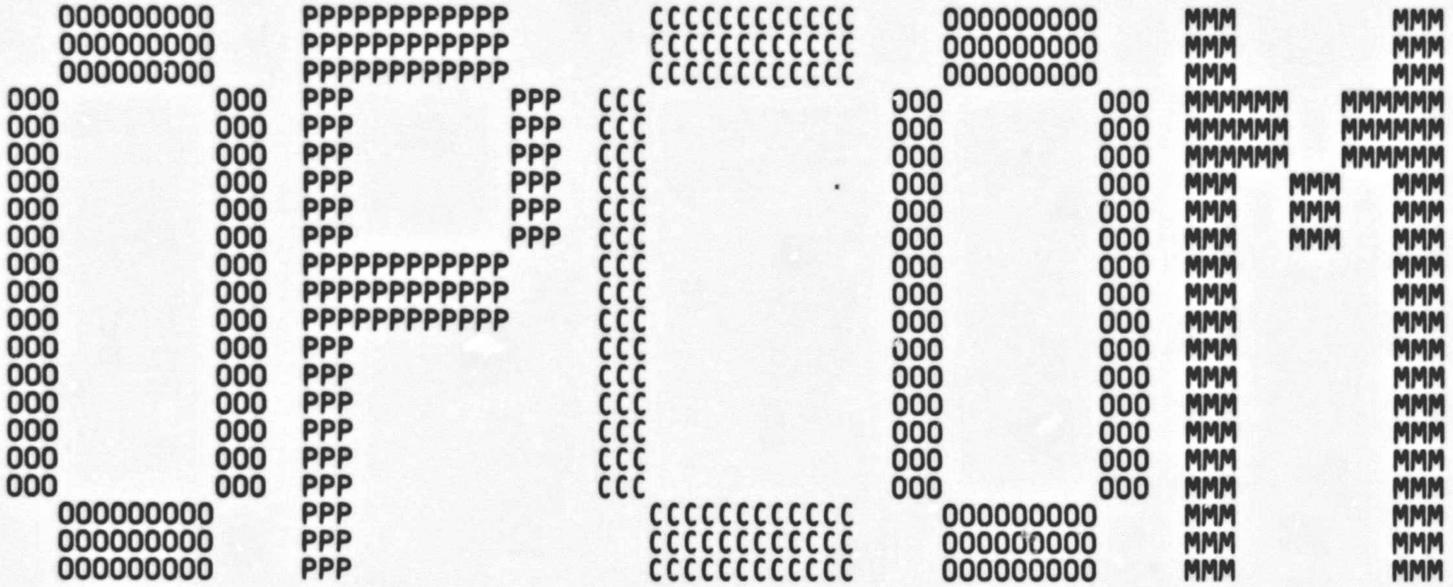

_$2

Sym

ALL

ASC

BOD
BOD
BOD
BOD
BOD
BOD
BOD
BOD
BUG
BYP
CAN
CAN
CAN
CHE
CHE

CLU
CLU
CLU
CLU
CLU
CLU
CLU

CLU
CLU
CLU
CLU
CLU
CLU
CLU

CLU
CLU

```
  OOOOOO    PPPPPPPP      CCCCCCCC    OOOOOO    MM      MM  DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF
  OOOOOO    PPPPPPPP      CCCCCCCC    OOOOOO    MM      MM  DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF
OO      OO  PP      PP  CC          OO      OO  MMMM  MMMM  DD      DD  EE          FF
OO      OO  PP      PP  CC          OO      OO  MMMM  MMMM  DD      DD  EE          FF
OO      OO  PP      PP  CC          OO      OO  MM  MM  MM  DD      DD  EE          FF
OO      OO  PP      PP  CC          OO      OO  MM  MM  MM  DD      DD  EE          FF
OO      OO  PPPPPPPP    CC          OO      OO  MM      MM  DD      DD  EEEEEEEE    FFFFFFFF
OO      OO  PPPPPPPP    CC          OO      OO  MM      MM  DD      DD  EEEEEEEE    FFFFFFFF
OO      OO  PP          CC          OO      OO  MM      MM  DD      DD  EE          FF
OO      OO  PP          CC          OO      OO  MM      MM  DD      DD  EE          FF
OO      OO  PP          CC          OO      OO  MM      MM  DD      DD  EE          FF
OO      OO  PP          CC          OO      OO  MM      MM  DD      DD  EE          FF
  OOOOOO    PP            CCCCCCCC    OOOOOO    MM      MM  DDDDDDDD    EEEEEEEEEE  FF
  OOOOOO    PP            CCCCCCCC    OOOOOO    MM      MM  DDDDDDDD    EEEEEEEEEE  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
```

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

```
{ OPCOMDEF.SDL - system definition file for OPCOM internal 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:  OPCOM  -  Operator Communications
{
{ ABSTRACT:
{
{       This file contains the SDL source for OPCOM internal structure
{       definitions.
