
PPPPPPPPPPPP AAAAAAAAA TTTTTTTTTTTTTTT CCCCCCCCCCC HHH HHH
PPPPPPPPPPPP AAAAAAAAA TTTTTTTTTTTTTTT CCCCCCCCCCC HHH HHH
PPPPPPPPPPPP AAAAAAAAA TTTTTTTTTTTTTTT CCCCCCCCCCC HHH HHH
PPP PPP AAA AAA TTT CCC HHH HHH
PPP PPP AAA AAA TTT CCC HHH HHH
PPP PPP AAA AAA TTT CCC HHH HHH
PPP PPP AAA AAA TTT CCC HHH HHH
PPP PPP AAA AAA TTT CCC HHH HHH
PPPPPPPPPPPP AAA AAA TTT CCC HHHHHHHHHHHHHHH
PPPPPPPPPPPP AAA AAA TTT CCC HHHHHHHHHHHHHHH
PPPPPPPPPPPP AAA AAA TTT CCC HHHHHHHHHHHHHHH
PPP AAAAAAAAAAAAAAA TTT CCC HHH HHH
PPP AAAAAAAAAAAAAAA TTT CCC HHH HHH
PPP AAAAAAAAAAAAAAA TTT CCC HHH HHH
PPP AAA AAA TTT CCC HHH HHH
PPP AAAAAAAAAAAAAAA TTT CCC HHH HHH
PPP AAA AAA TTT CCCCCCCCCCC HHH HHH
PPP AAA AAA TTT CCCCCCCCCCC HHH HHH
PPP AAA AAA TTT CCCCCCCCCCC HHH HHH

FILEID**DATBAS

```
DDDDDDDD      AAAAAA     TTTTTTTTTT  BBBBBBBB      AAAAAA      SSSSSSSS
DDDDDDDD      AAAAAA     TTTTTTTTTT  BBBBBBBB      AAAAAA      SSSSSSSS
DD      DD   AA    AA        TT      BB      BB   AA    AA   SS
DD      DD   AA    AA        TT      BB      BB   AA    AA   SS
DD      DD   AA    AA        TT      BB      BB   AA    AA   SS
DD      DD   AA    AA        TT      BB      BB   AA    AA   SS
DD      DD   AA    AA        TT      BBBBBBBB     AA    AA    SSSSSS
DD      DD   AA    AA        TT      BBBBBBBB     AA    AA    SSSSSS
DD      DD   AAAAAAAAAA      TT      BB      BB   AAAAAAAAAA          SS
DD      DD   AAAAAAAAAA      TT      BB      BB   AAAAAAAAAA          SS
DD      DD   AA    AA        TT      BB      BB   AA    AA            SS      ....
DD      DD   AA    AA        TT      BB      BB   AA    AA            SS      ....
DDDDDDDD     AA    AA        TT      BBBBBBBB     AA    AA   SSSSSSSS       ....
DDDDDDDD     AA    AA        TT      BBBBBBBB     AA    AA   SSSSSSSS       ....
```

```
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'
```



```
++
MODULE: DATBAS.REQ
FACILITY: LINKER
ABSTRACT: DATA BASE COMPILE TIME FORMATS
HISTORY:
        AUTHOR: T.J. PORTER 01-MAR-77
        MODIFICATIONS:
NO.   DATE            PROGRAMMER      PURPOSE
---   ----            ----------      -------
--
```

```
!
!++
! FUNCTIONAL DESCRIPTION:
!
! This is a require file that defines the layout
! (at compile time) of most of the internal data
! structures of the linker. Symbol table entries are
! defined separately.
!--
!
! Define the layout of and accessing macros for the
! file descriptor blocks which forma doubly linked list in
! the order of specification by the user. The FDB contains
! an RMS auxiliary file name block so that the file may be
! opened by file id after the first time. The auxiliary
! file name block contains a descriptor of the resultant file
! name string (after all logical names and defaults have been
! applied by RMS on the first open) so that this complete name
! may be used in error messages and the map. Note however that
! there is also a descriptor of the name that the user
! supplied in the command.
!
