

{ MAILDEF.SDL - define mail data structures

{ Version 'V04-000'

```

{*****
{*
{* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
{* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
{* ALL RIGHTS RESERVED.
{*
{* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
{* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
{* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
{* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
{* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
{* TRANSFERRED.
{*
{* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
{* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
{* CORPORATION.
{*
{* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
{* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
{*
{*****

```

```

{++
{ FACILITY: VAX/VMS Personal Mail Utility
{ ABSTRACT:
{ This file contains the SDL definitions for MAIL data structures

```

{ AUTHOR: Benn Schreiber CREATION DATE: 28-Jun-1982

{ MODIFIED BY:

- { V03-013 BLS0292 Benn Schreiber 8-APR-1984
New flag for attachment.
- { V03-012 BLS0272 Benn Schreiber 18-FEB-1984 13:35:27
New flag for server loop.
- { V03-011 BLS0263 Benn Schreiber 5-FEB-1984
Add new flags for distinguishing between editors.
- { V03-010 BLS0255 Benn Schreiber 28-Dec-1983
New flags for create_isam and mail global control
- { V03-009 BLS0246 Benn Schreiber 28-Nov-1983
Add mail system global flags
- { V03-008 BLS0237 Benn Schreiber 1-Sep-1983

MA
MA
MA
:
:
LI
:
:
MA
:
:
SL
PS

{ Add allocated length field to TXT block
{
{ V03-007 BLS0235 Benn Schreiber 23-Aug-1983
{ New flag for non-standard file format.
{
{ V03-006 BLS0229 Benn Schreiber 16-Jul-1983
{ Changes to support network block io
{
{ V03-005 BLS0222 Benn Schreiber 7-May-1983
{ Add CCDESC field to connect descriptor
{
{ V03-004 BLS0220 Benn Schreiber 30-Apr-1983
{ New bit for NOAUTOPURGE and add mail subdir length field
{
{ V03-003 BLS0217 Benn Schreiber 13-Apr-1983
{ Add COPYSELF flags to VMD, and external message info
{ to MMB and MMH.
{
{ V03-002 BLS0214 Benn Schreiber 27-Mar-1983
{ Stabilize, remove extraneous definitions. Add NREC
{ offsets in MMB and MMH.
{
{--

```

( Define data structures used by MAIL
(
( The structure of the VMSMAIL.DAT file. This file contains forwarding and
( other information about a username. If a record exists and does not have
( a forwarding address VMD_B_FWDLNG=0.
(
module vmdef;

aggregate vmdef structure prefix VMD_;

  constant USERNAME equals 31 prefix VMD_ tag K;
  constant USERNAME equals 31 prefix VMD_ tag C;
  USERNAME character length 31; /*USERNAME BLANK FILLED
  FLAGS structure word unsigned; /*FLAGS
    selfsend bitfield mask; /* send a copy to self on SEND
    selfreply bitfield mask; /* send a copy to self on REPLY
    noautopurge bitfield mask; /* Don't autopurge on exit or ^Z
  end flags;
  MAIL word unsigned; /*COUNT OF NEW MESSAGES
  SPARE longword unsigned dimension(7); /*SPARE LUNGWORDS
  SPARE1 byte unsigned dimension(2); /*SPARE BYTES
  DIRLNG byte unsigned; /*LENGTH OF DEV/DIR IF SET
  FNMLNG byte unsigned; /*LENGTH OF FULL USERNAME
  FWDLNG byte unsigned; /*LENGTH OF FORWARDED NAME
  FWDNAM character length 0; /*FORWARDED NAME
  constant 'LENGTH' equals . prefix VMD_ tag K;
  constant 'LENGTH' equals . prefix VMD_ tag C;

end vmdef;

end_module vmdef;

module adrdef;
/*+
/* Address list block entry
/*-

aggregate adrdef structure prefix ADR_;

  flink longword unsigned; /*Forward link to next entry
  blink longword unsigned; /*Backward link to previous
  lnk longword unsigned; /*Logical link entry address
  spares longword unsigned dimension(4);
  flays structure byte unsigned; /*Flags
    nosend bitfield mask; /*Do not send to this user
  end flags;
  namlng byte unsigned; /*Length of addressee name
  name character length 0; /*Start of addressee name
  constant 'LENGTH' equals . prefix ADR_ tag C;

end adrdef;

end_module adrdef;

module lnkdef;