{
{ AUTHOR:  CW Hobbs      CREATION DATE:  27-Jun-1983
{
{ MODIFICATION HISTORY:
{
{       V03-003 CWH3169         CW Hobbs                        5-May-1984
{               Second pass for cluster-wide OPCOM:
{                - Remove CLUSTER_ENABLED bit, redo EFN numbers, add NODE_LEAVING
{                - Add BOD and COD structures
{                - Add CSID to clm header & bump structure version numbers for clm structures
{
{       V03-002 CWH3002         CW Hobbs                        16-Sep-1983
{               Add io error bit
{--
module OPCOMDEF;
```

```
/*
/* Define bit vector names for global status
/*
constant (
    SHUTDOWN_PENDING,           /* OPCOM is doing an orderly shutdown
    LOGGING_ENABLED,            /* Log events
    LAST_LOG_FAILED,            /* Last log attempt failed
    LOGFILE_CLOSED,             /* Logfile closed
    OLD_FORMAT_MSG,             /* This is an old format msg
    TIMESTAMP_PENDING,          /* Timestamp function pending
    BUSY,                       /* OPCOM is busy
    FLUSH_PENDING,              /* Must flush log file
    IN_VAXcluster               /* Member of VAXcluster
    ) equals 0 increment 1 prefix GBLSTS_;

/*
/* Define event flag numbers
/*
constant (
    ASYNCH,                     /* Throwaway for asynch i/o
    BRKTHRU,                    /* Break through service
    MAILBOX,                    /* Mailbox reads
    TIME_STAMP                  /* Timer ast
    ) equals 1 increment 1 prefix EFN_;

/*
/* Define the common data structure header fields.
/* All of the data structure definitions that follow
/* this one have the first few fields in common.
/* However, some of the fields are used differently,
/* depending on the type of data structure.
/*
aggregate HDR_AGGREGATE structure prefix HDR_ fill;
    FLINK       longword unsigned;      /* Flink to next data structure
    BLINK       longword unsigned;      /* Blink to last data structure
    SIZE        word unsigned;          /* Size of data structure
    TYPE        byte unsigned;          /* Type of data structure
    SCOPE       byte unsigned;          /* Scope of data structure
    SEQNUM      longword unsigned;      /* Sequence # of data structure
    IDENT       longword unsigned;      /* Cluster-wide ident (seqnum from original create)
    CSID        longword unsigned;      /* Cluster system id
    NOD         longword unsigned;      /* Pointer to NOD structure
    SCS_ID union fill;
        SYSTEMID character length 6;    /* SCS System ID
        SCS_ID_S structure fill;
            SYSTEMIDL longword unsigned;        /* Low order longword
            SYSTEMIDH word unsigned;            /* High order word
            end SCS_ID_S;
        end SCS_ID;
    FILLER      word unsigned;          /* Spare
    BPTR        longword unsigned;      /* Back pointer to main data structure
    /*
    /* Define the common header status bits.
    /* The low 16 bits are common for all data structures.
    /* The high 16 bits are data structure specific.
