```
VVV         VVV  MMM         MMM     SSSSSSSSSSSS  LLL                 IIIIIIIII     BBBBBBBBBBBB
VVV         VVV  MMM         MMM     SSSSSSSSSSSS  LLL                 IIIIIIIII     BBBBBBBBBBBB
VVV         VVV  MMM         MMM     SSSSSSSSSSSS  LLL                 IIIIIIIII     BBBBBBBBBBBB
VVV         VVV  MMMMMM   MMMMMM  SSS              LLL                    III        BBB         BBB
VVV         VVV  MMMMMM   MMMMMM  SSS              LLL                    III        BBB         BBB
VVV         VVV  MMMMMM   MMMMMM  SSS              LLL                    III        BBB         BBB
VVV         VVV  MMM   MMM   MMM  SSS              LLL                    III        BBB         BBB
VVV         VVV  MMM   MMM   MMM  SSS              LLL                    III        BBB         BBB
VVV         VVV  MMM   MMM   MMM  SSS              LLL                    III        BBB         BBB
VVV         VVV  MMM         MMM     SSSSSSSSS     LLL                    III        BBBBBBBBBBBB
VVV         VVV  MMM         MMM     SSSSSSSSS     LLL                    III        BBBBBBBBBBBB
VVV         VVV  MMM         MMM     SSSSSSSSS     LLL                    III        BBBBBBBBBBBB
VVV         VVV  MMM         MMM              SSS  LLL                    III        BBB         BBB
VVV         VVV  MMM         MMM              SSS  LLL                    III        BBB         BBB
VVV         VVV  MMM         MMM              SSS  LLL                    III        BBB         BBB
   VVV   VVV     MMM         MMM              SSS  LLL                    III        BBB         BBB
   VVV   VVV     MMM         MMM              SSS  LLL                    III        BBB         BBB
   VVV   VVV     MMM         MMM              SSS  LLL                    III        BBB         BBB
      VVV        MMM         MMM  SSSSSSSSSSSS     LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB
      VVV        MMM         MMM  SSSSSSSSSSSS     LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB
      VVV        MMM         MMM  SSSSSSSSSSSS     LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB
```

```
**FILE**ID**STARDEFFL                                   B  8



  SSSSSSSS  TTTTTTTTTT    AAAAAA    RRRRRRRR    DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF  FFFFFFFFFF  LL
  SSSSSSSS  TTTTTTTTTT    AAAAAA    RRRRRRRR    DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF  FFFFFFFFFF  LL
SS              TT      AA      AA  RR      RR  DD      DD  EE          FF          FF          LL
SS              TT      AA      AA  RR      RR  DD      DD  EE          FF          FF          LL
SS              TT      AA      AA  RR      RR  DD      DD  EE          FF          FF          LL
SS              TT      AA      AA  RR      RR  DD      DD  EE          FF          FF          LL
  SSSSSS        TT      AA      AA  RRRRRRRR    DD      DD  EEEEEEEE    FFFFFFFF    FFFFFFFF    LL
  SSSSSS        TT      AA      AA  RRRRRRRR    DD      DD  EEEEEEEE    FFFFFFFF    FFFFFFFF    LL
        SS      TT      AAAAAAAAAA  RR  RR      DD      DD  EE          FF          FF          LL
        SS      TT      AAAAAAAAAA  RR  RR      DD      DD  EE          FF          FF          LL
        SS      TT      AA      AA  RR    RR    DD      DD  EE          FF          FF          LL              ....
        SS      TT      AA      AA  RR    RR    DD      DD  EE          FF          FF          LL              ....
SSSSSSSS        TT      AA      AA  RR      RR  DDDDDDDD    EEEEEEEEEE  FF          FF          LLLLLLLLLL      ....
SSSSSSSS        TT      AA      AA  RR      RR  DDDDDDDD    EEEEEEEEEE  FF          FF          LLLLLLLLLL      ....


  SSSSSSSS  DDDDDDDD    LL
  SSSSSSSS  DDDDDDDD    LL
SS          DD      DD  LL
SS          DD      DD  LL
SS          DD      DD  LL
SS          DD      DD  LL
  SSSSSS    DD      DD  LL
  SSSSSS    DD      DD  LL
        SS  DD      DD  LL
        SS  DD      DD  LL
        SS  DD      DD  LL
        SS  DD      DD  LL
SSSSSSSS    DDDDDDDD    LLLLLLLLLL
SSSSSSSS    DDDDDDDD    LLLLLLLLLL
```

```
{ STARCEFFL.SDL - system user interface definitions
{
{ 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 ANC 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  SHOULC  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
{*  CORPORA'''ON.                                                           *
{*                                                                          *
{*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELiABILITY OF ITS   *
{*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
{*                                                                          *
{*                                                                          *
{****************************************************************************

{++
{
{ FACILITY:  VAX/VMS System Macro Libraries
{
{ ABSTRACT:
{
{       This file contains the SDL source for all user visible operating
{       system interfaces from F to L.
{
{ ENVIRONMENT:
{
{       n/a
{
{--
{
{
{ AUTHOR:  The VMS Group          CREATION DATE:  1-Aug-1976
{
{ MODIFIED BY:
{
{       V03-073 MSH0067         Michael S. Harvey       19-Jul-1984
{               Add some JPI items that had been overlooked.
{
{       V03-072 ACG0438         Andrew C. Goldstein,    17-Jul-1984  19:14
{               Add flush cache ACP control function
{
{       V03-071 SRB0130         Steve Beckhardt         17-May-1984
{               Added LCK$M_NODLCKBLK to $LCKDEF
{
```

```
{       V03-070 EMD0093         Ellen M. Dusseault      30-Apr-1984
{               Rearrange IO modifier bits for disk and tape devices
{               so that they remain in the same places as they were
{               for V3.  I had moved them previously so that they
{               were not consistent with V3.
{
{       V03-069 TMH0069         Tim Halvorsen           12-Apr-1984
{               Fix V03-068 so that it compiles - the new bits caused
{               the structure to exceed 32 bits.
{
{       V03-068 EAD0144         Elliott A. Drayton      12-Apr-1984
{               Added LP$_SIXELS and LP$_BITMAPPED for printers.
{
{       V03-067 CDS0002         Christian D. Saether    11-Apr-1984
{               Remove FIB$L_ACCLKID field.
{
{       V03-066 HWS0057         Harold Schultz          11-Apr-1984
{               Add JPI$_MASTER_PID item.
{
{       V03-065 SRB0120         Steve Beckhardt         9-Apr-1984
{               Added LCK$M_NODLCKWT to $LCKDEF
{
{       V03-064 RNG0064         Rod Gamache             09-Apr-1984
{               Change name of $LKIDEF LKI$B_STATE to LKI$B_QUEUE, and
(               change name of LKI$L_REMSYSTEM to LKI$L_REMSYSID.
{
{       V03-063 EMD0075         Ellen M. Dusseault      07-Apr-1984
{               Add modifier, IO$M_ENCRYPT, to $IODEF and rearrange
{               the sequence of disk and tape modifiers to make room
{               for it.
{
{       V03-062 TMK0005         Todd M. Katz            29-Mar-1984
{               Add LNM$C_TABNAMLEN, the maximum size of a logical name table
{               name or a logical name contained within a directory table, and
{               LNM$C_MAXDEPTH, the maximum logical name recursion depth, to
{               $LNMDEF.
{
{       V03-061 RNG0061         Rod Gamache             24-Mar-1984
{               Add LKI$L_REMLKID and LKI$L_REMSYSTEM to $LKIDEF.
{
{       V03-060 CWH3060         CW Hobbs                20-Mar-1984
{               Add JPI$_PROC_INDEX item
{
{       V03-059 LMP0209         L. Mark Pilant,         9-Mar-1984  9:46
{               Add an ACL return status field to the FIB.
{
{       V03-058 MSH0010         Michael S. Harvey       16-Feb-1984
{               Add JPI$_TABLENAME
{               Add JPI$_CREPRC_FLAGS
{               Add JPI$_UAF_FLAGS
{
{       V03-057 LMP0189         L. Mark Pilant,         6-Feb-1984  13:45
{               Add FIB$V_DIRACL to make the ACP propagate the ACL on
{               directory files.
{
{       V03-056 EMD0044         Ellen M. Dusseault      1-Feb-1984
{               Add bitmask for fallback which is a new line printer
{               characteristic in LPDEF.
{
{       V03-055 LMP0186         L. Mark Pilant,         1-Feb-1984  9:48
{               Add FIB$V_PROPAGATE to allow the propagation of attributes
{               on an ACP enter as well as a create.
{
{       V03-054 MMD0227         Meg Dumont,     26-Jan-1984  16:55
{               Add the code FIB$C_CLSEREXCP to the FIB$W_CNTRLFUNC field
{               This will allow users to clear serious exceptions from
{               the tape drive when running in USER HANDLING fo EOT mode.
{
{       V03-053 ACG0385         Andrew C. Goldstein,    29-Dec-1983  15:45
{               Add JPI$_JOBTYPE
{
{       V03-052 ROW0265         Ralph O. Weber          27-DEC-1983
{               Add IO$M_MSCP (for IO$_SENSEMODE) and IO$M_FORCERR (for
{               IO$_WRITExBLK); two new I/O function modifiers required by the
{               disk class driver for various elements of shadowed volumes
{               support.
{
{       V03-051 TMK0004         Todd M. Katz            19-Dec-1983
{               Delete LNM$V_GROUP and LNM$V_SYSTEM from $LNMDEF.
{
{       V03-050 RSH0087         R. Scott Hanna          07-Dec-1983
{               Move $KGBDEF from SYSDEFFL.SDL to STARDEFFL.SDL
{
{       V03-049 ROW0250         Ralph O. Weber          10-NOV-1983
{               Add IO$M_EXPRESS modifier for DUDRIVER devices.  Eventually,
{               setting this bit will cause requests to be delivered with the
{               MSCP$M_MD_EXPRS modifier.  Add the IO$M_SHADOW modifier for
{               DUDRIVER devices.  This modifier will be used by various
{               commands related to shadowed devices support.  Add
{               IO$M_NORSWAIT modifier for mailbox requests.  Eventually,
{               setting this will prevent mailbox driver from placing a
{               process in resource wait (something like a per request disable
{               resource wait mode).  This is required by the job controller
{               to prevent malicious users from hanging the job controller
{               in MWAIT.
{
{       V03-048 ACG0359         Andrew C. Goldstein,    16-Sep-1983  14:41
{               Fix name of ALT_GRANTED bit in FIB
{
{       V03-047 ACG0354         Andrew C. Goldstein,    13-Sep-1983  14:55
{               Add alternate access mask to FIB
{
{       V03-046 KFH0005         Ken Henderson            8 Sep 1983
{               Add to $JPIDEF: JPI$_MODE, JPI$_CLINAME,
{               and JPI$K_xxx. Remove from $JPIDEF:
{               individual bits of PCB$L_STS because they're
{               defined in SYSDEF, not STARDEF.
{
{       V03-045 MIR0084         Michael I. Rosenblum    24-Aug-1983
{               Add io$m_autoxof_ena and dis to iodef.
{
{       V03-044 RNG0044         Rod N. Gamache          22-Aug-1983
{               Add some field definitions and LKI$_REMLKID to $LKIDEF.
{
{       V03-043 TMK0003         Todd M. Katz            15-Aug-1983
{               Add LNM$V_GROUP and LNM$V_SYSTEM to $LNMDEF.
