```
RRRRRRRRRRRR        UUU           UUU  NNN          NNN     000000000      FFFFFFFFFFFFFFFF    FFFFFFFFFFFFFFFF
RRRRRRRRRRRR        UUU           UUU  NNN          NNN     000000000      FFFFFFFFFFFFFFFF    FFFFFFFFFFFFFFFF
RRRRRRRRRRRR        UUU           UUU  NNN          NNN     000000000      FFFFFFFFFFFFFFFF    FFFFFFFFFFFFFFFF
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
RRRRRRRRRRRR        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    NNNNN  000      000     FFF                 FFF
RRR     RRR         UUU           UUU  NNN     NNNNN  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       UUUUUUUUUUUUUUUUU  NNN          NNN    000000000      FFF                 FFF
RRR       RRR       UUUUUUUUUUUUUUUUU  NNN          NNN    000000000      FFF                 FFF
RRR       RRR       UUUUUUUUUUUUUUUUU  NNN          NNN    000000000      FFF                 FFF
```

```
DDDDDDD   MM      MM  DDDDDDD   EEEEEEEEEE  FFFFFFFFFF  SSSSSSSS
DDDDDDD   MM      MM  DDDDDDD   EEEEEEEEEE  FFFFFFFFFF  SSSSSSSS
DD     DD MMMM  MMMM  DD     DD EE          FF         SS
DD     DD MMMM  MMMM  DD     DD EE          FF         SS
DD     DD MM  MM  MM  DD     DD EE          FF         SS
DD     DD MM  MM  MM  DD     DD EE          FF         SS
DD     DD MM      MM  DD     DD EEEEEEEE    FFFFFFFF      SSSSSS
DD     DD MM      MM  DD     DD EEEEEEEE    FFFFFFFF      SSSSSS
DD     DD MM      MM  DD     DD EE          FF              SS
DD     DD MM      MM  DD     DD EE          FF              SS
DD     DD MM      MM  DD     DD EE          FF              SS
DDDDDDD   MM      MM  DDDDDDD   EEEEEEEEEE  FF         SSSSSSSS     ....
DDDDDDD   MM      MM  DDDDDDD   EEEEEEEEEE  FF         SSSSSSSS     ....


RRRRRRRR  EEEEEEEEEE    QQQQQQ
RRRRRRRR  EEEEEEEEEE    QQQQQQ
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

! ABSTRACT:
!      Defines literals and macros used in defining, controlling, and
!      accessing the dynamic memory pool.

! ENVIRONMENT:  Transportable BLISS

! AUTHOR:       Rich Friday

! CREATION DATE: 1978

! MODIFIED BY:

!      004      KAD00004          Keith Dawson    07-Mar-1983
!               Global edit of all modules. Updated module names, idents,
!               copyright dates. Changed require files to BLISS library.
!--

!         Structures defining information stored in a dynamic memory pool.
MACRO

        POOL = VECTOR [POOL_CNTRL_SIZE] %,
        PAD  = VECTOR [PAD_CNTRL_SIZE]  %;

LITERAL

        POOL_CNTRL_SIZE = 3,            !Size of POOL control area.
        PAD_CNTRL_SIZE  = 2;            !Size of a Pooled Area Descriptor.
```

```
!        Offsets into pool control area (POOL) and pool area descriptor (PAD).
LITERAL

        POOL_MAX_PADS   = 0,    !Maximum number of PADs that can be accommodated.
        POOL_ACT_PADS   = 1,    !Current number of allocated PADs.
        POOL_ACT_SIZE   = 2;    !Number of BPVALS in pool control area.

LITERAL

        PAD_SIZE        = 0,    !Size of pooled area (BLISS VALUES).
        PAD_ADDRESS     = 1;    !Start of pooled area.

! The GET_SEG_ADDR macro returns the starting address of a segment from the
! specified pool.
MACRO
    GET_SEG_ADDR(AREA,INDEX) =
        BEGIN
        LOCAL
            PADTAB : REF VECTOR;
        PADTAB = .AREA+POOL_CNTRL_SIZE*%UPVAL;
        .PADTAB[PAD_CNTRL_SIZE*(INDEX-1)+PAD_ADDRESS]
        END
    %;

!                       End of DMDEFS.REQ
```

CONVRT REQ · FRMSTK REQ · GETQSC REQ · NDXCLI REQ · NDXRTY REQ

ECC REQ · KWITEM REQ · PHDEF REQ · RUNTAB REQ

FLIPRECS REQ · FNCT REQ · GNCC REQ · IFSTK REQ · LSTOPS REQ · OUTOPT 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