    /*
```

```
    STATUS      structure longword unsigned;
        LAL             bitfield mask;  /* Block is from LAL
        BRD             bitfield mask;  /* Broadcast this message
        LOG             bitfield mask;  /* Log this message
        end STATUS;

    end HDR_AGGREGATE;

/*
/* A system mailbox message contains a header.  Here we will define
/* the offsets to the fields inside that header.
/*
aggregate MSG_AGGREGATE structure prefix MSG_ fill;

    MSGTYPE     word unsigned;          /* Message type code
    REPLYMBX    word unsigned;          /* Reply mailbox unit number
    PRIVMASK1   longword unsigned;      /* First LW of privilege mask
    PRIVMASK2   longword unsigned;      /* Second LW of privilege mask
    SENDERUIC   longword unsigned;      /* UIC of request sender
    USERNAME    character length 12;    /* Pointer to start of username
    ACCOUNT     character length 8;     /* Pointer to start of account
    BASEPRI     byte unsigned;          /* Sender's base priority
    FILLER_2    byte unsigned fill;     /* **  1 spare byte  **

    end MSG_AGGREGATE;
```

```
/*
/* Define Request Context Block (RQCB) fields.
/*

aggregate RQCB_AGGREGATE structure prefix RQCB_ fill;

/*
/* common header
/*
    FLINK       longword unsigned;      /* Flink to next RQCB
    BLINK       longword unsigned;      /* Blink to last RQCB
    SIZE        word unsigned;          /* Size of data structure
    TYPE        byte unsigned;          /* Type of data structure
    FILLER_1    byte unsigned fill;     /* **  1 spare byte  **
    SEQNUM      longword unsigned;      /* RQCB sequence number
/*
/* Data above this mark is owned by the RQCB allocation routines
/*
    OVERLAY     character length 0;     /* Address where OK to change RQCB
    #overlay_mark = .;
/*
    IDENT       longword unsigned;      /* Cluster-wide ident (seqnum from original create)
    CSID        longword unsigned;      /* Cluster system id
    NOD         longword unsigned;      /* Pointer to NOD structure
    SCS_ID union fill;
        SYSTEMID character length 6;    /* SCS System ID
        SCS_ID_S structure fill;
            SYSTEMIDL longword unsigned;        /* Low order longword
            SYSTEMIDH word unsigned;            /* High order word
            end SCS_ID_S;
        end SCS_ID;
    FILLER      word unsigned;          /* Spare
    OCD         longword unsigned;      /* Backpointer to OCD
    /*
    /* Define request status bits.
    /*
    STATUS      structure longword unsigned prefix RQSTS_;
        FILLER_1        bitfield length 16 fill;
        IMPCANCEL       bitfield mask;  /* Request implicitly canceled
        end STATUS;
/*
/* Following 38 bytes are copied from the system mailbox message header
/*
    MSGTYPE     word unsigned;          /* Message type code
    REPLYMBX    word unsigned;          /* Reply mailbox unit number
    PRIVMASK1   longword unsigned;      /* First LW of privilege mask
    PRIVMASK2   longword unsigned;      /* Second LW of privilege mask
    SENDERUIC   longword unsigned;      /* UIC of request sender
    USERNAME    character length 12;    /* Pointer to start of username
    ACCOUNT     character length 8;     /* Pointer to start of account
    BASEPRI     byte unsigned;          /* Sender's base priority
    FILLER_2    byte unsigned fill;     /* **  1 spare byte  **
    RQSTCODE    byte unsigned;          /* Request type code
    SCOPE       byte unsigned;          /* Scope of request
    OPTIONS     longword unsigned;      /* Request independent options
    RQ_OPTIONS  longword unsigned;      /* Request dependent options
```

```
    ATTNMASK1   longword unsigned;      /* Attention mask part 1
    ATTNMASK2   longword unsigned;      /* Attention mask part 2
    RQSTID      longword unsigned;      /* User request identifier
    UIC         longword unsigned;      /* Requestor UIC
/*
/* End of copied area, note that offset of MCB is used to determine end of copy
/*
    MCB         longword unsigned;      /* Address of MCB
/*
    RQSTNUM     longword unsigned;      /* Request number
    USERNAMELEN word unsigned;          /* Length of username, blanks trimmed
    ACCOUNTLEN  word unsigned;          /* Length of account, blanks trimmed
    /*
    /* Define operator status bits.