{
{       V03-042 JSV0364         Joost Verhofstad        27-JUL-1983
{               Add IO$V_GETMINFO
{
{       V03-041 MKL0128         Mary Kay Lyons          24-Jul-1983
{               Add IO$M_JNL_INIT for cluster journals.
{
{       V03-030 WMC0030         Wayne Cardoza           05-Jul-1983
{               Chained item lists for GETJPI.
{
{       V03-029 WMC0029         Wayne Cardoza           22-JUN-1983
{               New parent item code for logical names.
{
{       V03-038 MKL0115         Mary Kay Lyons          22-Jun-1983
{               Add IO$M_UPDADL for cluster control journals.
{
{       V03-037 MIR0051         Michael I. Rosenblum    22-Jun-1983
{               Add IO$M_BREAKTHRU to the io deffinitions
{               Add tb and truncate to lpdef.
{
{       V03-036 MKL0099         Mary Kay Lyons          08-Jun-1983
{               Add REPOSITION function modifier to $IODEF.
{
{       V03-035 MKL0091         Mary Kay Lyons          25-May-1983
{               Add IO$M_ACKWRITE to $IODEF.
{
{       V03-034 SRB0087         Steve Beckhardt         24-May-1983
{               Added new flag bits RECOVER and PROTECT to $LCKDEF.
{
{       V03-033 LY0372          Larry Yetto             24-MAY-1983 14:21:05
{               Add new function codes and modifiers for journal
{               fail over
{
{       V03-032 DMW4028         DMWalp                  24-May-1983
{               Added LNMB_ADDR item to $LNMDEF, for internal use by
{               MTL and mailbox UCB
{
{       V03-031 STJ3098         Steven T. Jeffreys,     03-May-1983
{               - Identify each group of device-specific I/O function modifiers.
{               - Shuffle some disk and tape function code modifiers around.
{                       IO$x_INTSKIP (bit 8) replaced by IO$x_SWAP (bit 10)
{                       IO$x_CECYL (bit 10) replaced by IO$x_ERASE
{                 The result is that disk and tapes have a common defintion
{                 of IO$x_ERASE.
{
{       V03-030 MIR0041         Michael I. Rosenblum    29-Apr-1983
{               Define new terminal write modifier IO$M_NEWLINE
{
{       V03-029 RKS0029         RICK SPITZ              29-APR-1983
{               ADD SEVERAL NEW MODIFIERS TO THE TERMINAL DRIVER
{               SET MODE FUNCTION TO SUPPORT CONNECT AND SPAWN FEATURES
{
```

```
{       V03-028 TMK0002         Todd M. Katz            27-Apr-1983
{               Make several changes to $LNMDEF. Change LNM$V_NOT_EXIST to
{               LNM$V_EXISTS, and delete LNM$_DUMP_NAMES.
{
{       V03-027 TMK0001         Todd M. Katz            14-Apr-1983
{               Make several changes to $LNMDEF. Change LNM$V_SUPERSEDE to LNM$V_CREATE_IF
{               and add LNM$V_NOT_EXIST.
{
{       V03-026 LMP0098         L. Mark Pilant,         8-Apr-1983  12:53
{               Add a field for the FIB to specify the agents access mode.
{
{       V03-025 SRB0072         Steve Beckhardt         24-Mar-1983
{               Added the following flags to $LCKDEF:  CANCEL, CVTSYS,
{               and INVVALBLK.
{
{       V03-024 KFH0004         Ken Henderson           23 Mar 1983
{               Added PHDFLAGS to $JPIDEF.
{
{       V03-023 DMW4027         DMWalp                  23-Mar-1983
{               Changed $LNMDEF itemlist to start at 1, not 0
{
{       V03-022 SRB0069         Steve Beckhardt         9-Mar-1983
{               Added NOQUOTA bit to $LCKDEF.
{
{       V03-021 JSV0184         Joost Verhofstad        09-MAR-1983
{               Add NEWVERSION modifiers.
{
{       V03-020 KFH0003         Ken Henderson            1 Mar 1983
{               Added item-codes for each bit in
{               PCB$L_STS (in $JPIDEF).
{
{       V03-019 KFH0002         Ken Henderson           10 Feb 1983
{               Add JPI$_MSGMASK to $JPIDEF.
{
{       V03-018 JSV0140         Joost Verhofstad        08-FEB-1983
{               Add IO$_NEWVERSION and IO$M_NEWVERSION
{
{       V03-017 RNG0017         Rod N. Gamache           8-Feb-1983
{               Add $LKIDEF.
{
{       V03-016 CDS0001         Christian D. Saether     7-Jan-1983
{               Add FIB$L_ACCLKID field to fib.
{
{       V03-015 STJ3049         Steven T. Jeffreys       3-Jan-1983
{               Added IO$M_ERASE i/o function code modifier.
{
{       V03-014 DMW4014         DMWalp                   1-Dec-1982
{               Added $LNMDEF
{
{       V03-013 ACG0303         Andrew C. Goldstein,    9-Dec-1982  16:05
{               Add FILL attribute to extraneous field names
{
{       V03-012         KFH0001                 Ken Henderson           24 Nov 1982
{               Add $FDLDEF section
{
{       V03-011         JSV0085                 Joost Verhofstad        20-Oct-1982
{               Add IO$M_CREAJNLDIR and IO$V_CREAJNLDIR
{
{       V03-010         RLRSEREX                Robert L. Rappaport     25-Aug-1982
{               Add IO$M_CLSEREXCP I/O function modifier for tape functions.
{
{       V03-009         JSV0044                 Joost Verhofstad        12-Aug-1982
{               Change IO$_RUCONTROL from physical to virtual IO
{               and remove IO$_ENDRU1 and IO$_ENDRU2.
{
{        V03-008        JSV0030                Joost Verhofstad        27-Jul-1982
{                Add some RUCONTROL function modifiers and
{                remove obsolete ones that were never used
{
{        V03-007        LMP0039                L. Mark Pilant,               15-Jul-1982  10:36
{                Put the FIB ACL context in the correct place.
{
{        V03-006        KBT0076                Keith B. Thompson        6-Jul-1982
{                Add a warning about changing the size of the fib
{
{        V03-005        LMP0036                L. Mark Pilant,               29-Jun-1982 11:00
{                Add FIB field to contain ACL context.  This is used when
{                reading the entire ACL for a file.
{
{        V03-004        JSV005                Joost Verhofstad       10-Jun-1982
{                Add function codes and modifiers for journaling
{
{        V03-003        STJ0311                Steven T. Jeffreys        2-Jun-1982
{                Add REMOUNT modifier for ACP control functions.
{
{        V03-002        LJK0157                Lawrence J. Kenah        7-Apr-1982
{                Add JPI$_IMAGECOUNT for LIB$SPAWN's benefit
{
{        V03-001        MMD0001                Meg Dumont,       5-Apr-1982  14:31
{                Add function modifier definitions to $IODEF for XWDRIVER
{
```

```
module $FDLDEF;
/*+
/* FDL CALL INTERFACE CONTROL FLAGS
/*-

aggregate FDLDEF union prefix FDL$;
    FDLDEF_BITS structure fill;
        SIGNAL bitfield mask;                  /* SIGNAL ERRORS, DON'T RETURN
        FDL_STRING bitfield mask;              /* MAIN FDL SPEC IS A CHAR STRING
        DEFAULT_STRING bitfield mask;          /* DEFAULT FDL SPEC IS A CHAR STRING
        FULL_OUTPUT bitfield mask;             /* PRODUCE A 'COMPLETE' FDL SPEC
        $CALLBACK bitfield mask;               /* USED BY EDF ON INPUT (DEC ONLY)
    end FDLDEF_BITS;
end FDLDEF;

end_module $FDLDEF;
```

```
module $FIBDEF;
/*----
/*
/* LAYOUT OF THE FILE IDENTIFICATION BLOCK (FIB)
/*
/*****
/*
/*        NOTE:        If the size of the FIB is changed the following must be changed
/*                to reflect the change:
/*
/*                In Module:        [RMS.SRC]RMSFWADEF.SDL
/*
/*                Field:                FWA$T_FIBBUF
/*
/*                Constant:        FWA$C_FIBLEN
/*
/*                Both the field and constant must be GEQ to the size of
/*                the FIB, i.e. FIB$C_LENGTH. FIB length is currently 64.