```

```
aggregate lnkdef structure prefix LNK_;
```

```

fblink longword unsigned; /*Forward link to next entry
blink longword unsigned; /*Backward link to previous
adrcnt longword unsigned; /*Count of addressees this link
context longword unsigned; /*Context for alternate protocol
tfradr longword unsigned; /*Transfer address of alt prot. image
iosb quadword; /*IOSB
sts longword unsigned; /*Staus from failing access
spares longword unsigned dimension(2);
sparew word unsigned;
mbxchan word unsigned; /*Mailbox channel number
chan word unsigned; /*Channel number
flags structure byte unsigned; /*Flags
  msgsent bitfield mask; /*Message already sent
  dead bitfield mask; /*Error occurred on initialization
  altp bitfield mask; /*Send with alt protocol
  fsent hitfield mask; /*FROM has been sent
  blkmode bitfield mask; /*Send/receive block mode
end flags;
nodlen byte unsigned; /*Node name length
node character length 31; /*Storage for node name
pnlenn byte unsigned; /*Protocol name length
pnam character length 0; /*Start of protocol name
constant 'LENGTH' equals . prefix LNK_ tag C;
constant (

```

```

  out_connect
  ,out_sender
  ,out_ckuser
  ,out_to
  ,out_subj
  ,out_file
  ,out_cksend
  ,out_deaccess
  ,in_connect
  ,in_sender
  ,in_ckuser
  ,in_to
  ,in_subj
  ,in_file
  ,io_read
  ,io_write

```

```

) equals 0 increment 1 prefix LNK_ tag C;
end lnkdef;
```

```
end_module lnkdef;
```

```
module mmbdef;
```

```

/*+
/* Message descriptor block. There is one message descriptor block for each
/* message, created when the isam message file is scanned.
/*-

```

```
aggregate mmbdef structure prefix MMB_;
```

```

rfa0 longword unsigned; /*VBN of message in file
rfa4 word unsigned; /*Byte offset of message in file
txtlst longword unsigned; /*Pointer to text of message if present
flags structure word unsigned; /*Flags word
  newmsg bitfield mask; /*This is a new message
  replied bitfield mask; /*This message has been replied to
  del bitfield mask; /*This message is deleted
  extmsg bitfield mask; /*Message text in external file
  extfnf bitfield mask; /*External message file not found
  sysmsg bitfield mask; /*Message text in system file
  extnstd bitfield mask; /*External file is not var-seq file
end flags;
flagval longword unsigned; /*Flag value assigned by user
rdisp byte unsigned; /*Dispatch for mail$get_record
constant (
  none /*No action
  ,from /*Send from line
  ,to /* to
  ,subj /* subj
  ,blank /* blank line after subject
  ,text /* setup and recurse for msg text
  ,fromxdt /* from line with no date
  ,cc /* cc
) equals 0 increment 1 prefix MMB_ tag C;
hdreclen word unsigned; /*Length of header record
fromdesc quadword; /*descriptor of FROM string in hd rec
todesc quadword; /* ditto for TO string
subjdesc quadword; /* and SUBJ
ccdesc quadword; /* and CC
nrecs longword unsigned; /*Number of records in message
datid quadword; /*Date/time id of message text
nblks longword unsigned; /*Number of blocks in msg
spare longword unsigned dimension(1);
hdrec character length 0; /*Start of header record
constant 'LENGTH' equals . prefix MMB_ tag C;

```

end mmbdef;

end_module mmbdef;

module txtdef;

```

/*+
/* Text descriptor block. A linked list of text records is created as the text
/* c' the message is read into memory.
/*-

```

aggregate txtdef structure prefix txt_;

```

next longword unsigned; /*Next text block or 0 if end
textlen longword unsigned; /*Length of record text
alloc longword unsigned; /*Size of allocated txt block
text character length 0; /*Start of message text
constant 'LENGTH' equals . prefix TXT_ tag C;