    /*
    OPRSTS      structure word unsigned prefix OPRSTS_;
        TRM             bitfield mask;  /* Terminal
        REMTRM          bitfield mask;  /* Remote terminal
        MBX             bitfield mask;  /* Mailbox
        IMPDISABLE      bitfield mask;  /* Operator implicitly disabled
        end OPRSTS;

    MBXSIZE     word unsigned;          /* Mailbox buffer size
    OPER_LEN    longword unsigned;      /* Operator device name size
    OPER_PTR    longword unsigned;      /* Operator device name pointer
    TEXT_LEN    longword unsigned;      /* Optional text size
    TEXT_PTR    longword unsigned;      /* Optional text pointer
    DSBLFLINK   longword unsigned;      /* Flink to next disabled oper
    DSBLBLINK   longword unsigned;      /* Blink to last disabled oper

constant SIZE equals .;                 /* Size of RQCB in bytes
constant OVERLAY_SIZE equals .-#overlay_mark;

end RQCB_AGGREGATE;
```

```
/*
/* Define Request Control Block (RCB) fields.
/*

aggregate RCB_AGGREGATE structure prefix RCB_ fill;

/*
/* common header
/*
    FLINK       longword unsigned;      /* Flink to next RCB
    BLINK       longword unsigned;      /* Blink to last RCB
    SIZE        word unsigned;          /* Size of data structure
    TYPE        byte unsigned;          /* Type of data structure
    SCOPE       byte unsigned;          /* Scope of RCB
    SEQNUM      longword unsigned;      /* RCB sequence number
    IDENT       longword unsigned;      /* Cluster-wide ident (seqnum from original create)
    CSID        longword unsigned;      /* Cluster system id
    NOD         longword unsigned;      /* Pointer to NOD structure
    SCS_ID union fill;
        SYSTEMID character length 6;    /* SCS System ID
        SCS_ID_S structure fill;
            SYSTEMIDL longword unsigned;        /* Low order longword
            SYSTEMIDH word unsigned;            /* High order word
            end SCS_ID_S;
        end SCS_ID;
    FILLER      word unsigned;          /* Spare
    RQCB        longword unsigned;      /* Pointer to RQCB
    STATUS      longword unsigned;      /* Status longword
/*
    RQSTNUM     longword unsigned;      /* Operator request number
    TEXTLEN     longword unsigned;      /* Length of request text
    TEXTPTR     longword unsigned;      /* Address of request text

constant SIZE equals .;                 /* Size of RCB in bytes
end RCB_AGGREGATE;
```

```
/*
/* Define Operator Control Block (OCB) fields
/*

aggregate OCB_AGGREGATE structure prefix OCB_ fill;

/*
/* common header
/*
    FLINK       longword unsigned;      /* Forward link to next OCB
    BLINK       longword unsigned;      /* Backward link to last OCB
    SIZE        word unsigned;          /* Size of OCB
    TYPE        byte unsigned;          /* Type of data structure
    SCOPE       byte unsigned;          /* Scope of OCB
    SEQNUM      longword unsigned;      /* OCB sequence number
    IDENT       longword unsigned;      /* Cluster-wide ident (seqnum from original create)
    CSID        longword unsigned;      /* Cluster system id
    NOD         longword unsigned;      /* Pointer to NOD structure
    SCS_ID union fill;
        SYSTEMID character length 6;    /* SCS System ID
        SCS_ID_S structure fill;
            SYSTEMIDL longword unsigned;        /* Low order longword
            SYSTEMIDH word unsigned;            /* High order word
            end SCS_ID_S;
        end SCS_ID;
    FILLER      word unsigned;          /* Spare
    RQCB        longword unsigned;      /* Address of RQCB
    /*
    /* Define OCB status bits.