/*
/*****
/*

aggregate FIBDEF structure prefix FIB$;
    ACCTL_OVERLAY union fill;
        ACCTL longword unsigned;                         /* ACCESS CONTROL BITS
        ACCTL_BITS0 structure fill;
            NOWRITE bitfield mask;                      /* NO OTHER WRITERS
            DLOCK bitfield mask;                        /* ENABLE DEACCESS LOCK
            BLK_LOCK bitfield mask;                     /* ENABLE RMS-11 BLOCK LOCKING
            FILL_1 bitfield fill prefix FIBDEF tag $$;  /* UNUSED
            SPOOL bitfield mask;                        /* SPOOL FILE ON CLOSE
            WRITECK bitfield mask;                      /* ENABLE WRITE CHECK
            SEQONLY bitfield mask;                      /* SEQUENTIAL ONLY ACCESS
            FILL_2 bitfield fill prefix FIBDEF tag $$;  /* SPARE
            WRITE bitfield mask;                        /* WRITE ACCESS
            READCK bitfield mask;                       /* ENABLE READ CHECK
            NOREAD bitfield mask;                       /* NO OTHER READERS
            NOTRUNC bitfield mask;                      /* FILE MAY NOT BE TRUNCATED
            FILL_3 bitfield length 4 fill prefix FIBDEF tag $$;/* SPARE
                                                        /* THE HIGH 8 BITS CANNOT BE COPIED
                                                        /* INTO THE ACCESS MODE WORD IN THE WINDOW
            EXECUTE bitfield mask;                      /* ACCESS FOR EXECUTE (USE EXECUTE PROTECTION)
            PRSRV_ATR bitfield mask;                    /* PRESERVE ORIGINAL ATTRIBUTES OF FILE
            RMSLOCK bitfield mask;                      /* OPEN WITH RMS RECORD LOCKING
            WRITETHRU bitfield mask;                    /* FORCE CACHE WRITE-THROUGH ON OPERATION
            NOLOCK bitfield mask;                       /* OVERRIDE ACCESS INTERLOCKS
            NORECORD bitfield mask;                     /* DO NOT RECORD FILE ACCESS
            FILL_4 bitfield length 2 fill prefix FIBDEF tag $$;/* SPARE
        end ACCTL_BITS0;
        ACCTL_BITS1 structure fill;
            FILL_5 bitfield length 3 fill prefix FIBDEF tag $$;
            REWIND bitfield mask;                       /* REWIND TAPE
            CURPOS bitfield mask;                       /* CREATE AT CURRENT TAPE POSITION
            FILL_6 bitfield fill prefix FIBDEF tag $$;
            UPDATE bitfield mask;                       /* UPDATE MODE (POSITION TO START OF FILE)
        end ACCTL_BITS1;
        ACCTL_FIELDS2 structure fill;
            FILL_13 byte dimension 3 fill prefix FIBDEF tag $$;
            WSIZE byte;                                 /* WINDOW SIZE
        end ACCTL_FIELDS2;
    end ACCTL_OVERLAY;
    FID_OVERLAY union fill;
        FID word unsigned dimension 3;                  /* FILE ID
        constant ACCDATA equals . prefix FIB$ tag K;    /* ABOVE DATA NECESSARY FOR ACCESS
        constant ACCDATA equals . prefix FIB$ tag C;    /* ABOVE DATA NECESSARY FOR ACCESS
        FID_FIELDS structure fill;
            FID_NUM word unsigned;                      /* FILE NUMBER
            FID_SEQ word unsigned;                      /* FILE SEQUENCE NUMBER
            FID_RVN_OVERLAY union fill;
                FID_RVN word unsigned;                  /* RELATIVE VOLUME NUMBER
                FID_RVN_FIELDS structure fill;
                    FID_RVN byte unsigned;              /* SHORT FORM RVN
                    FID_NMX byte unsigned;              /* EXTENDED FILE NUMBER
                end FID_RVN_FIELDS;
            end FID_RVN_OVERLAY;
        end FID_FIELDS;
    end FID_OVERLAY;
    DID_OVERLAY union fill;
        DID word unsigned dimension 3;                  /* DIRECTORY ID
        DID_FIELDS structure fill;
            DID_NUM word unsigned;                      /* FILE NUMBER
            DID_SEQ word unsigned;                      /* FILE SEQUENCE NUMBER
            DID_RVN_OVERLAY union fill;
                DID_RVN word unsigned;                  /* RELATIVE VOLUME NUMBER
                DID_RVN_FIELDS structure fill;
                    DID_RVN byte unsigned;              /* SHORT FORM RVN
                    DID_NMX byte unsigned;              /* EXTENDED FILE NUMBER
                end DID_RVN_FIELDS;
            end DID_RVN_OVERLAY;
        end DID_FIELDS;
    end DID_OVERLAY;
    WCC longword unsigned;                              /* WILD CARD CONTEXT
    NMCTL_OVERLAY union fill;
        NMCTL word unsigned;                            /* NAME CONTROL BITS
        constant DIRDATA equals . prefix FIB$ tag K;    /* ABOVE DATA NECESSARY FOR DIRECTORY OPS
        constant DIRDATA equals . prefix FIB$ tag C;    /* ABOVE DATA NECESSARY FOR DIRECTORY OPS
        NMCTL_BITS structure fill;
            FILL_7 bitfield length 3 fill prefix FIBDEF tag $$;
            ALLVER bitfield mask;                       /* MATCH ALL VERSIONS
            ALLTYP bitfield mask;                       /* MATCH ALL TYPES
            ALLNAM bitfield mask;                       /* MATCH ALL NAMES
            FILL_8 bitfield length 2 fill prefix FIBDEF tag $$;
            WILD bitfield mask;                         /* WILD CARDS IN FILE NAME
            NEWVER bitfield mask;                       /* MAXIMIZE VERSION NUMBER
            SUPERSEDE bitfield mask;                    /* SUPERSEDE EXISTING FILE
            FINDFID bitfield mask;                      /* SEARCH FOR FILE ID
            FILL_9 bitfield length 2 fill prefix FIBDEF tag $$;
            LOWVER bitfield mask;                       /* LOWER VERSION OF FILE EXISTS
            HIGHVER bitfield mask;                      /* HIGHER VERSION OF FILE EXISTS
        end NMCTL_BITS;
    end NMCTL_OVERLAY;
    EXCTL_OVERLAY union fill;
```

```
        EXCTL word unsigned;                            /* EXTEND CONTROL
        EXCTL_BITS structure fill;
            ALCON bitfield mask;                        /* ALLOCATE CONTIGUOUS
            ALCONB bitfield mask;                       /* CONTIGUOUS BEST EFFORT
            FILCON bitfield mask;                       /* MARK FILE CONTIGUOUS
            ALDEF bitfield mask;                        /* ALLOCATE DEFAULT AMOUNT
            ALLOCATR bitfield mask;                     /* PLACEMENT DATA PRESENT IN ATTRIBUTE LIST
            FILL_10 bitfield length 2 fill prefix FIBDEF tag $$;
            EXTEND bitfield mask;                       /* ENABLE EXTENSION
            TRUNC bitfield mask;                        /* ENABLE TRUNCATION
            NOHDREXT bitfield mask;                     /* INHIBIT EXTENSION HEADERS
            MARKBAD bitfield mask;                      /* MARK BLOCKS BAD
            FILL_11 bitfield length 4 fill prefix FIBDEF tag $$;
            NOCHARGE bitfield mask;                     /* DON'T CHARGE DISKQUOTA
        end EXCTL_BITS;
    end EXCTL_OVERLAY;
    EXSZ longword unsigned;                             /* EXTEND SIZE
    EXVBN longword unsigned;                            /* EXTENSION VBN
    constant EXTDATA equals . prefix FIB$ tag K;        /* ABOVE NECESSARY FOR BASIC FILE EXTENSION
    constant EXTDATA equals . prefix FIB$ tag C;        /* ABOVE NECESSARY FOR BASIC FILE EXTENSION
    ALOPTS_OVERLAY union fill;
        ALOPTS byte unsigned;                           /* ALLOCATION OPTIONS
        ALOPTS_BITS structure fill;
            EXACT bitfield mask;                        /* EXACT PLACEMENT REQUIRED
            ONCYL bitfield mask;                        /* PUT ALLOCATION ON ONE CYLINDER
        end ALOPTS_BITS;
    end ALOPTS_OVERLAY;
    ALALIGN byte unsigned;                              /* ALLOCATION ALIGNMENT
    constant CYL        equals 1  prefix FIB tag $C;    /* CYLINDER ADDRESS SPECIFIED
    constant LBN        equals 2  prefix FIB tag $C;    /* LBN SPECIFIED
    constant VBN        equals 3  prefix FIB tag $C;    /* PROXIMATE VBN SPECIFIED
    constant RFI        equals 4  prefix FIB tag $C;    /* RELATED FILE ID SPECIFIED
    ALLOC_OVERLAY union fill;
        ALLOC word unsigned dimension 5;                /* ALLOCATION LOCATION
        constant ALCDATA equals . prefix FIB$ tag K;    /* ABOVE DATA NECESSARY FOR PLACEMENT
        constant ALCDATA equals . prefix FIB$ tag C;    /* ABOVE DATA NECESSARY FOR PLACEMENT
        ALLOC_FIELDS structure fill;
            LOC_FID_OVERLAY union fill;
                LOC_FID word unsigned dimension 3;      /* RELATED FILE ID
                LOC_FID_FIELDS structure fill;
                    LOC_NUM word unsigned;              /* RELATED FILE NUMBER
                    LOC_SEQ word unsigned;              /* FILE SEQUENCE NUMBER
                    LOC_RVN_OVERLAY union fill;
                        LOC_RVN word unsigned;          /* RELATED RVN
                        LOC_RVN_FIELDS structure fill;
                            LOC_RVN byte unsigned;      /* SHORT FORM RVN
                            LOC_NMX byte unsigned;      /* EXTENDED FILE NUMBER
                        end LOC_RVN_FIELDS;
                    end LOC_RVN_OVERLAY;
                end LOC_FID_FIELDS;
            end LOC_FID_OVERLAY;
            LOC_ADDR longword unsigned;                 /* LOCATION ADDRESS (VBN, LBN, CYL)
        end ALLOC_FIELDS;
    end ALLOC_OVERLAY;
    VERLIMIT word unsigned;                             /* DIRECTORY ENTRY VERSION LIMIT
    AGENT_MODE byte unsigned;                           /* AGENTS ACCESS MODE
    FILL_12 byte fill prefix FIBDEF tag $$;             /* SPARE
    ACLCTX longword unsigned;                           /* ACL CONTEXT FOR READ
    ACL_STATUS longword unsigned;                       /* RETURN STATUS FROM ACL OPERATION
    STATUS_OVERLAY union fill;
        STATUS longword unsigned;                       /* GENERAL STATUS LONGWORD
        STATUS_BITS structure fill;
            ALT_REQ bitfield mask;                      /* ALTERNATE ACCESS REQUIRED
            ALT_GRANTED bitfield mask;                  /* ALTERNATE ACCESS GRANTED
            DIRACL bitfield mask;                       /* DO DIRECTORY ACL PROPAGATION
            PROPAGATE bitfield mask;                    /* DO PROPAGATION ON ENTER
        end STATUS_BITS;
    end STATUS_OVERLAY;
    ALT_ACCESS longword unsigned;                       /* ALTERNATE ACCESS MASK
    constant "LENGTH" equals . prefix FIB$ tag K;
    constant "LENGTH" equals . prefix FIB$ tag C;

end FIBDEF;

aggregate FIBDEF1 structure prefix FIB$;
    FILL_14 byte dimension 22 fill prefix FIBDEF tag $$;
    CNTRLFUNC word unsigned;                            /* ACP CONTROL FUNCTION
/* DEFINE ACP CONTROL FUNCTION CODES
/*
    constant(
          REWINDVOL                                     /* REWIND VOLUME SET
        , POSEND                                        /* POSITION TO END OF VOLUME SET
        , NEXTVOL                                       /* FORCE NEXT VOLUME
        , SPACE                                         /* SPACE MAGNETIC TAPE
        , ILLEGAL                                       /*
        , REWINDFIL                                     /* REWIND FILE
        , LOCK_VOL                                      /* LOCK VOLUME AGAINST ALLOCATION
        , UNLK_VOL                                      /* UNLOCK VOLUME
                                                        /* QUOTA FILE OPERATIONS
        , ENA_QUOTA                                     /* ENABLE QUOTA FILE
        , DSA_QUOTA                                     /* DISABLE QUOTA FILE
        , ADD_QUOTA                                     /* ADD QUOTA FILE ENTRY
        , EXA_QUOTA                                     /* EXAMINE QUOTA FILE ENTRY
        , MOD_QUOTA                                     /* MODIFY QUOTA FILE ENTRY
        , REM_QUOTA                                     /* REMOVE QUOTA FILE ENTRY
        , USEREOT                                       /* ENABLE USER END OF TAPE HANDLING
        , REMAP                                         /* REMAP FILE WINDOW
        , CLSEREXCP                                     /* ALLOW THE USER TO CLEAR A SERIOUS EXCP FROM A TAPE DRIVE
        , FLUSH_CACHE                                   /* FLUSH SELECTED CACHE
        ) equals 1 increment 1  prefix FIB tag $C;
    CNTRLVAL_OVERLAY union fill;
        CNTRLVAL longword unsigned;                     /* ACP CONTROL FUNCTION VALUE PARAMETER
        constant MTALEN equals . prefix FIB$ tag K;     /* LENGTH OF MTAACP DATA
        constant MTALEN equals . prefix FIB$ tag C;     /* LENGTH OF MTAACP DATA
/*
/* CACHE IDENTIFIER CODES FOR FLUSH_CACHE
/*
        constant (
          FID_CACHE
        , EXTENT_CACHE
        , QUOTA_CACHE
        ) equals 1 increment 1 prefix FIB tag $C;
```

```
/*
/* CONTROL BITS FOR QUOTA FILE OPERATIONS
/*
        CNTRLVAL_BITS structure fill;
            ALL_MEM bitfield mask;                          /* MATCH ALL MEMBER NUMBERS
            ALL_GRP bitfield mask;                          /* MATCH ALL GROUP NUMBERS
            MOD_USE bitfield mask;                          /* MODIFY USAGE DATA