```

end txtdef;

```
end_module txtdef;
```

```
module mfrdef;
```

```
/*+
/*Define the format of a mail file record common part of record.
/*-
```

```
aggregate mfrdef structure prefix MFR_;
```

```
    datim quadword;          /*Date/time message received
                             /*Date/times with high quadword 0 are reserved
                             /* as info records
    constant (
        lastread             /*Date/time of last read message
        ,wastename           /*Name of wastebasket folder
        ,fileinfo           /*File status information
        ,mw                  /*Mailwatch info record
        ,newmsg              /*New message information
    ) equals 1 increment 1 prefix MFR_ tag C;
    filkeylen byte unsigned; /*Length of the filing key
    filekey character length 39; /*Filing key
    data character length 0; /*Start of data part of record

    constant maxlen equals 2048 prefix MFR_ tag C; /*Maximum length of a record
end mfrdef;
```

```
end_module mfrdef;
```

```
module mmhdef;
```

```
/*+
/*Define a mail message header record
/*-
```

```
aggregate mmhdef structure prefix MMH_;
```

```
    flags structure word unsigned;
        newmsg bitfield mask; /*This is a new message
        replied bitfield mask; /*This message has been replied to
        skip1 bitfield mask; /*Skip a bit
        extmsg bitfield mask; /*Message text in external file
        sysmsg bitfield mask; /*Message text in system file
    end flags;
    flagsiz byte unsigned; /*Size of flag string if ascii
    flagval longword unsigned; /*Value of /FLAG=
    hdspace byte unsigned; /*Spare byte to align datid
    datid quadword; /*Key ID of data record
    miscdat character length 0; /*Start of variable data
                             /*Format: type.w,size.w,data.v
    constant (
        from /*Defined type values
        ,to /*the FROM field
        ,subj /*the TO field
        ,cc /*the SUBJ field
        ,cc /*the CC field
        ,mw /*MAILWATCH info
        ,nrec /*Number of records in message
        ,extmsg /*Message is in external file; this is spec
```



```

) equals 0 increment 1 prefix MMH tag C;
/*5-32767 reserved to DIGITAL
/*32768-65535 reserved for customers

```

```
end mmhdef;
```

```
end_module mmhdef;
```

```
module cnctdef;
```

```
/*+
/*Descriptor for a message
/*-

```

```
aggregate cnctdef structure prefix cnct_;
```

```

fblink longword unsigned; /*Forward link in queue
blink longword unsigned; /*Backward link
fromdesc quadword; /*Descriptor of "from"
todesc quadword; /*Descriptor of "to"
subjdesc quadword; /*Descriptor of "subject"
ccdesc quadword; /*Descriptor of "CC list"
adrlst quadword; /*Listhead of addressees this message
lnklst quadword; /*Listhead for connections outbound
naddrs longword unsigned; /*Number of addressees this message
msgblks longword unsigned; /*Number of blocks in message
iosb structure quadword; /*IOSB
    iosb1 word unsigned;
    iosb2 word unsigned;
    iosb3 word unsigned;
    iosb4 word unsigned;
end iosb;
filerr longword unsigned; /*First error message (i.e. OPENOUT)
filests longword unsigned; /*Status if error opening temp file
filestv longword unsigned; /*STV from the error
esdesc quadword; /*Descriptor of temp file
sendq quadword; /*Fwd/bkwd links for send queue
nrecs longword unsigned; /*Number of records in message
spares longword unsigned dimension(6); /*To avoid recompilation!
filrat byte unsigned; /*Record attributes of message file
filrfm byte unsigned; /*Record format of message file
chan word unsigned; /*Channel number
flags structure byte unsigned;
    disconnect bitfield mask; /*A disconnect has occurred this link
    sq bitfield mask; /*This link is in SENDQ
    blkmode bitfield mask; /*Incoming link is block mode
end flags;
tran byte unsigned; /*Operation reading from remote link
constant ( /*Defined type values
    getsender /*the sender name
    .getaddr /*the addressee names
    .getto /*the TO field
    .getsubj /*the SUBJ field
    .getmsg /*the message itself
) equals 0 increment 1 prefix CNCT_ tag C;

buffer character length 512; /*Input buffer
constant bufsiz equals 512 prefix CNCT_ tag C;

```

```

rab character length 0;          /*Start of RAB
constant 'LENGTH' equals . prefix CNCT_ tag C;
end cnctdef;

end_module cnctdef;
module cnfdef;
/**
/*Configuration data passed/read from remote node
/*-
aggregate cnfdef structure prefix cnf_;

version byte unsigned;          /*MAIL-11 protocol version
constant vers equals 3 prefix CNF_ tag C;
eco byte unsigned;             /*MAIL-11 protocol ECO level
constant eco equals 0 prefix CNF_ tag C;
custeco byte unsigned;         /*MAIL-11 protocol Customer ECO
os byte unsigned;              /*Operating system number
constant vaxvms equals 7 prefix CNF_ tag C;
options structure longword unsigned; /*Options
notifymsg bitfield mask;       /*Send "receiving user notified msgs"
end options;
iomode structure longword unsigned; /*Message send/receive mode
blksend bitfield mask;         /* Sender wants to send in block mode
blkrcv bitfield mask;          /* Receiver will accept blocks
pfxsend bitfield mask;         /* Sender wants to add node name
pfxrcv bitfield mask;          /* Receiver will let sender do it
ccsend bitfield mask;          /* Sender wants to send CC field
ccrcv bitfield mask;           /* Receiver will accept CC field
end iomode;
rfm byte unsigned;             /*Message record format
rat byte unsigned;             /*Message record attributes
spare1 byte unsigned;          /*Spare byte
spare2 byte unsigned;          /*Spare byte

constant 'LENGTH' equals . prefix CNF_ tag C;
end cnfdef;

end_module cnfdef;
module sysfdef;
/**
/*Flags defined in MAIL$SYSTEM_FLAGS logical name
/*-
aggregate sysfdef structure prefix sysf_;

flags structure longword unsigned;
clu_mail bitfield mask;        /*All accounts are cluster accessible
/* (i.e. homogeneous cluster)
clu_brkthru bitfield mask;     /*Use set brk$m_cluster
notify_time bitfield mask;     /*Include time in notify message
end flags;
end sysfdef;