    /*
    STATUS      structure longword unsigned;
        FILLER_1        bitfield length 16 fill;
        PRMOPR          bitfield mask;  /* Operator is permanent
        MAILBOX         bitfield mask;  /* Operator is a mailbox
        end STATUS;
/*
    ATTNMASK1   longword unsigned;      /* Operator attention mask
    ATTNMASK2   longword unsigned;      /* Operator attention mask
    DEVNAMLEN   longword unsigned;      /* Operator device name length
    DEVNAMPTR   longword unsigned;      /* Operator device name string address
    BUFSIZ      longword unsigned;      /* Operator device buffer size

    constant SIZE equals .;             /* Size of OCB in bytes
    end OCB_AGGREGATE;
```

```
/*
/* Define Operator Class Descriptor (OCD) fields.
/*

aggregate OCD_AGGREGATE structure prefix OCD_ fill;

/*
/* common header
/*
    FLINK       longword unsigned;      /* Flink to first OCB
    BLINK       longword unsigned;      /* Blink to last OCB
    SIZE        word unsigned;          /* OCD size
    TYPE        byte unsigned;          /* OCD type
    SCOPE       byte unsigned;          /* Scope of OCD
    SEQNUM      longword unsigned;      /* OCD sequence number
    IDENT       longword unsigned;      /* Cluster-wide ident (seqnum from original create)
    CSID        longword unsigned;      /* Cluster system id
    NOD         longword unsigned;      /* Pointer to NOD structure
    SCS_ID union fill;
        SYSTEMID character length 6;    /* SCS System ID
        SCS_ID_S structure fill;
            SYSTEMIDL longword unsigned;        /* Low order longword
            SYSTEMIDH word unsigned;            /* High order word
            end SCS_ID_S;
        end SCS_ID;
    FILLER      word unsigned;          /* Spare
    UIC         longword unsigned;      /* UIC associated w/ this OCD
    /*
    /* Define OCD status flags.
    /*
    STATUS      structure longword unsigned;
        FILLER_1        bitfield length 16 fill;
        IMPCANCEL       bitfield mask;  /* At least one request was
                                        /* implicitly canceled.
        end STATUS;
/*
    NOTIFYMASK1 longword unsigned;      /* Operator notification mask
    NOTIFYMASK2 longword unsigned;      /* Operator notification mask
    LCB         longword unsigned;      /* Pointer to logfile control block
    FILLER_1    word unsigned fill;     /* **  2 spare bytes **
    RQSTCOUNT   word unsigned;          /* Number of outstanding requests
    RQSTFLINK   longword unsigned;      /* Flink to first request RQCB
    RQSTBLINK   longword unsigned;      /* Blink to last request RQCB
    FILLER_2    word unsigned fill;     /* **  2 spare bytes **
    OPERCOUNT   word unsigned;          /* Count of operators
    ATTNMASK1   longword unsigned;      /* Operator attention mask1
    ATTNMASK2   longword unsigned;      /* Operator attention mask2
    OPERFLINK   longword unsigned;      /* FLINK to first OCB
    OPERBLINK   longword unsigned;      /* BLINK to last OCB
    COUNTVECTOR character length 128;   /* Count vector (64 words)

    constant SIZE equals .;             /* Size of OCD in bytes
    end OCD_AGGREGATE;
```

```
/*
/* Define the cluster node information block (NOD) offsets.