            MOD_PERM bitfield mask;                         /* MODIFY PERMANENT QUOTA
            MOD_OVER bitfield mask;                         /* MODIFY OVERDRAFT LIMIT
        end CNTRLVAL_BITS;


    end CNTRLVAL_OVERLAY;
end FIBDEF1;

end_module $FIBDEF;
```

```
module $FIDDEF;
/*
/* STRUCTURE OF A FILE ID (FID)
/*


aggregate FIDDEF structure prefix FID$;
    NUM word unsigned;                                  /* FILE NUMBER
    SEQ word unsigned;                                  /* FILE SEQUENCE NUMBER
    RVN_OVERLAY union fill;
        RVN word unsigned;                              /* RELATIVE VOLUME NUMBER
        constant "LENGTH" equals . prefix FID$ tag K;
        constant "LENGTH" equals . prefix FID$ tag C;
        RVN_FIELDS structure fill;
            RVN byte unsigned;                          /* BYTE FORM OF RVN
            NMX byte unsigned;                          /* FILE NUMBER EXTENSION
                                                        /* FILE ID'S OF THE RESERVED FILES
            constant(
                  INDEXF                                /* INDEX FILE
                , BITMAP                                /* STORAGE MAP FILE
                , BADBLK                                /* BAD BLOCK FILE
                , MFD                                   /* MASTER FILE DIRECTORY
                , CORIMG                                /* CORE IMAGE FILE
                , VOLSET                                /* VOLUME SET LIST FILE
                , CONTIN                                /* STANRARD CONTINUATION FILE
                , BACKUP                                /* BACKUP LOG FILE
                , BADLOG                                /* BAD BLOCK LOG FILE
                , FREFIL                                /* FREE SPACE FILE
                ) equals 1 increment 1  prefix FID tag $C;

        end RVN_FIELDS;
    end RVN_OVERLAY;
end FIDDEF;

end_module $FIDDEF;
```

```
module $IACDEF;
/*+
/* IMAGE ACTIVATION CONTROL FLAGS
/*-


aggregate IACDEF  union prefix IAC$;
    IACDEF_BITS structure fill;
        NOACT bitfield mask;                               /*DO NOT ACTIVATE THE IMAGE (FOR INSTALL)
        WRITABLE bitfield mask;                            /*MAKE IMAGE FILE WRITABLE
        SHAREABLE bitfield mask;                           /*ACT. SHAREABLE IMAGE FOR EXECUTABLE IMG
        PRIVILEGE bitfield mask;                           /*ACT. SHARE. IMG FOR PRIV EXECUTABLE IMG
        MERGE bitfield mask;                               /*ACT 2ND EXECUTABLE IMG INTO ADR SPACE
        EXPREG bitfield mask;                              /*MAP IMAGE INTO NEXT FREE VA SPACE
        P1MERGE bitfield mask;                             /*P1 MERGED ACTIVATION (LEGAL INPUT FLAG)
        FILL_1 bitfield fill prefix IACDEF tag $$;         /*SPARE
        LASTCLU bitfield mask;                             /*LAST CLUSTER FLAG
        LIM bitfield mask;                                 /*LINKABLE IMAGE
        RETRY bitfield mask;                               /*RETRY IMAGE ACTIVATION
        NOCMKRNL bitfield mask;                            /*SHUT OFF CMKRNL,CMEXEC-SYSVER DIFF
        SEQDEVLOD bitfield mask;                           /*LOADING FROM SEQUENTIAL DEVICE (NET)
        XLINKER bitfield mask;                             /*CROSS LINKER FORMAT
        KP_RESHDR bitfield mask;                           /*MAKE IMAGE HEADER RESIDENT
        IS_RESHDR bitfield mask;                           /*IMAGE HEADER IS RESIDENT
        NOTSHARED bitfield mask;                           /*DO NOT SET IS_SHARED IN KFI ENTRY
        GBLCLUSTR bitfield mask;                           /*CURRENTLY PROCESSING GBL ISD CLUSTER
        SHMIDENT bitfield mask;                            /*SHARED MEMORY IDENT USED FOR GBL SEC
        NOCOPY bitfield mask;                              /*NO PRIVATE COPY OF SECTION IN EXEC IMG
        P1MERG_P0 bitfield mask;                           /*P1 MERGED ACTIVATION WITH P0 ADDRESS
                                                           /* RANGE (INTERNAL FLAG ONLY)
        SETVECTOR bitfield mask;                           /*SIGNAL ATLERNATE ENTRY TO SET VECTORS
    end IACDEF_BITS;

    constant LARGEST    equals 6  prefix IAC tag $C;      /*LARGEST FLAG CALLER MAY SPECIFY
end IACDEF;

end_module $IACDEF;
```

```
module $IODEF;
/*+
/* I/O FUNCTION CODE DEFINITIONS
/*-


/*
/* *** START PHYSICAL I/O FUNCTION CODES ***
/*

constant NOP              equals 0  prefix IO tag $;       /*NO OPERATION
constant UNLOAD           equals 1  prefix IO tag $;       /*UNLOAD DRIVE
constant LOADMCODE        equals 1  prefix IO tag $;       /*LOAD MICROCODE
constant SEEK             equals 2  prefix IO tag $;       /*SEEK CYLINDER
constant SPACEFILE        equals 2  prefix IO tag $;       /*SPACE FILES
constant STARTMPROC       equals 2  prefix IO tag $;       /*START MICROPROCESSOR
constant RECAL            equals 3  prefix IO tag $;       /*RECALIBRATE DRIVE
constant STOP             equals 3  prefix IO tag $;       /*STOP
constant SNDJNLMSG        equals 3  prefix IO tag $;       /*GENERIC SEND JOURNAL CI MESSAGE
constant DRVCLR           equals 4  prefix IO tag $;       /*DRIVE CLEAR
constant INITIALIZE       equals 4  prefix IO tag $;       /*INITIALIZE
constant RELEASE          equals 5  prefix IO tag $;       /*RELEASE PORT
constant SETCLOCKP        equals 5  prefix IO tag $;       /*SET CLOCK (PHYSICAL)
constant OFFSET           equals 6  prefix IO tag $;       /*OFFSET READ HEADS
constant ERASETAPE        equals 6  prefix IO tag $;       /*ERASE TAPE
constant STARTDATAP       equals 6  prefix IO tag $;       /*START DATA TRANSFER (PHYSICAL)
constant RETCENTER        equals 7  prefix IO tag $;       /*RETURN TO CENTERLINE
constant QSTOP            equals 7  prefix IO tag $;       /*QUEUE STOP REQUEST
constant PACKACK          equals 8  prefix IO tag $;       /*PACK ACKNOWLEDGE
constant SEARCH           equals 9  prefix IO tag $;       /*SEARCH FOR SECTOR
constant SPACERECORD      equals 9  prefix IO tag $;       /*SPACE RECORDS
constant WRITECHECK       equals 10  prefix IO tag $;      /*WRITE CHECK DATA
constant WRITEPBLK        equals 11  prefix IO tag $;      /*WRITE PHYSICAL BLOCK
constant READPBLK         equals 12  prefix IO tag $;      /*READ PHYSICAL BLOCK
constant WRITEHEAD        equals 13  prefix IO tag $;      /*WRITE HEADER AND DATA
constant JNLDRVREQ        equals 13  prefix IO tag $;      /*JOURNAL DRIVER JNLACP REQUEST
constant RDSTATS          equals 13  prefix IO tag $;      /*READ STATISTICS
constant READHEAD         equals 14  prefix IO tag $;      /*READ HEADER AND DATA
constant WRITETRACKD      equals 15  prefix IO tag $;      /*WRITE TRACK DATA
constant READTRACKD       equals 16  prefix IO tag $;      /*READ TRACK DATA
constant AVAILABLE        equals 17  prefix IO tag $;      /*AVAILABLE (DISK AND TAPE CLASS)
constant DSE              equals 21  prefix IO tag $;      /*DATA SECURITY ERASE (AND REWIND)
constant REREADN          equals 22  prefix IO tag $;      /*REREAD NEXT
constant REREADP          equals 23  prefix IO tag $;      /*REREAD PREVIOUS
constani WRITERET         equals 24  prefix IO tag $;      /*WRITE RETRY
constant WRITECHECKH      equals 24  prefix IO tag $;      /*WRITE CHECK HEADER AND DATA
constant READPRESET       equals 25  prefix IO tag $;      /*READIN ?RESET
constant STARTSPNDL       equals 25  prefix IO tag $;      /*START SPINDLE
constant SETCHAR          equals 26  prefix IO tag $;      /*SET CHARACTERISTICS
constant SENSECHAR        equals 27  prefix IO tag $;      /*SENSE TAPE CHARACTERISTICS
constant WRITEMARK        equals 28  prefix IO tag $;      /*WRITE TAPE MARK
constant WRTTMKR          equals 29  prefix IO tag $;      /*WRITE TAPE MARK RETRY
constant DIAGNOSE         equals 29  prefix IO tag $;      /*DIAGNOSE
constant FLUSH            equals 29  prefix IO tag $;      /*FLUSH JOURNAL BUFFERS
constant FORMAT           equals 30  prefix IO tag $;      /*FORMAT
constant CLEAN            equals 30  prefix IO tag $;      /*CLEAN TAPE
constant PHYSICAL       equals 31  prefix IO tag $;     /*HIGHEST PHYSICAL I/O FUNCTION CODE

/*
/* *** START LOGICAL I/O FUNCTION CODES ***
/*

constant WRITELBLK      equals 32  prefix IO tag $;     /*WRITE LOGICAL BLOCK
constant READLBLK       equals 33  prefix IO tag $;     /*READ LOGICAL BLOCK
constant REWINDOFF      equals 34  prefix IO tag $;     /*REWIND AND SET OFFLINE
constant SETMODE        equals 35  prefix IO tag $;     /*SET MODE
constant REWIND         equals 36  prefix IO tag $;     /*REWIND TAPE
constant SKIPFILE       equals 37  prefix IO tag $;     /*SKIP FILES
constant SKIPRECORD     equals 38  prefix IO tag $;     /*SKIP RECORDS
constant SENSEMODE      equals 39  prefix IO tag $;     /*SENSE TAPE MODE
constant WRITEOF        equals 40  prefix IO tag $;     /*WRITE END OF FILE
constant LOGICAL        equals 47  prefix IO tag $;     /*HIGHEST LOGICAL I/O FUNCTION CODE

/*
/* *** START VIRTUAL I/O FUNCTION CODES
/*

constant WRITEVBLK      equals 48  prefix IO tag $;     /*WRITE VIRTUAL BLOCK
constant READVBLK       equals 49  prefix IO tag $;     /*READ VIRTUAL BLOCK
constant ACCESS         equals 50  prefix IO tag $;     /*ACCESS FILE
constant CREATE         equals 51  prefix IO tag $;     /*CREATE FILE
constant DEACCESS       equals 52  prefix IO tag $;     /*DEACCESS FILE
constant DELETE         equals 53  prefix IO tag $;     /*DELETE FILE
constant MODIFY         equals 54  prefix IO tag $;     /*MODIFY FILE
constant NETCONTROL     equals 54  prefix IO tag $;     /*X25 NETWORK CONTROL FUNCTION
constant READPROMPT     equals 55  prefix IO tag $;     /*READ TERMINAL WITH PROMPT
constant SETCLOCK       equals 55  prefix IO tag $;     /*SET CLOCK
constant FORCE          equals 55  prefix IO tag $;     /*JOURNALING BUFFER FLUSH
constant ACPCONTROL     equals 56  prefix IO tag $;     /*MISCELLANEOUS ACP CONTROL
constant STARTDATA      equals 56  prefix IO tag $;     /*START DATA
constant MOUNT          equals 57  prefix IO tag $;     /*MOUNT VOLUME
constant TTYREADALL     equals 58  prefix IO tag $;     /* TERMINAL READ PASSALL
constant RUCONTROL      equals 58  prefix IO tag $;     /*RECOVERY UNIT JOURNAL CONTROL FUNCTION
constant TTYREADPALL    equals 59  prefix IO tag $;     /* TERM READ W/PROMPT PASSALL
constant CONINTREAD     equals 60  prefix IO tag $;     /* Connect to interrupt readonly
constant READINIT       equals 60  prefix IO tag $;     /* JOURNAL READ INITIALIZE
constant CONINTWRITE    equals 61  prefix IO tag $;     /* Connect to interrupt with write
constant NEWVERSION     equals 61  prefix IO tag $;     /* Create new journal file version
constant VIRTUAL        equals 63  prefix IO tag $;     /*HIGHEST VIRTUAL I/O FUNCTION

/*
/* FUNCTION MODIFIER BIT DEFINITIONS
/*


aggregate IODEF  union prefix IO$;

    /* Basic I/O function code/modifiers structure

    #fcode_size = 6;

    FCODE_STRUCTURE structure fill;
```

```
        FCODE bitfield mask length #fcode_size;          /* Function Code Field
        FMODIFIERS bitfield mask length 16-#fcode_size;  /* Function Modifiers Field
    end FCODE_STRUCTURE;

    /* General disk and tape function code modifiers

    DISK_TAPE_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        reserved_for_device_dependent bitfield length 4 fill;
        ERASE bitfield mask;                             /* Erase data
        INHERLOG bitfield mask;                          /* Inhibit error logging
        reserved_for_device_dependent bitfield length 1 fill;
        ENCRYPT bitfield mask;                           /* Encryption
        DATACHECK bitfield mask;                         /* Write check data after transfer
        INHRETRY bitfield mask;                          /* Inhibit error retry
    end DISK_TAPE_MODIFIERS;

    /* General disk function code modifiers

    DISK_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        reserved_for_device_dependent bitfield length 6 fill;
        INHSEEK bitfield mask;                           /* Inhibit implied seek on physical functions
    end DISK_MODIFIERS;

    /* General tape function code modifiers

    TAPE_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        REVERSE bitfield mask;                           /* Reverse operation
        NOWAIT bitfield mask;                            /* No wait for rewind to complete
        reserved_for_device_dependent bitfield length 4 fill;
        INHEXTGAP bitfield mask;                         /* Inhibit extended inter-record gap
    end TAPE_MODIFIERS;

    /* DU, disk class driver function modifier bits

    DU_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        SHADOW bitfield mask;                            /* Do shadowing, as appropriate
        EXPRESS bitfield mask;                           /* Use MSCP express modifier
        FORCERR bitfield mask;                           /* Cause FORCED ERROR flag to be set
    end DU_MODIFIERS;

    /* DR driver function modifier bits.

