR EEEEEEEEE QQQQQQ
RRRRRRRR EEEEEEEEE QQQQQQ
RR      RR EE      QQ      QQ
RR      RR EE      QQ      QQ
RR      RR EE      QQ      QQ
RR      RR EE      QQ      QQ
RRRRRRRR EEEEEEEEE QQ      QQ
RRRRRRRR EEEEEEEEE QQ      QQ
RR      RR EE      QQ      QQ
RR      RR EE      QQ      QQ
RR      RR EE      QQ      QQ
RR      RR EE      QQ      QQ
RR      RR EEEEEEEEE QQQQ  QQ
RR      RR EEEEEEEEE QQQQ  QQ

```



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: Flag table: values and status of flag characters.

ENVIRONMENT: Transportable BLISS

AUTHOR: Rich Friday

CREATION DATE: 1978

MODIFIED BY:

- 005 KFA00005 Ken Alden 21-Mar-1983  
For DSRPLUS: eliminated the passthrough flag.
- 004 KFA00004 Ken Alden 15-Mar-1983  
For DSRPLUS: added bits for passthrough flag.
- 003 KAD00003 Keith Dawson 07-Mar-1983  
Global edit of all modules. Updated module names, idents,  
copyright dates. Changed require files to BLISS library.

--

FLIP

%sbt

\$FIE

LITE

MACR

! In the following definitions, 3-character abbreviations are used for the names of the flags:

abbrev- viation	flag	abbrev- viation	flag
COM	Comment	BLD	Bold
SUB	Substitute	SPA	Space
EFO	Endfootnote	IND	Index
CON	Control	HYP	Hyphenate
QUO	Accept (ne Quote)	OVR	Overstrike
UPP	Uppercase	SBX	Subindex
LOW	Lowercase	PER	Period
CAP	Capitalize	BRK	Break
UND	Underline	NPX	No-permute-index

LITERAL FLAG\_COUNT = 18; ! Number of flags.

LITERAL  
 SUB\_FLAG = 0,  
 EFO\_FLAG = 1,  
 CON\_FLAG = 2,  
 QUO\_FLAG = 3,  
 UPP\_FLAG = 4,  
 LOW\_FLAG = 5,  
 CAP\_FLAG = 6,  
 UND\_FLAG = 7,  
 BLD\_FLAG = 8,  
 SPA\_FLAG = 9,  
 IND\_FLAG = 10,  
 HYP\_FLAG = 11,  
 OVR\_FLAG = 12,  
 SBX\_FLAG = 13,  
 COM\_FLAG = 14,  
 PER\_FLAG = 15,  
 BRK\_FLAG = 16,  
 NPX\_FLAG = 17;

LITERAL  
 FLAG\_ENABLED = 0,  
 FLAG\_CHARACTER = 1;

! The FLAG\_TABLE structure does not need to use %UPVAL to be on -11s and VAX,  
 ! because the information there can be saved in single bytes.

STRUCTURE  
 FLAG\_TABLE[I,J;N=FLAG\_COUNT] =  
 [2 \* N \* %UPVAL]  
 (If J eql FLAG\_ENABLED  
 then FLAG\_TABLE + (I \* %UPVAL)  
 else FLAG\_TABLE + (FLAG\_COUNT + I)\*%UPVAL );

MACRO  
 FLGT\_DEFINITION =  
 FLAG\_TABLE[FLAG\_COUNT];

FLIP  
 %sbt  
 \$FIE  
  
 LITE  
  
 MACR



FLGT.REQ;1

16-SEP-1984 16:54:27.<sup>M</sup><sub>2</sub> Page 3

!

End of FLGT.REQ

FLI

Xsb

SFI

LIT

MAC



CONVRT REQ	ECC REQ	FLIPRECS REQ	FNCT REQ	FRMSTK REQ	GETQSC REQ	KWITEM REQ	LSTOPS REQ	NDXCLI REQ	NDXRTY REQ	PHDEF REQ	RUNTAB REQ
DIGIT REQ	FLIRCHRS REQ	FLPREC REQ	FNCT REQ	FSPACK REQ	GNCC REQ	IFSTK REQ	LSTOPS REQ	NDXXPL REQ	OUTOPT REQ	RUNHAN REQ	RUNTAB REQ
DMDEFS REQ	FFDEFS REQ	FOFIL REQ	GCA REQ	GSLUCC REQ	INDEX REQ	LETTER REQ	MAXIMA REQ	NDXLIN REQ	PAGEN REQ	RNODEF REQ	RUNHAN REQ
DSRLIB REQ	FLGT REQ	FOFIL REQ	GCA REQ	HCT REQ	IRAC REQ	LETTER REQ	MSG REQ	NDXPOL REQ	PASS REQ	RNODEF REQ	RUNHAN REQ
FLIRCHRS REQ	FLIRCHRS REQ	FOREC REQ	FOREC REQ	HC REQ	HLC REQ	LODEFS REQ	MSGTXT REQ	NBITS REQ	PDT REQ	RNODEF REQ	RUNHAN REQ
FLIRCHRS REQ	FLIRCHRS REQ	FOREC REQ	FOREC REQ	HC REQ	HLC REQ	LODEFS REQ	MSGTXT REQ	NBITS REQ	OPDEV REQ	RNODEF REQ	RUNHAN REQ