/*

aggregate NOD_AGGREGATE structure prefix NOD_ fill;

/*
/* common header
/*
    FLINK       longword unsigned;      /* FLINK to next NOD
    BLINK       longword unsigned;      /* BLINK to previous NOD
    SIZE        word unsigned;          /* Size of data structure
    TYPE        byte unsigned;          /* Type of data structure
    FILL_1      byte unsigned;          /*
    SEQNUM      longword unsigned;      /* NOD sequence number
    IDENT       longword unsigned;      /* Cluster-wide ident (seqnum from original create)
    CSID        longword unsigned;      /* Cluster system id (for this host)
    NOD         longword unsigned;      /* Pointer to NOD structure
    SCS_ID union fill;
        SYSTEMID character length 6;    /* SCS System ID
        SCS_ID_S structure fill;
            SYSTEMIDL longword unsigned;        /* Low order longword
            SYSTEMIDH word unsigned;            /* High order word
            end SCS_ID_S;
        end SCS_ID;
    STATE       byte unsigned;          /* State of the node
        constant (                      /* Values for NOD_B_STATE, in form NOD_K_STATE_xxx
            STATE_LOCAL,                /* This is the local node
            STATE_START,                /* Recognized by $GETSYI, but hasn't responded to messages
            STATE_ACTIVE,               /* Responded to messages, normally functioning partner
            STATE_DEPARTED              /* Has disappeared from $GETSYI
            ) equals 1 increment 1;
        constant STATE_MAX equals NOD_K_STATE_DEPARTED;
    FILL_2      byte unsigned;          /* Spare
    FILL_3      longword unsigned;      /*
    STATUS      structure longword unsigned;    /* Status longword
        FILLER_1        bitfield length 16 fill;
        ACK_PEND        bitfield mask;  /* Waiting for acknowledgement from this node
        ACK_ATTEMPTED   bitfield mask;  /* At least one ACK has been sent to this node
        IOERR_DISPLAYED bitfield mask;  /* We have displayed an i/o error message
        NODE_LEAVING    bitfield mask;  /* Node is leaving the cluster, flush messages without signaling
        end STATUS;
/*
    NODE_CSID   longword unsigned;      /* CSID for the node (for this NOD block)
    NAME_DESC structure quadword unsigned; /* Desc for node name
        NAME_LEN  longword unsigned;    /* Name length
        NAME_PTR  longword unsigned;    /* Name address
        end NAME_DESC;
    NAME_BUF    character length 16;    /* Buffer for actual name
    SWINCARN    quadword unsigned;      /* S/W incarnation number
    SCS_ID union fill;
        NODE_SYSTEMID character length 6;       /* SCS System ID
        SCS_ID_S structure fill;
            NODE_SYSTEMIDL longword unsigned;   /* Low order longword
            NODE_SYSTEMIDH word unsigned;       /* High order word
            end SCS_ID_S;
```

```
        end SCS_ID;

    constant SIZE equals .;              /* Size of NOD in bytes
    end NOD_AGGREGATE;
```

```
/*
/* Define the Message control block (MCB) offsets.
/*

aggregate MCB_AGGREGATE structure prefix MCB_ fill;

/*
/* common header
/*
    FLINK       longword unsigned;      /* FLINK to next MCB
    BLINK       longword unsigned;      /* BLINK to previous MCB
    SIZE        word unsigned;          /* Size of data structure
    TYPE        byte unsigned;          /* Type of data structure
    SCOPE       byte unsigned;          /* Scope of MCB
    SEQNUM      longword unsigned;      /* MCB sequence number
    IDENT       longword unsigned;      /* Cluster-wide ident (seqnum from original create)
    CSID        longword unsigned;      /* Cluster system id
    NOD         longword unsigned;      /* Pointer to NOD structure
    SCS_ID union fill;
        SYSTEMID character length 6;    /* SCS System ID
        SCS_ID_S structure fill;
            SYSTEMIDL longword unsigned;        /* Low order longword
            SYSTEMIDH word unsigned;            /* High order word
            end SCS_ID_S;
        end SCS_ID;
    FILLER      word unsigned;          /* Spare
    RQCB        longword unsigned;      /* Pointer to RQCB
    STATUS      longword unsigned;      /* Status longword
/*
    MSGID       longword unsigned;      /* Message Identifier
    TEXTLEN     longword unsigned;      /* Message text length
    TEXTPTR     longword unsigned;      /* Message text address
    IOSB        longword unsigned;      /* I/O status block

    constant SIZE equals .;             /* Size of MCB in bytes
    end MCB_AGGREGATE;
```

```
/*
/* Define the BRKTHRU Output Descriptor (BOD) offsets.