    DR_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        COMMOD bitfield mask;                            /* Diagnostic command
        MOVETRACKD bitfield mask;                        /* Move track descriptor
        DIAGNOSTIC bitfield mask;                        /* Diagnostic function
        SKPSECINH bitfield mask;                         /* Skip sector inhibit
    end DR_MODIFIERS;

    /* DY driver function modifier bits.
    DY_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        DELDATA bitfield mask;                              /* Write deleted data mark
    end DY_MODIFIERS;

    /* DD driver (TU58) function modifier bits.

    DD_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        NOMRSP bitfield mask;                               /* Do not use MRSP for this operation
    end DD_MODIFIERS;

    /* Magnetic tape I/O function modifier bits for the TS11.

    TS11_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        reserved_for_device_dependent bitfield length 2 fill;
        SWAP bitfield mask;                                 /* Swap byte (TS11)
        OPPOSITE bitfield mask;                             /* Opposite bit for rereads (TS11)
    end TS11_MODIFIERS;

    /* TU driver function modifier bits

    TU_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        reserved_for_device_dependent bitfield length 3 fill;
        CLSEREXCP bitfield mask;                            /* Clear serious exception condition
    end TU_MODIFIERS;

    /* ACP function modifier bits.

    ACP_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        ACCESS bitfield mask;                               /* Access file
        CREATE bitfield mask;                               /* Create file
        DELETE bitfield mask;                               /* Delete file
        MOUNT bitfield mask;                                /* Mount volume
        DMOUNT bitfield mask;                               /* Dismount volume
        REMOUNT bitfield mask;                              /* Remount volume
    end ACP_MODIFIERS;

    /* CR (card reader) driver function modifier bits.

    CR_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        BINARY bitfield mask;                               /* Binary read
        PACKED bitfield mask;                               /* Packed read
    end CR_MODIFIERS;

    /* MB (mailbox) driver function modifier bits.

    MB_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        NOW bitfield mask;                                  /* Do not wait for operation complete
        READATTN bitfield mask;                             /* Request ast on waiting reader
        WRTATTN bitfield mask;                              /* Request ast on waiting writer
        SETPROT bitfield mask;                              /* Set volume protection
        NORSWAIT bitfield mask;                             /* Do not allow resource waits
    end MB_MODIFIERS;

    /* TT driver READ function modifier bits.

    TT_READ_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        NOECHO bitfield mask;                               /* Noecho
        TIMED bitfield mask;                                /* Timed
        CVTLOW bitfield mask;                               /* Convert lower case
        NOFILTR bitfield mask;                              /* No filter
        DSABLMBX bitfield mask;                             /* Disable mailbox
        PURGE bitfield mask;                                /* Purge typeahead
        TRMNOECHO bitfield mask;                            /* Terminators are not echoed
        REFRESH bitfield mask;                              /* Control-R interrupted read
        ESCAPE bitfield mask;                               /* Terminate read on escape sequence
        EXTEND bitfield mask;                               /* Used by alternate class drivers
    end TT_READ_MODIFIERS;

    /* TT driver WRITE function modifier bits.

    TT_WRITE_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        CANCTRLO bitfield mask;                             /* Cancel control O
        ENABLMBX bitfield mask;                             /* Enable mailbox
        NOFORMAT bitfield mask;                             /* Do not format output
        BREAKTHRU bitfield mask;                            /* Broadcast I/O
        NEWLINE bitfield mask;                              /* Output a newline
                                                            {  NOTE: bit 13 is refresh
    end TT_WRITE_MODIFIERS;

    /* TT driver SENSEMODE function modifier bits.

    TT_SENSEMODE_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        TYPEAHDCNT bitfield mask;                           /* Sense typeahead information
        reserve_rd_modem bitfield length 1 fill;            {  Reserve RD_MODEM bit
    end TT_SENSEMODE_MODIFIERS;

    /* TT driver SETMODE subfunction modifier bits.

    TT_SETMODE_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        MAINT bitfield mask;                                /* Enable maint sub modifiers
        CTRLYAST bitfield mask;                             /* Set control Y AST
        CTRLCAST bitfield mask;                             /* Set Control C
        HANGUP bitfield mask;                               /* Set mode and hang up line
        OUTBAND bitfield mask;                              /* Set out of band AST
        TT_CONNECT BITFIELD MASK;                           /* Connect to detached terminal
        TT_DISCON BITFIELD MASK;                            /* Disconnect detached terminal
        TT_PROCESS BITFIELD MASK;                           /* Define controlling process
        BRDCST BITFIELD MASK;                               /* Define broadcast mask
    end TT_SETMODE_MODIFIERS;

    /* TT driver MAINTIANCE subfunction submodifier bits.
    TT_MAINT_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        reserve_maint_escape bitfield length 1 fill;      {  Reserve maintenance escape
        LOOP bitfield mask;                               /* Diagnostic loopback
        UNLOOP bitfield mask;                             /* Diagnostic reset loopback
        LINE_OFF bitfield mask;                           /* Disable line
        SET_MODEM bitfield mask;                          /* Diagnostic modem control
        LINE_ON bitfield mask;                            /* Enable line
        LOOP_EXT bitfield mask;                           /* Diagnostic external loopback
        AUTXOF_ENA bitfield mask;                         /* Enable auto XOFF
        AUTXOF_DIS bitfield mask;                         /* Disable auto XOFF
        reserve_int_disable bitfield length 1 fill;       {  Reserve internal disable bit
    end TT_MAINT_MODIFIERS;

    /* TT driver out-of-band modifier bits.

    TT_OUTOFBAND_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        reserve_maint_escape bitfield length 5 fill;      {  Reserve maintenance escape
        INCLUDE bitfield mask;                            /* Include character in stream
        TT_ABORT BITFIELD MASK;                           /* Abort current I/O
    end TT_OUTOFBAND_MODIFIERS;

    /* Network WRITE VIRTUAL function modifier bits.

    NET_WRITE_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        INTERRUPT bitfield mask;                          /* Interrupt message
    end NET_WRITE_MODIFIERS;

    /* Network ACCESS/DEACCESS function modifier bits.

    NET_ACCDEA_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        reserved bitfield length 2 fill;
        ABORT bitfield mask;                              /* Disconnect abort/connect reject
        SYNCH bitfield mask;                              /* Synchronous disconnect
    end NET_ACCDEA_MODIFIERS;

    /* DATALINK driver SETMODE subfunction modifier bits.

    DLINK_SETMODE_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        STARTUP bitfield mask;                            /* Start protocol
        SHUTDOWN bitfield mask;                           /* Stop protocol
        ATTNAST bitfield mask;                            /* Attention AST
        CTRL bitfield mask;                               /* Controller (ie. not station) function
        reserve_set_modem bitfield length 1 fill;         {  Reserve set modem bit
    end DLINK_SETMODE_MODIFIERS;

    /* DATALINK driver SENSEMODE subfunction modifier bits.

    DLINK_SENSEMODE_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        RD_MEM bitfield mask;                             /* Read device memory
        RD_MODEM bitfield mask;                         /* Read modem status
        RD_COUNT bitfield mask;                         /* Read counters
        reserve_ctrl bitfield length 1 fill;            {  Controller function (SETMODE compatible)
        CLR_COUNT bitfield mask;                        /* Clear counters
    end DLINK_SENSEMODE_MODIFIERS;

    constant SRRUNOUT   equals 0;                       /* Send or rcv until cnt runout
    constant PTPBSC     equals 8192;                    /* Point to point BSC control
    constant LOOPTEST   equals 57344;                   /* Loop test

    /* X25 driver WRITEBLK function modifier bits.

    X25_WRITE_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        MORE bitfield mask;                             /* More data follows (M-bit)
        QUALIFIED bitfield mask;                        /* Use qualified sub-channel (Q-bit)
    end X25_WRITE_MODIFIERS;

    /* X25 driver ACCESS function modifier bits.

    X25_ACCESS_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        REDIRECT bitfield mask;                         /* Redirect virtual call
        ACCEPT bitfield mask;                           /* Accept virtual call
    end X25_ACCESS_MODIFIERS;

    /* LPA-11 driver STARTDATA modifier bits.

    LPA_START_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        SETEVF bitfield mask;                           /* Set event flag
    end LPA_START_MODIFIERS;

    /* XA (DR11-W) driver function modifier bits.

    XA_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        'WORD" bitfield mask;                           /* Word (interrupt)/block (DMA) mode specifier
        filler bitfield length 2 fill;
        SETFNCT bitfield mask;                          /* Set "FNCT" bits in device CSR
        DATAPATH bitfield mask;                         /* Change UBA datapath (direct/buffered)
        RESET bitfield mask;                            /* Device reset specifier
        CYCLE bitfield mask;                            /* Set "cycle" bit in device CSR
    end XA_MODIFIERS;

    /* 3271 driver function modifier bits.