```

```
end_module sysfdef;
module prqdef;
```

```
/*+
/*Flags defined flags passed to mail$print
/*-
```

```
aggregate prqdef structure prefix prq_;
```

```
  flags structure longword unsigned;
    notify bitfield mask;      /*notify when done
    force bitfield mask;      /*close open job, start printing
    all bitfield mask;        /*print all selected messages
  end flags;
end prqdef;
```

```
end_module prqdef;
module mcrdef;
```

```
/*+
/*Flags passed to mail$create_isam and copy_msgs
/*-
```

```
aggregate mcrdef structure prefix mcr_;
```

```
  flags structure longword unsigned;
    snd_newmsg bitfield mask;  /*Deliver as new mail
    all bitfield mask;        /*All messages
    cif bitfield mask;        /*Create file if necessary
    nil bitfield mask;        /*Do not allow others access
    use_datim bitfield mask;   /*Use date/time passed, not current
    tell_new bitfield mask;    /*Tell that new folder created
    cnct_rab bitfield mask;    /*Input file RAB is in cnct block
    delmsg bitfield mask;     /*Delete message after copying
    confnew bitfield mask;    /*Confirm if new folder created
    exec bitfield mask;       /*Use exec mode logicals only
  end flags;
end mcrdef;
```

```
end_module mcrdef;
module maifdef;
```

```
/*+
/*MAIL control flags
/*-
```

```
aggregate maifdef structure prefix maif_;
```

```
  flags structure longword unsigned;
    isam bitfield mask;       /*Current file is isam file
    netjob bitfield mask;     /*Network receiver mode
    iterm bitfield mask;      /*Input is a terminal
    oterm bitfield mask;      /*Output is a terminal
    scope bitfield mask;     /*Output is a scope
    rdhold bitfield mask;     /*Record remainder from ff to print
    read_fl bitfield mask;    /*Msg has been read, can be deleted
    recl bitfield mask;       /*purge/reclaim, not auto-mode
    ctrlcfl bitfield mask;    /*CTRL/C typed
```

```
batch bitfield mask;      /*Batch job
altp bitfield mask;       /*Net job, alt protocol
filopn bitfield mask;     /*Mail file is open
dfmsg bitfield mask;     /*In the default message file
useredit bitfield mask;  /*Use user-specified mailedit.com
captive bitfield mask;   /*User is in captive account
caledt bitfield mask;    /*Callable editor specified
edtedt bitfield mask;    /*Callable EDT specified
edtxyz bitfield mask;    /*Callable XYZ specified
serverloop bitfield mask; /*Server should loop
attachment bitfield mask; /*SEND/ATTACH specified
end flags;
end maifdef;
end_module maifdef;
```

The image displays a grid of 100 terminal window screenshots, arranged in 10 rows and 10 columns. Each window shows a different view of a VAX/VMS system utility or command-line interface. The windows are organized into several groups, with some highlighted by white labels:

- CMDS LIS**: Located in the top-left area, showing a list of commands.
- DIRECTORY LIS**: Located in the top-right area, showing a directory listing.
- EDIT LIS**: Located in the middle-right area, showing a text editor interface.
- GETUSER LIS**: Located in the middle-right area, showing user information.
- MAILREQ REQ**: Located in the middle-left area, showing mail request details.
- MAILDEF SDL**: Located in the middle-left area, showing mail definition settings.
- ADDRLIST LIS**: Located in the bottom-left area, showing an address list.
- GETNEW LIS**: Located in the bottom-right area, showing new mail information.

The remaining windows in the grid show various other system utilities, command-line prompts, and data outputs, all rendered in a monospaced font typical of early computer terminals.