BYTEBLOCKFIELDS(FDB,                    ! Names are FDB$X_...
        L_FLINK,4,                      ! Forward link
        L_BLINK,4,                      ! Backward link
        L_OMDLST,4,                     ! Listhead for object module descriptors
                                        ! also used to point to module name list
        W_LIBLSTLNG,2,                  ! Length of the string which is the
                                        ! module name list if this is a library
                                        ! with explicit module extraction
        B_FILFLGS,1,                    ! File specific flags
        Q_USRNAMDSC,8,                  ! String descriptor of the user supplied filename
        A_AUXFNB,NAM$C_BLN);            ! The RMS auxiliary filename block
!
! Define input file flags
!
GLOBAL LITERAL
        !**NOTE BIT 0 RESERVED - SET ALWAYS BY CLI***
        !**     RE-USED AS FLAG IN LIBRARY SEARCHES**
        LNK$S_NEWUDF = 0 : WEAK,        ! A module from library added a new undefined symbol to list
        LNK$S_LIBR = 1 : WEAK,          ! Library flag bit
        LNK$S_LIM = 2 : WEAK,           ! Linkable image file flag
        LNK$S_SELSER = 3 : WEAK,        ! Selective search file
        LNK$S_OPTION = 4 : WEAK,        ! Option file
        LNK$S_DEBUGER = 5 : WEAK,       ! File contains the debugger
        LNK$S_LIBEXTR = 6 : WEAK,       ! Explicit module extraction from library
        LNK$S_LIBSRCH = 7 : WEAK,       ! Library to be searched for undefined symbols
!
        LNK$M_NEWUDF = 1 ^ LNK$S_NEWUDF : WEAK,
        LNK$M_LIBR = 1 ^ LNK$S_LIBR : WEAK,             ! Make the mask
        LNK$M_SELSER = 1 ^ LNK$S_SELSER : WEAK,         ! For selective search
        LNK$M_DEBUGER = 1 ^ LNK$S_DEBUGER : WEAK,       ! File contains the debugger
        LNK$M_LIBEXTR = 1 ^ LNK$S_LIBEXTR : WEAK,       ! Explicit extraction
        LNK$M_LIBSRCH = 1 ^ LNK$S_LIBSRCH : WEAK;       ! Search the library
```

```
!
! Define offsets into a p-section mapping table (appended
!       to module descriptors)
!
GLOBAL LITERAL
        PMT$L_PSCDES = 0 : WEAK,        ! Pointer to p-section descriptor
        PMT$L_MODCON = 1 : WEAK,        ! Pointer to module contribution data block
        PMT$L_SYMLST = PMT$L_MODCON : WEAK,     ! Forward list of prematurely defined symbols
        PMT$C_SIZE   = 8 : WEAK;                ! Size of an entry
!
! Define the layout of an object module
! descriptor and the accessing macros
!
LITERAL
        NPSECTS=12;             ! Initial number of p-sects
BYTEBLOCKFIELDS(OMD,            ! Names are OMD$X_YY...
        L_NXTOMD,4,             ! Link to next in file
        L_ALLOC,4,              ! Module's contribution to memory
                        ! *** NEXT 2 FIELDS MUST BE CONTIGUOUS
        L_MODVBN,4,             ! Virtual block number part
        W_BYTOFF,2,             ! And byte offset part of rfa of a library module
        B_HIPSCT,1,             ! Highest p-sect number
        B_FLAGS,1,              ! Module flags
        B_MAPLNG,1,             ! Length of mapping table
        B_NAMLNG,1,             ! Name length
        T_NAME,SYM$C_MAXLNG,    ! Name field
        AC_PSCMAP,PMT$C_SIZE*NPSECTS);  ! P-Section map table
!
! Macros to access the RFA of a module
!
MACRO
        MODVBN = 0,0,32,0%,             ! Virtual block part
        MODBYTE = 4,0,16,0%;            ! Offset within block
!
! Object module flags
!
GLOBAL LITERAL
        OMD$M_NOPSCT = 1 : SHORT WEAK,                  ! Set until a p-section is seen
        OMD$M_SELSER = LNK$M_SELSER : SHORT WEAK;       ! Set if selective search module
!