    IBM3271_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        filler bitfield length 2 fill;
        SETCUADR bitfield mask;                         /* Set a new CU address
        SETBSIZE bitfield mask;                         /* Set a new maximum buffer size
        SETPOOLSZ bitfield mask;                        /* Set a new pool size
        SETENQCNT bitfield mask;                        /* Set a new ENQ threshold
        CLEAR bitfield mask;                            /* Zero status counters
        LPBEXT bitfield mask;                           /* Loopback is external loopback
        LPBINT bitfield mask;                           /* Loopback is internal loopback (DUP11)
        READCSR bitfield mask;                          /* Read CSRs on DUP11
    end IBM3271_MODIFIERS;

    /* XW driver function modifier bits.

    XW_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        NOCTSWAIT bitfield mask;                        /* Diagnostic function
        SLAVLOOP bitfield mask;                         /* Diagnostic function
        NODSRWAIT bitfield mask;                        /* Do not wait for DSR (diag)
        MAINTLOOP bitfield mask;                        /* Internal maint loop
        LASTBLOCK bitfield mask;                        /* Last block of message
        filler bitfield length 1 fill;
        INTCLOCK bitfield mask;                         /* Internal clock
    end XW_MODIFIERS;

    /* CJF - write journal modifiers

    CJF_WRITE_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        FORCE bitfield mask;                            /* Force out journal entries to media
        CNTRLENTRY bitfield mask;                       /* Write control entry
        MULTIPLE bitfield mask;                         /* Gather write - (multiple buffers)
        ENDRU1 bitfield mask;                           /* Phase 1 end-recovery-unit
        ENDRU2 bitfield mask;                           /* Phase 2 end-recovery-unit
        DIO bitfield mask;                              /* Direct I/O
        ADDFLTR bitfield mask;                          /* Add filter
        DELFLTR bitfield mask;                          /* Delete filter
        NEWVERSION bitfield mask;                       /* Create new version
    end CJF_WRITE_MODIFIERS;

    /* CJF - ACPCONTROL modifiers

    CJF_ACPCTRL_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        ALLJNLDEV bitfield mask;                        /* Allocate journal device
        MNTJNLDEV bitfield mask;                        /* Mount journal device
        DALLJNLDEV bitfield mask;                       /* Deallocate journal device
        DMNTJNLDEV bitfield mask;                       /* Dismount journal device
        CREAJNLDIR bitfield mask;                       /* Create journal directory
        REMASTER bitfield mask;                         /* Remaster in progress
    end CJF_ACPCTRL_MODIFIERS;

    /* CJF - modifiers for internal JNLdriver requests for JNLACP

    CJF_INT_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        CANCELIO bitfield mask;                         /* Cancel I/O to tape
        STOPSP bitfield mask;                           /* Stop spooling to spool file
        STARTSP bitfield mask;                          /* Start spooling to spool file
        SYNCHCAN bitfield mask;                         /* Synchronize with the cance
        SLVCRUCB bitfield mask;                         /* Slave UCB create
        SLVDELUCB bitfield mask;                        /* Slave UCB delete
        REPOSITION bitfield mask;                       /* Reposition
    end CJF_INT_MODIFIERS;
    /* CJF - RUCONTROL modifiers

    CJF_RUDTRL_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        RUIDLIST bitfield mask;                          /* Get RU-ID list for journal
        RUJLIST bitfield mask;                           /* Get RU-journal list
        RUEBIT bitfield mask;                            /* Set RU state for journal (RUE bit)
    end CJF_RUCTRL_MODIFIERS;

    /* CJF - NEWVERSION modifiers

    CJF_NEWVERS_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        CONNECT bitfield mask;                           /* Connect old version
        DISCONNECT bitfield mask;                        /* Disconnect old version(s)
    end CJF_NEWVERS_MODIFIERS;

    /* CJF - CREATE modifiers

    CJF_CREATE_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        CREMASTER bitfield mask;                         /* Create remaster request
    end CJF_CREATE_MODIFIERS;

    /* CJF - send journal message modifiers

    CJF_SENDMSG_MODIFIERS structure fill;
        fcode_fill bitfield length #fcode_size fill;
        INQWRTBUF bitfield mask;                         /* Inquire for write buffer info
        RESUBWRT bitfield mask;                          /* Resubmit write buffer
        WRTFOVRCPL bitfield mask;                        /* Write failover complete
        GETPART bitfield mask;                           /* Get partial entry
        ACKWRITE bitfield mask;                          /* ACK journal write
        JNL_INIT bitfield mask;                          /* Get CJF startup info
        GETMINFO bitfield mask;                          /* Get CJF master info
    end CJF_SENDMSG_MODIFIERS;

end IODEF;

end_module $IODEF;
```

```
module $JPIDEF;
/*+
/*
/* Get Job Process Information Data Identifier Definitions
/*
/* **** NOTE ****
/*
/*          New items must always be added to the END of each item list
/*          (preceeding the ENDlistname item) so that users will not have
/*          to relink.
/*-



                                                              /* DEFINE TABLE NUMBERS

                                                              /* STRUCTURE CODES MUST START AT 1
constant ADRTYPE         equals 1  prefix JPI tag $C;    /* ITEM IS ADDRESS OF DATA
constant CTLTYPE         equals 2  prefix JPI tag $C;    /* ITEM IS IN CONTROL REGION
constant PCBTYPE         equals 3  prefix JPI tag $C;    /* ITEM IN PCB
constant PHDTYPE         equals 4  prefix JPI tag $C;    /* ITEM IN PHD
constant PCBFLDTYPE      equals 5  prefix JPI tag $C;    /* ITEM IS BIT FIELD
constant PHDFLDTYPE      equals 6  prefix JPI tag $C;    /* ITEM IS BIT FIELD
constant JIBTYPE         equals 7  prefix JPI tag $C;    /* ITEM IN JIB (SUBTYPE)
constant MAXSTRUC        equals 6  prefix JPI tag $C;    /* TOTAL OF 6 TABLES
constant LISTEND         equals 0  prefix JPI tag $C;    /* DEFINE ITMLST TERMINATOR
constant CHAIN           equals -1  prefix JPI tag $;    /* CHAIN INDICATOR
/*
/* NOTE!! The following constants match values in $JIBDEF.
/*
constant OTHER           equals 0  prefix JPI tag $K;    /* JPI$_Mode values
constant NETWORK         equals 1  prefix JPI tag $K;    /* JPI$_Mode values
constant BATCH           equals 2  prefix JPI tag $K;    /* JPI$_Mode values
constant INTERACTIVE     equals 3  prefix JPI tag $K;    /* JPI$_Mode values
constant DETACHED        equals 0  prefix JPI tag $K;    /* JPI$_JOBTYPE values
constant LOCAL           equals 3  prefix JPI tag $K;    /* JPI$_JOBTYPE values
constant DIALUP          equals 4  prefix JPI tag $K;    /* JPI$_JOBTYPE values
constant REMOTE          equals 5  prefix JPI tag $K;    /* JPI$_JOBTYPE values
                                                              /* (also BATCH & INTERACTIVE)
                         /* ITEM IDENTIFIERS FOR PCB


constant(
      ASTACT             /* ACCESS MODES WITH ACTIVE ASTS
    , ASTEN              /* ACCESS MODES WITH ASTS ENABLED
    , PRI                /* CURRENT PROCESS PRIORITY
    , OWNER              /* PID OF CREATOR PROCESS
    , UIC                /* UIC OF PROCESS
    , STS                /* PROCESS STATUS
    , STATE              /* PROCESS STATE
    , MEM                /* MEMBER FIELD OF UIC
    , GRP                /* GROUP FIELD OF UIC
    , PRIB               /* PROCESS BASE PRIORITY
    , APTCNT             /* ACTIVE PAGE TABLE COUNT
    , TMBU               /* TERMINATION MAILBOX UNIT
    , GPGCNT             /* GLOBAL PAGE COUNT IN WORKING SET
    , PPGCNT             /* PROCESS PAGE COUNT IN WORKING SET
    , ASTCNT             /* AST COUNT REMAINING
    , BIOCNT             /* BUFFERED I/O COUNT REMAINING
    , BIOLM              /* BUFFERED I/O LIMIT
    , BYTCNT             /* BYTE COUNT REMAINING FOR BUFFERED I/O
    , DIOCNT             /* DIRECT I/O COUNT REMAINING
    , DIOLM              /* DIRECT I/O COUNT LIMIT
    , FILCNT             /* COUNT REMAINING OF OPEN FILES
    , TQCNT              /* COUNT REMAINING OF TIMER QUEUE ENTRIES
    , EFWM               /* EVENT FLAG WAIT MASK
    , EFCS               /* LOCAL EVENT FLAGS 0-31
    , EFCU               /* LOCAL EVENT FLAGS 32-63
    , PID                /* PROCESS IDENTIFICATION
    , BYTLM              /* BUFFERED I/O BYTE COUNT LIMIT
    , PRCCNT             /* SUBPROCESS COUNT
    , PRCNAM             /* PROCESS NAME
    , TERMINAL           /* LOGIN TERMINAL NAME
    , JOBPRCCNT          /* TOTAL COUNT OF SUBPROCESSES IN A JOB
    , ENQCNT             /* ENQUEUE COUNT REMAINING
    , ENQLM              /* ENQUEUE COUNT LIMIT
    , SWPFILLOC          /* SWAP FILE BACKING STORE ADDRESS
    , MODE               /* MODE VALUE (FROM STS) - DEFINED ABOVE
    , JOBTYPE            /* JOB TYPE CODE
    , PROC_INDEX         /* PROCESS INDEX
    , MASTER_PID         /* MASTER PROCESS PID
/* ADD ITEM-CODES BEFORE THIS COMMENT
    , LASTPCB            /* MAX INDEX IN PCB TABLE
