


```

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

```

```

....
....
....
....

```

```

SSSSSSS  DDDDDDD  LL
SSSSSSS  DDDDDDD  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
SSSSSS   DDDDDDD  LLLLLLLLLL
SSSSSS   DDDDDDD  LLLLLLLLLL

```

```

-----
{
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_1 longword dimension 4 fill prefix MSCDEF tag $$; /* RESERVED
constant "LENGTH" equals . prefix MSC$ tag K; /* LENGTH OF FIXED PORTION (MSG)
constant "LENGTH" equals . prefix MSC$ tag C; /* LENGTH OF FIXED PORTION (MSG)

end MSCDEF;

aggregate MSCDEF1 structure prefix MSC$;
  FILL_2 byte dimension 8 fill prefix MSCDEF tag $$; /* LENGTH OF COUNTED STRING
  INDNAMLEN byte unsigned; /* FILE SPEC OF MESSAGE FILE
  INDNAME 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 MIDX$;
  SIZE word unsigned; /* LENGTH OF INDEX IN BYTES
  SANITY byte unsigned; /* SANITY CHECK (MUST = MIDX$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 MIDX$ tag K; /* LENGTH OF FIXED BUCKET OVERHEAD
  constant "LENGTH" equals . prefix MIDX$ tag C; /* LENGTH OF FIXED BUCKET OVERHEAD
  constant ENTRIES equals . prefix MIDX$ tag K; /* START OF INDEX ENTRIES
  constant ENTRIES equals . prefix MIDX$ 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 MREC$;
```

```
SIZE word unsigned;
TYPE byte unsigned;
FLAGS byte unsigned;
LEVEL byte unsigned;
FAOCNT byte unsigned;
USERVAL byte unsigned;
FILL_1 byte fill prefix MRECDEF tag $$;
LANG byte unsigned;
constant(
    ENGLISH
    , GERMAN
    , FRENCH
) equals 0 increment 1 prefix MREC tag $C;
constant FIXEDLEN equals . prefix MREC$ tag K;
constant FIXEDLEN equals . prefix MREC$ tag C;
IDENTLEN byte unsigned;
"IDENT" character;
```

```
/* LENGTH OF MESSAGE DEFINITION RECORD
/* TYPE OF MESSAGE RECORD
/* FLAGS
/* MESSAGE DETAIL LEVEL (0-255)
/* NUMBER OF FAO ARGUMENTS IN MESSAGE
/* USER SPECIFIED VALUE
/* UNUSED
/* LANGUAGE NUMBER
/* ENGLISH LANGUAGE (DEFAULT)
/* GERMAN LANGUAGE
/* FRENCH LANGUAGE
/* LENGTH OF FIXED PORTION OF RECORD
/* LENGTH OF FIXED PORTION OF RECORD
/* MESSAGE IDENT STRING LENGTH
/* MESSAGE IDENT STRING
/* MESSAGE TEXT FOLLOWS (ASCIC)
```

```
end MRECDEF;
```

```
end_module $MRECDEF;
```

```
module $MFACDEF;
```

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

```
aggregate MFACDEF structure prefix MFAC$;
```

```
NUMBER word unsigned;
NAMELEN byte unsigned;
NAME character;
end MFACDEF;
```

```
/* FACILITY NUMBER
/* LENGTH OF FACILITY NAME
/* FACILITY NAME (VARIABLE LENGTH)
```

```
end_module $MFACDEF;
```

This section of the document displays a large grid of approximately 12 columns and 12 rows of small, low-contrast images. Each image represents a frame of data, likely from a microfiche or a set of punched cards. The individual frames contain very faint, illegible characters and patterns, which appear to be digital data or code. The overall appearance is that of a high-density storage or processing output from a digital system.

MSGINT
MDL

SQLGENREQ
REQ

MSGDEF
SQL

CUTMSGCOM
FOR

CUTMSG
LIS

MSG
REQ

MAIN
LIS

LISTING
LIS