/*

aggregate BOD_AGGREGATE structure prefix BOD_ fill;

/*
/* common header
/*
    FLINK       longword unsigned;      /* FLINK to next BOD
    BLINK       longword unsigned;      /* BLINK to previous BOD
    SIZE        word unsigned;          /* Size of data structure
    TYPE        byte unsigned;          /* Type of data structure
    FILL_B      byte unsigned;          /*
/*
    STATUS structure longword unsigned; /* Status longword
        DEAD            bitfield mask;  /* BRKTHRU was very slow
        WAIT            bitfield mask;  /* A REPLY /WAIT is being executed
        LOCAL_NODE      bitfield mask;  /* Target is on the local node
        SHORT_TIMEOUT   bitfield mask;  /* Use short timeout period
        end STATUS;
/*
    COMPLETION_ROUTINE address;         /* Address of I/O completion routine
    CSID        longword unsigned;      /* CSID of remote node
    NODDSC structure quadword unsigned; /* Node name descriptor
        NODLEN  longword unsigned;      /* Node name length
        NODPTR  address;                /* Node name address
        end NODDSC;
    TRMDSC structure quadword unsigned; /* Terminal name descriptor
        TRMLEN  longword unsigned;      /* Terminal name length
        TRMPTR  address;                /* Terminal name address
        end TRMDSC;
/*
/* Items for parameter list for actual call to $BRKTHRU
/*
    MSGBUF structure quadword unsigned; /* MSGBUF parameter for call
        MSGLEN  longword unsigned;      /* Message text length
        MSGPTR  address;                /* Message text address
        end MSGBUF;
    SENDTO structure quadword unsigned; /* SENDTO parameter for call
        SENLEN  longword unsigned;      /* Device name length
        SENPTR  address;                /* Device name address
        end MSGBUF;
    SNDTYP      longword unsigned;      /* Type code for send
    IOSB structure quadword unsigned;   /* I/O status block
        IOSB0   word unsigned;
        IOSB1   word unsigned;
        IOSB2   word unsigned;
        IOSB3   word unsigned;
        end IOSB;
    CARCON      longword unsigned;      /* Carriage control
    FLAGS       longword unsigned;      /* Flags longword
    REQID       longword unsigned;      /* Requestor id
/*
    QUETIME     quadword unsigned;      /* Time that $BRKTHRU issued
    SENBUF      character length 64;    /* Send name buffer
```

```
    NODBUF      character length 16;    /* Node name buffer
    TRMBUF      character length 20;    /* Terminal name buffer
/*
    constant SIZE equals .;             /* Size of BOD in bytes
    end BOD_AGGREGATE;
```

```
/*
/* Define the Cluster Output Descriptor (COD) offsets.
/*

aggregate COD_AGGREGATE structure prefix COD_ fill;

/*
/* common header
/*
    FLINK       longword unsigned;      /* FLINK to next COD
    BLINK       longword unsigned;      /* BLINK to previous COD
    SIZE        word unsigned;          /* Size of data structure
    TYPE        byte unsigned;          /* Type of data structure
    FILL_B      byte unsigned;          /*
/*
    STATUS structure longword unsigned; /* Status longword
        DEAD            bitfield mask;  /* EXE$CSP_CALL was very slow
        end STATUS;
/*
    CSID        longword unsigned;      /* CSID of remote node
    CSD         address;                /* CSD address
    NOD         address;                /* Address of the nod for the system
    ERRSTAT     longword unsigned;      /* Routine status code for error signal
    MSGBUF structure quadword unsigned; /* Message for remote node
        MSGLEN  longword unsigned;      /* Message text length
        MSGPTR  address;                /* Message text address
        end MSGBUF;
/*
    QUETIME     quadword unsigned;      /* Time that FXE$CSP_CALL issued
/*
    constant SIZE equals .;             /* Size of COD in bytes
    end COD_AGGREGATE;
```

```
/*
/* Define Structure Control Block (SCB) fields.