    ) equals JPI$C_PCBTYPE@8 increment 1  prefix JPI tag $;

                         /* ITEM IDENTIFIERS FOR PHD

constant(
      CURPRIV            /* CURRENT PROCESS PRIVILEGE MASK
    , WSAUTH             /* AUTHORIZED WORKING SET SIZE
    , WSQUOTA            /* QUOTA ON WORKING SET SIZE
    , DFWSCNT            /* DEFAULT WORKING SET SIZE
    , FREP0VA            /* FIRST FREE VIRTUAL ADDR. AT END OF P0 SPACE
    , FREP1VA            /* FIRST FREE ADDR. AT END OF P1 SPACE
    , DFPFC              /* DEFAULT PAGE FAULT CLUSTER
    , CPUTIM             /* ACCUMULATED CPU TIME
    , PRCLM              /* SUBPROCESS QUOTA
    , ASTLM              /* AST LIMIT
    , PAGEFLTS           /* COUNT OF PAGE FAULTS
    , DIRIO              /* PROCESS DIRECT I/O OPERATIONS
    , BUFIO              /* PROCESS BUFFERED I/O OPERATIONS
    , CPULIM             /* LIMIT ON CPUTIM FOR PROCESS
    , PGFLQUOTA          /* MAX. VIRTUAL PAGE COUNT
    , FILLM              /* OPEN FILE LIMIT
    , TQLM               /* TIMER QUEUE LIMIT
    , WSSIZE             /* CURRENT WORKING SET SIZE
    , AUTHPRIV           /* AUTHORIZED PRIVILEGE MASK
    , IMAGPRIV           /* INSTALLED IMAGE PRIVILEGE MASK
    , PAGFILCNT          /* PAGES CHARGED TO PAGE FILE QUOTA
    , FREPTECNT          /* ROOM FOR EXPANSION OF P0 OR P1 SPACE
    , WSEXTENT           /* EXTENT OF WORKING SET SIZE
    , WSAUTHEXT           /* MAX EXTENT OF WORKING SET SIZE
    , AUTHPRI             /* AUTHORIZED PRIORITY FOR $SETPRI
    , PAGFILLOC           /* PAGE FILE BACKING STORE ADDRESS
    , IMAGECOUNT          /* IMAGE COUNTER (CLOCKED BY RUNDOWN)
    , PHDFLAGS            /* PROCESS HEADER FLAGS WORD
/* ADD ITEM-CODES BEFORE THIS COMMENT
    , LASTPHD             /* MAX INDEX IN PROCESS HEADER
    ) equals JPI$C_PHDTYPE@8 increment 1  prefix JPI tag $;


                          /* ITEM IDENTIFIERS FOR CONTROL REGION

constant(
      VIRTPEAK            /* PEAK VIRTUAL SIZE
    , WSPEAK              /* PEAK WORKING SET SIZE
    , USERNAME            /* USERNAME STRING
    , ACCOUNT             /* ACCOUNT NAME STRING
    , PROCPRIV            /* PROCESS PRIVILEGE MASK
    , VOLUMES             /* VOLUMES MOUNTED
    , LOGINTIM            /* TIME OF LOGIN OR PROCESS CREATION
    , IMAGNAME            /* CURRENT IMAGE FILE NAME
    , SITESPEC            /* PER-PROCESS SITE-SPECIFIC CELL
    , MSGMASK             /* PROCESS DEFAULT MESSAGE FLAGS
    , CLINAME             /* COMMAND LANGUAGE INTERPRETER NAME
    , TABLENAME           /* COMMAND LANGUAGE INTERPRETER TABLE NAME
    , CREPRC_FLAGS        /* $CREPRC FLAGS USED TO CREATE THIS PROCESS
    , UAF_FLAGS           /* FLAGS FROM UAF RECORD
    , MAXDETACH           /* MAXIMUM DETACHED PROCESSES FOR SINGLE USER
    , MAXJOBS             /* MAXIMUM ACTIVE PROCESSES FOR SINGLE USER
    , SHRFILLM            /* MAXIMUM OPEN SHARED FILES
/* ADD ITEM-CODES BEFORE THIS COMMENT
    , LASTCTL             /* MAX INDEX IN CONTROL REGION
    ) equals JPI$C_CTLTYPE@8 increment 1  prefix JPI tag $;


                          /* ITEM IDENTIFIERS FOR ADDRESS DATA

constant(
      EXCVEC              /* ADDRESS OF "EXCEPTION VECTOR" VECTOR
    , FINALEXC            /* ADDRESS OF LAST CHANCE EXCEPTION VECTORS
/* ADD ITEM-CODES BEFORE THIS COMMENT
    , LASTADR             /* MAX INDEX IN ADDRESS TABLE
    ) equals JPI$C_ADRTYPE@8 increment 1  prefix JPI tag $;

                          /* ITEM IDENTIFIERS FOR PCBFLD

constant(
/* ADD ITEM-CODES BEFORE THIS COMMENT
     LASTPCBFLD /* MAX INDEX IN PCBFLD TABLE
    ) equals JPI$C_PCBFLDTYPE@8 increment 1  prefix JPI tag $;

                          /* ITEM IDENTIFIERS FOR PHDFLD

constant(
/* ADD ITEM-CODES BEFORE THIS COMMENT
     LASTPHDFLD /* MAX INDEX IN PHDFLD TABLE
    ) equals JPI$C_PHDFLDTYPE@8 increment 1  prefix JPI tag $;
end_module $JPIDEF;
```

```
module $KGBDEF;

/*++
/* Key Grant Block definitions: Format of records in the rights database
/* file. These records (1) associate identifier codes with names, and
/* (2) list the holders of all identifiers in the system.
/*--

aggregate KGBDEF structure prefix KGB$;

    IDENTIFIER longword unsigned;       /* Binary identifier code
    ATTRIBUTES structure longword unsigned; /* Attribute bit definitions
        RESOURCE bitfield mask;         /* Resource use allowed
    end ATTRIBUTES;
    HOLDER quadword unsigned;           /* Holder identifier
    constant HOLD_RECORD equals .;      /* End of holder record

    NAME character length 32;           /* Identifier name (blank filled string)
    constant IDENT_RECORD equals .;     /* End of identifier record

    LEVEL word unsigned;                /* File structure level
    constant LEVEL1 equals %X0101;      /* Version 1 structure level
    FILL_1 word fill tag $$;
    SYS_ID quadword unsigned;           /* System identifier
    NEXT_ID longword unsigned;          /* Next available identifier
    constant MAINT_RECORD equals .;     /* End of maintenance record

end KGBDEF;

/*++
/* Define the environmental rights ID values
/*--

    #ID = %x80000000;
    constant (BATCH_ID,                 /* Batch ID value
              DIALUP_ID,                /* Dialup ID value
              INTERACTIVE_ID,           /* Interactive ID value
              LOCAL_ID,                 /* Local ID value
              NETWORK_ID,               /* Network ID value
              REMOTE_ID)                /* Remote ID value
              equals #ID+1 increment 1 counter #ID prefix KGB$;

end_module $KGBDEF;
```

```
module $LADEF;
/*+
/*
/* LPA-11 CHARACTERISTICS DEFINITIONS
/*
/*-


                                                             /*MICROCODE TYPE VALUES
constant MRMCODE          equals 1  prefix LA tag $K;      /* MULTIREQUEST MICROCODE
constant ADMCODE          equals 2  prefix LA tag $K;      /* DEDICATED A/D MICROCODE
constant DAMCODE          equals 3  prefix LA tag $K;      /* DEDICATED D/A MICROCODE


aggregate LADEF  union prefix LA$;
    LADEF_BITS0 structure fill;
        MCVALID bitfield mask;                               /* MICROCODE VALID
        MCTYPE bitfield length 2;                            /* MICROCODE TYPE
        CONFIG bitfield length 10;                           /* DEVICE CONFIGURATION BITS
        RATE bitfield length 3;                              /* CLOCK RATE
        PRESET bitfield length 16;                           /* CLOCK PRESET
    end LADEF_BITS0;

    LADEF_BITS1 structure fill;
        FILL_1 bitfield length 3 fill prefix LADEF tag $$;/* SKIP OVER MICROCODE VALID AND TYPE
        CLOCKA bitfield mask;                                /* CLOCK A
        CLOCKB bitfield mask;                                /* CLOCK B
        AD1 bitfield mask;                                   /* A/D ! 1
        AD2 bitfield mask;                                   /* A/D ! 2
        DA bitfield mask;                                    /* D/A
        DIO1 bitfield mask;                                  /* DIGITAL I/O ! 1
        DIO2 bitfield mask;                                  /* DIGITAL I/O ! 2
        DIO3 bitfield mask;                                  /* DIGITAL I/O ! 3
        DIO4 bitfield mask;                                  /* DIGITAL I/O ! 4
        DIO5 bitfield mask;                                  /* DIGITAL I/O ! 5
    end LADEF_BITS1;

    LADEF_BITS2 structure fill;
        FILL_2 bitfield length 23 fill prefix LADEF tag $$;/*(IN USER'S COMMAND TABLE)
        BFROVRN bitfield mask;                               /* BUFFER OVERRUN NON-FATAL BIT
    end LADEF_BITS2;

end LADEF;

end_module $LADEF;
```

```
module $LCKDEF;
/*+
/* LOCK MANAGER DEFINITIONS
/*-


aggregate LCKDEF  union prefix LCK$;
    LCKDEF_BITS0 structure fill;
        VALBLK bitfield mask;            /* VALUE BLOCK INCLUDED
        CONVERT bitfield mask;           /* CONVERSION REQUEST
        NOQUEUE bitfield mask;           /* DO NOT QUEUE REQUEST
        SYNCSTS bitfield mask;           /* SYNCHRONOUS STATUS REQUESTED
        SYSTEM bitfield mask;            /* SYSTEM LOCK
        NOQUOTA bitfield mask;           /* DON'T CHARGE QUOTA
        CVTSYS bitfield mask;            /* CONVERT TO SYSTEM
        RECOVER bitfield mask;           /* RECOVER LOCK DURING FAILOVER
        PROTECT bitfield mask;           /* PROTECT LOCK DURING FAILOVER
        NODLCKWT bitfield mask;          /* NO DEADLOCK WAITING
        NODLCKBLK bitfield mask;         /* NO DEADLOCK BLOCKING
    end LCKDEF_BITS0;

    LCKDEF_BITS1 structure fill;
        DEQALL bitfield mask;            /* DEQUEUE ALL LOCKS
        CANCEL bitfield mask;            /* CANCEL REQUEST
        INVVALBLK bitfield mask;         /* INVALIDATE VALUE BLOCK
    end LCKDEF_BITS1;

                                         /*LOCK MODES
    constant(
          NLMODE                         /* NULL
        , CRMODE                         /* CONCURRENT READ
        , CWMODE                         /* CONCURRENT WRITE
        , PRMODE                         /* PROTECTED READ
        , PWMODE                         /* PROTECTED WRITE
        , EXMODE                         /* EXCLUSIVE
        ) equals 0 increment 1  prefix LCK tag $K;

end LCKDEF;

end_module $LCKDEF;
```

```
module $LKIDEF;
/*+
/*
/* Get Lock Information Data Identifier Definitions
/*
/* **** NOTE ****
/*
/*        New items must always be added to the END of each item list
/*        (preceeding the ENDlistname item) so that users will not have
/*        to relink.
