


```

MM      MM      SSSSSSSS  GGGGGGGG  DDDDDDDD  EEEEEEEEE  FFFFFFFF
MM      MM      SSSSSSSS  GGGGGGGG  DDDDDDDD  EEEEEEEEE  FFFFFFFF
MMMM    MMMM    SS        GG        DD        DD  EE        FF
MMMM    MMMM    SS        GG        DD        DD  EE        FF
MM      MM      SS        GG        DD        DD  EE        FF
MM      MM      SS        GG        DD        DD  EE        FF
MM      MM      SSSSSS    GG        DD        DD  EEEEEEE   FFFFFFFF
MM      MM      SSSSSS    GG        DD        DD  EEEEEEE   FFFFFFFF
MM      MM      SS        GG  GGGGGG  DD        DD  EE        FF
MM      MM      SS        GG  GGGGGG  DD        DD  EE        FF
MM      MM      SS        GG        GG    DD        DD  EE        FF
MM      MM      SS        GG        GG    DD        DD  EE        FF
MM      MM      SSSSSSSS  GGGGGG  DDDDDDDD  EEEEEEEEE  FF
MM      MM      SSSSSSSS  GGGGGG  DDDDDDDD  EEEEEEEEE  FF

```

```

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

```

```

MAC
MAC
LI
LI

```

```
{-----
{
  Structure definitions for message sections
{-----
{
```

```
  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.
{*
{*
{******
```

```
{
{
  Define the message section format
{
```

```
module $MSCDEF;
```

```
aggregate MSCDEF structure prefix MSC$:
  TYPE byte unsigned; /* TYPE OF MESSAGE SECTION
  constant(
    MSG /* CONTAINS ACTUAL MESSAGE TEXT
    . IND /* CONTAINS FILE SPEC OF MESSAGE FILE
    ; MAXTYPE /* MAXIMUM ALLOWABLE TYPE CODE
  ) equals 0 increment 1 prefix MSC tag $C;
  FLAGS_OVERLAY union;
    FLAGS byte unsigned; /* FLAGS
    FLAGS_BITS structure; /* TRUE IF FILE HAS BEEN MAPPED (IND ONLY)
      MAPPED bitfield mask;
    end FLAGS_BITS;
  end FLAGS_OVERLAY;
  SANITY word unsigned; /* SANITY CHECK (MUST = MSC$C_SANITY)
  constant SANITY equals 1234 prefix MSC tag $C;
  SIZE longword unsigned; /* LENGTH OF ENTIRE MESSAGE SECTION (MSG)
  INDEX_OFF longword unsigned; /* OFFSET TO PRIMARY INDEX PORTION
```

```
FAC OFF longword unsigned; /* OFFSET TO FACILITY TABLE
TEXT OFF longword unsigned; /* OFFSET TO TEXT PORTION
SUBST OFF longword unsigned; /* OFFSET TO TEXT SUBSTITUTION TABLE
FILL T longword dimension 4 fill prefix MSCDEF tag $$; /* RESERVED
constant "LENGTH" equals . prefix MSCS tag K; /* LENGTH OF FIXED PORTION (MSG)
constant "LENGTH" equals . prefix MSCS tag C; /* LENGTH OF FIXED PORTION (MSG)

end MSCDEF;

aggregate MSCDEF1 structure prefix MSCS;
  FILL 2 byte dimension 8 fill prefix MSCDEF tag $$; /* LENGTH OF COUNTED STRING
  INDXMLEN byte unsigned; /* FILE SPEC OF MESSAGE FILE
  INDXNAME character;
end MSCDEF1;

end_module $MSCDEF;

module $MIDXDEF;

/*
/* Define message section index
/*
/* All message indicies are longword aligned and the
/* size is always rounded to the next longword boundary.
/* This is done for additional verification checks.
/*

constant BKTSIZ equals 512 prefix MIDX tag $C; /* BUCKET SIZE IN BYTES

aggregate MIDXDEF structure prefix MIDXS;
  SIZE word unsigned; /* LENGTH OF INDEX IN BYTES
  SANITY byte unsigned; /* SANITY CHECK (MUST = MIDX$C_SANITY)
  constant SANITY equals 123 prefix MIDX tag $C;
  FILL 1 byte dimension 5 fill prefix MIDXDEF tag $$; /* UNUSED, ROUND SO BYTES LEFT DIV. BY 8
  constant "LENGTH" equals . prefix MIDXS tag K; /* LENGTH OF FIXED BUCKET OVERHEAD
  constant "LENGTH" equals . prefix MIDXS tag C; /* LENGTH OF FIXED BUCKET OVERHEAD
  constant ENTRIES equals . prefix MIDXS tag K; /* START OF INDEX ENTRIES
  constant ENTRIES equals . prefix MIDXS tag C; /* START OF INDEX ENTRIES
/* EACH IS 8 BYTES,
/* 0-3: MESSAGE CODE
/* 4-7: OFFSET TO MESSAGE RECORD
/* (IF LOW BIT SET, THEN OFFSET
/* POINTS TO SUBINDEX RATHER THAN
/* A MESSAGE RECORD)

end MIDXDEF;

end_module $MIDXDEF;

module $MRECDEF;

/*
/* Message definition record in message section
/*
```

```

/*      All message definition records are word aligned and
/*      the size is rounded to the next word boundary. This
/*      is done for additional verification checks.
/*

```

```

aggregate MRECDEF structure prefix MRECS;
  SIZE word unsigned;          /* LENGTH OF MESSAGE DEFINITION RECORD
  TYPE byte unsigned;         /* TYPE OF MESSAGE RECORD
  FLAGS byte unsigned;        /* FLAGS
  LEVEL byte unsigned;        /* MESSAGE DETAIL LEVEL (0-255)
  FAOCNT byte unsigned;       /* NUMBER OF FAO ARGUMENTS IN MESSAGE
  USERVAL byte unsigned;      /* USER SPECIFIED VALUE
  FILL_1 byte fill prefix MRECDEF tag $$; /* UNUSED
  LANG byte unsigned;         /* LANGUAGE NUMBER
  constant(
    ENGLISH                    /* ENGLISH LANGUAGE (DEFAULT)
    , GERMAN                    /* GERMAN LANGUAGE
    , FRENCH                    /* FRENCH LANGUAGE
  ) equals 0 increment 1 prefix MREC tag $C;
  constant FIXEDLEN equals . prefix MRECS tag K; /* LENGTH OF FIXED PORTION OF RECORD
  constant FIXEDLEN equals . prefix MRECS tag C; /* LENGTH OF FIXED PORTION OF RECORD
  IDENTLEN byte unsigned;     /* MESSAGE IDENT STRING LENGTH
  "IDENT" character;          /* MESSAGE IDENT STRING
  /* MESSAGE TEXT FOLLOWS (ASCIC)
end MRECDEF;

end_module $MRECDEF;

module $MFACDEF;

/*
/*      Define facility name table within section
/*

aggregate MFACDEF structure prefix MFACS;
  NUMBER word unsigned;       /* FACILITY NUMBER
  NAMELEN byte unsigned;     /* LENGTH OF FACILITY NAME
  NAME character;             /* FACILITY NAME (VARIABLE LENGTH)
end MFACDEF;

end_module $MFACDEF;

```