! Define the layout of a program section descriptor
! and the accessing macros
!
BYTEBLOCKFIELDS(PSC,            ! Names are PSC$X_YY...
        L_FLINK,4,              ! Forward link
        L_BLINK,4,              ! Backward link
        L_MPCLST,4,             ! Contributing module list
        L_SYMLST,4,             ! Owned relocatable symbol list
        L_BASE,4,               ! Base address
        L_LENGTH,4,             ! Accumulated (if con)/maximum (if ovr) length
        B_ALIGN,1,              ! Allignment of p-section base
        W_FLAGS,2,              ! P-Section flags
        B_NAMLNG,1,             ! P-Section name length
        T_NAME,SYM$C_MAXLNG);   ! P-Section name
!
! Define the layout of a module's p-section contribution data
```

```
!       block and macros to access it.
!
BYTEBLOCKFIELDS(MPC,                    ! Names are MPC$X_YY...
        L_NXTMPC,4,                     ! Forward pointer (singly linked list)
        L_OWNOMD,4,                     ! Pointer to module descriptor
        L_OFFSET,4,                     ! Offset of this contribution from base
        L_LENGTH,4,                     ! Length of this contribution
        B_ALIGN,1);                     ! This contribution's alignment
!
! Define the layout of an image section descriptor
!
BYTEBLOCKFIELDS(ISD,                    ! Names are ISD$X_YY...
        L_NXTISD,4,                     ! Singly linked list
        W_SIZE,2,                       ! Size of this ISD
        W_PAGES,2,                      ! Number of pages in image section
        V_BASVPN,3,                     ! Base virtual page number
        B_PAGFCL,1,                     ! Page fault cluster size
        V_FLAGS,3,                      ! I-Sect control flags
        B_TYPE,1,                       ! Type code
        L_BASVBN,4,                     ! Base virtual block number
        L_IDENT,4,                      ! I-Sect identification
        B_NAMLNG,1,                     ! Length of name
        T_NAME,SYM$C_MAXLNG);           ! I-Sect name.
!
! Define isd flags
!
GLOBAL LITERAL
        ISD$M_GBL     = 1 : WEAK,       ! Global section
        ISD$M_CRF     = 2 : WEAK,       ! Copy on reference
        ISD$M_DZRO    = 4 : WEAK,       ! Demand zero
        ISD$M_WRT     = 8 : WEAK,       ! Writable
        ISD$V_MATCHCTL = 4 : WEAK,      ! Bit offset to match control field
        ISD$C_MATNEV = 0 : WEAK,        ! Match never
        ISD$C_MATALL = 1 : WEAK,        ! Match always
        ISD$C_MATEQU = 2 : WEAK,        ! Match equal
        ISD$C_MATLEQ = 3 : WEAK;        ! Match less or equal
!
! Define the layout of the image header's constant data
! record
!
BYTEBLOCKFIELDS(HDR,                    ! Names are HDR$X_YY...
        W_RECSIZ,2,                     ! Size of this record
        W_HDRBLKS,2,                    ! Number of header blocks
        L_TFRADR1,4,                    ! Transfer address 1
        L_TFRADR2,4,                    ! Transfer address 2
        W_LIDMAJ,2,                     ! Linker ident major
        W_LIDMIN,2,                     ! Linker ident minor
        B_NAMLNG,1,                     ! Length of image name
        T_NAME,SYM$C_MAXLNG);           ! Image name
!
! Some constants of the image header
!
GLOBAL LITERAL
        HDR$C_FILLCHR = 255 : BYTLIT WEAK,      ! Fill character for header blocks
        HDR$C_FILL = 2 : WEAK;                  ! Minimum number of fill bytes per header block
                                        ! ** MUST EQUAL WIDTH OF ISD SIZE FIELD
```

```
!
! Define the linker version array. Its content is written to image
! header and checked by the image activator.
!
GENBLOCKFIELDS(LID,                     ! Names are LID$X_YY...
        MAJOR,2,                        ! Major ident (version)
        MINOR,2);                       ! Minor ident (alteration)
```