/*-


                                                        /* ITEM LIST BLOCKS
aggregate LKIDEF structure prefix LKI$;
    LOCKID longword unsigned;                           /*LOCK ID
    PID    longword unsigned;                           /*PROCESS ID
    SYSID  longword unsigned;                           /*SYSTEM ID (RSB)
    RQMODE byte unsigned;                               /*REQUEST MODE
    GRMODE byte unsigned;                               /*GRANTED MODE
    QUEUE  byte unsigned;                               /*LOCK QUEUE
    SPARE  byte fill tag $$;                            /*SPARE BYTE
    REMLKID longword unsigned;                          /*REMOTE LOCK ID
    REMSYSID longword unsigned;                         /*REMOTE SYSTEM ID (LKB)
    constant 'LENGTH'' equals . prefix LKI$ tag K;      /*LENGTH OF LIST BLOCK
    constant 'LENGTH'' equals . prefix LKI$ tag C;      /*LENGTH OF LIST BLOCK
end LKIDEF;

                                                        /* DEFINE NAMSPACE BITS
aggregate NAMSPACE structure prefix LKI$;
    GROUP  word unsigned;                               /*GROUP OF OWNER UIC
    RMOD   byte unsigned;                               /*ACCESS MODE OF REQUEST
    STATUS byte unsigned;                               /*STATUS OF RESOURCE
        STATUS_BITS structure;
            FILL bitfield length 31 fill prefix LKI tag $$; /* SPARE
            SYSNAM bitfield mask;                       /* SYSTEM WIDE RSB
        end STATUS_BITS;
end NAMSPACE;

                                                        /* DEFINE STATE FIELDS
aggregate STATEF structure prefix LKI$;
    STATE_RQMODE byte unsigned;                         /*REQUEST MODE
    STATE_GRMODE byte unsigned;                         /*GRANTED MODE
    STATE_QUEUE  byte unsigned;                         /*LOCK STATE CODE
end STATEF;

constant (                                              /*LOCK STATE CODE VALUES
      GRANTED                                           /* GRANTED
    , CONVERT                                           /* CONVERSION
    , WAITING                                           /* WAITING
    , RETRY                                             /* RETRY REQUEST
    , SCSWAIT                                           /* SCS WAIT
    , RSPNOTQED                                         /* RESPONSE NOT QUEUED
    , RSPQUEUED                                         /* RESPONSE QUEUED
    , RSPGRANTD                                         /* RESPONSE GRANTED
    , RSPDOLOCL                                         /* RESPONSE DO LOCALLY
```

```
     , RSPRESEND                                          /* RESPONSE RESEND
     ) equals 1 increment -1  prefix LKI tag $C;

                                                          /* DEFINE TABLE NUMBERS
                                                          /* STRUCTURE CODES MUST START AT 1
constant LKBTYPE         equals 1  prefix LKI tag $C;     /* ITEM IN LKB
constant RSBTYPE         equals 2  prefix LKI tag $C;     /* ITEM IN RSB
constant LISTEND         equals 0  prefix LKI tag $C;     /* DEFINE ITMLST TERMINATOR

                                                          /* ITEM IDENTIFIERS FOR LKB


constant(
       PID                                                /* PROCESS ID
     , STATE                                              /* CURRENT LOCK STATE
     , PARENT                                             /* ID OF PARENT LOCK
     , LCKREFCNT                                          /* SUB-LOCK REFERENCE COUNT
     , LOCKID                                             /* LOCK ID
     , REMLKID                                            /* REMOTE LOCK ID
     , LASTLKB                                            /* MAX INDEX IN LKB TABLE
     ) equals LKI$C_LKBTYPE@8 increment 1  prefix LKI tag $;

                                                          /* ITEM IDENTIFIERS FOR RSB

constant(
       NAMSPACE                                           /* RESOURCE NAME SPACE
     , RESNAM                                             /* RESOURCE NAME
     , RSBREFCNT                                          /* SUB-RESOURCE REFERENCE COUNT
     , VALBLK                                             /* VALUE BLOCK
     , SYSTEM                                             /* SYSTEM ID OF SYSTEM WITH RESOURCE
     , LCKCOUNT                                           /* COUNT OF LOCKS ON RESOURCE
     , BLOCKEDBY                                          /* LIST OF LOCKS BLOCKED BY CURRENT LOCK
     , BLOCKING                                           /* LIST OF LOCKS BLOCKING CURRENT LOCK
     , LOCKS                                              /* LIST OF ALL LOCKS ON RESOURCE
     , LASTRSB                                            /* MAX INDEX IN RSB
     ) equals LKI$C_RSBTYPE@8 increment 1  prefix LKI tag $;

end_module $LKIDEF;
```

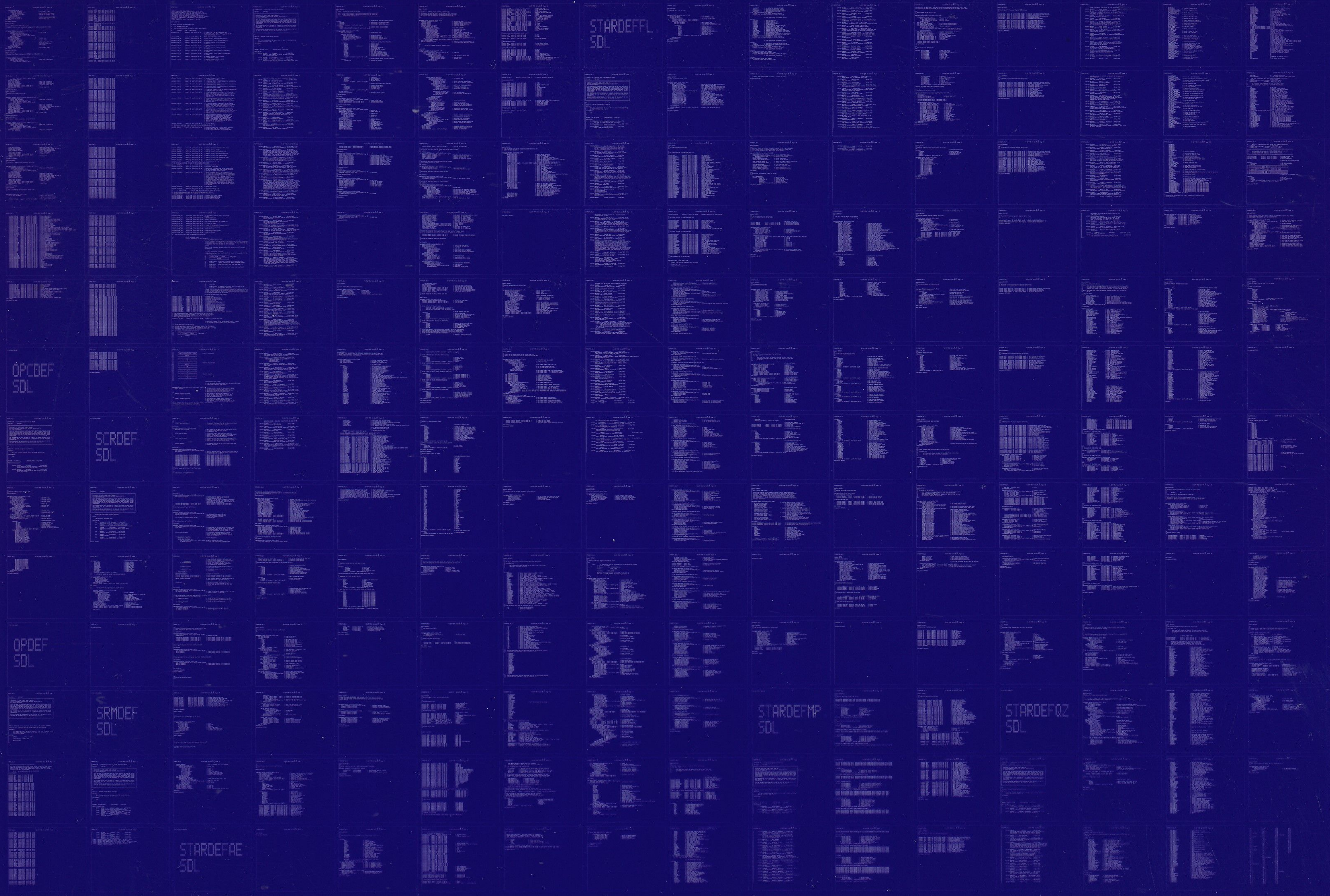
```
module $LNMDEF;
{+
{ LNM - LOGICAL NAME FLAGS
{
{ These flags combine logical name attributes, logical name translation
{ attributes, logical name table characteristics, and system service options.
{ These are all lumped into one definition for convenience in the user
{ interface.  The longword of bits that this defines is divided into four
{ bytes corresponding to the four categories just listed.  It is assumed
{ that these definitions correspond to equivalent bits in other structures.
{ This definition is used in the $CRELNT, $CRELNM, and $TRNLNM system service
{ interface definitions.
{-

aggregate LNMDEF structure prefix LNM$;
                                                /* Logical name attributes -- bits 0-7
    NO_ALIAS bitfield mask;                             /* Do not allow outer mode alias
    CONFINE bitfield mask;                              /* Do not copy into subprocess
    CRELOG bitfield mask;                               /* Created with old $CRELOG service
    TABLE bitfield mask;                                /* This is a table name
    FILL_0 bitfield length 4 fill;                      /* Fill out logical name byte
                                                /* Logical name translation attributes -- bits 8-15
    CONCEALED bitfield mask;                            /* Do not display result of translation
    TERMINAL bitfield mask;                             /* Do not retranslate result of translation
    EXISTS bitfield mask;                               /* Translation does exist at this index
    FILL_1 bitfield length 5 fill;                      /* Fill out translation byte
                                                /* Logical name table characteristics -- bits 16-23
    SHAREABLE bitfield mask;                            /* Logical name table is shareable (S0 space)
    FILL_2 bitfield length 3 fill;                      /* Reserved bit numbers 17-19
    FILL_3 bitfield length 4 fill;                      /* Fill out table byte
                                                /* System service options -- bits 24-31
    CREATE_IF bitfield mask;                            /* May map to existing logical name table
    CASE_BLIND bitfield mask;                           /* Perform case-insensitive translation
    FILL_4 bitfield length 6 fill;                      /* Fill out options byte

end LNMDEF;

constant "TABNAMLEN" equals  31 prefix LNM$ tag C;      /* Maximum length of a name contained within a directory table
constant "NAMLENGTH" equals 255 prefix LNM$ tag C;      /* Maximum logical name / translation length
constant "MAXDEPTH"  equals  10 prefix LNM$ tag C;      /* Maximum logical name recursion depth

constant (
    INDEX,                                              /* Translation index
    STRING,                                             /* Translation string
    ATTRIBUTES,                                         /* Attribute bits
    TABLE,                                              /* Logical name table name
    LENGTH,                                             /* Length of translation string
    ACMODE,                                             /* Access mode of name
    MAX_INDEX,                                          /* Maximum translation index
    PARENT,                                             /* Parent logical name table name
    LNMB_ADDR                                           /* Return LNM block address
                                                        /*  internal use by MTL
                                                        /*  and mailbox UCB
    ) equals 1 increment 1 prefix LNM$ tag '"';         /* Define item list codes

constant "CHAIN" equals -1 prefix LNM$ tag '"';         /* Chain to next list
```

```
end_module $LNMDEF;
```

```
module $LPDEF;
/*+
/* LINE PRINTER CHARACTERISTICS DEFINITIONS
/*-


aggregate LPDEF  union prefix LP$;
    LPDEF_BITS structure fill;
        CR bitfield mask;                               /* CARRIAGE RETURN NEEDED
        MECHFORM bitfield mask;                         /* MECHANICAL FORM FEED
        PRINTALL bitfield mask;                         /* NON PRINTABLE CHAR. ACCEPT(DMF-32)
        DAVFU bitfield mask;                            /* DAVFU AVAIL.
        WRAP bitfield mask;                             /* CHAR. WRAP MODE
        TAB bitfield mask;                              /* TAB needed
        TRUNCATE bitfield mask;                         /* TRUNCATE output at carriage width
        LOWER bitfield mask;                            /* PRINTER HAS LOWER CASE
        PASSALL bitfield mask;                          /* PASSALL MODE
        FALLBACK bitfield mask;                         /* FALLBACK MODE
        SIXELS bitfield mask;                           /* HANDLES SIXELS
        BITMAPPED bitfield mask;                        /* HANDLES BIT MAPS
        FILL_2 bitfield length 12 fill prefix LPDEF tag $$;/* SPARE UNUSED BITS
        PAGE_L bitfield mask length 8;                  /* PAGE LENGTH
    end LPDEF_BITS;

    constant LP11       equals 1  prefix LP tag $;
    constant LA11       equals 2  prefix LP tag $;
    constant LA180      equals 3  prefix LP tag $;

end LPDEF;

end_module $LPDEF;
```

0433 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



STARDEFFL
SDL

OPCDEF
SDL

SCRDEF
SDL

OPDEF
SDL

SRMDEF
SDL

STARDEFMP
SDL

STARDEFQZ
SDL

STARDEFAE
SDL