/* This control block contains information about all
/* data structures, and is used by the create and
/* delete data structure routines.  Each data structre
/* is represented in the SCB table by an entry of the
/* following form.  (LAL = Look Aside List)
/*

aggregate SCB_AGGREGATE structure prefix SCB_ fill;

    SIZE        word unsigned;          /* Size of data structure
    LAL_COUNT   word unsigned;          /* # of Look-aside list entries
    SEQNUM      longword unsigned;      /* Count of blocks created
    FLINK       longword unsigned;      /* Flink to first LAL entry
    BLINK       longword unsigned;      /* Blink to last LAL entry

    constant SIZE equals .;             /* Size of SCB in bytes
    end SCB_AGGREGATE;
```

```
/*
/* Define the Request Descriptor Block (RDB) fields.  Each
/* RDB structure contains some control information on a
/* particular request.  Each type of request that is known
/* by OPCOM is represented by an RDB.
/*

aggregate RDB_AGGREGATE structure prefix RDB_ fill;

    HANDLER     longword unsigned;      /* Address of request handler
    COUNT       longword unsigned;      /* Count of requests received
    OPTIONS     longword unsigned;      /* Options bit mask
    ATTNMASK1   longword unsigned;      /* Operator attention mask
    ATTNMASK2   longword unsigned;      /* Operator attention mask

    constant SIZE equals .;             /* Size of RDB in bytes
    end RDB_AGGREGATE;
```

```
/**
/*  Cluster communications messages.  These messages are sent between OPCOMs on
/* different nodes to implement cluster operations.  These messages have a short
/* header which is identical for all messages.
/*-

/*
/* Define header for all cluster messages.  This fills the first part of the message
/*

aggregate CLUSTER_MESSAGE structure prefix CLM_ fill;
    RQSTCODE            byte unsigned;          /* Same as OPC$B_RQSTCODE  (OPC$_X_CLUSMSG)
    CLM_CODE            byte unsigned;          /* Cluster-specific request code
    DS_VERSION          byte unsigned;          /* Version of data structure
    SW_VERSION          byte unsigned;          /* Version of OPCOM software
    LENGTH              word unsigned;          /* Size of structure
    fill_1              word unsigned;          /* spare word */
    CSID                longword unsigned;      /* CSID of sender

    constant SIZE equals .;

    end CLUSTER_MESSAGE;

end CLUSTER_MESSAGE;
```

```
/*
/* Define RPYBRD (REPLY broadcast) message fields.  This message is sent from REPLY command
/* to OPCOM.  OPCOM then sends the same packet to other nodes with a new request code.
/*

aggregate RPYBRD_MESSAGE structure prefix RPYBRD_ fill;

    CLM_HEADER  character length CLM_K_SIZE fill;
    constant DS_VERSION equals 6;
    /*
    OPTIONS     structure word unsigned;
        ALL             bitfield mask;          /* /ALL involved
        BELL            bitfield mask;          /* /BELLs involved
        NODE            bitfield mask;          /* /NODE
        NOTIFY          bitfield mask;          /* Notify of any actions
        SHUTDOWN        bitfield mask;          /* /SHUTDOWN
        TERMINAL        bitfield mask;          /* /TERM involved
        URGENT          bitfield mask;          /* /URGENT
        USERNAME        bitfield mask;          /* /USER involved
        WAIT            bitfield mask;          /* /WAIT, do it locally
        BROAD_LOCAL     bitfield mask;          /* Broadcast is going to local node
        BROAD_REMOTEALL bitfield mask;          /* Broadcast is going to all remotes
        BROAD_REMOTELST bitfield mask;          /* List of nodes (rpybrd_w_targ_node_len <> 0)
        LOCAL_NODE      bitfield mask;          /* Command originated on the local node
        end OPTIONS;
    fill_2              word unsigned;
    /*
    /* Length fields point into the text area at the end of the block.  Text
    /* fields are concatenated at the end, in the same order as the length
    /* fields.
    /*
    SEND_CSID           longword unsigned;      /* CSID for sending node
    SEND_TERM_LEN       word unsigned;          /* Terminal name of sender
    SEND_USER_LEN       word unsigned;          /* User name of sender
    SEND_NODE_LEN       word unsigned;          /* Name of sending node
    MESSAGE_LEN         word unsigned;          /* Length of message text
    OPTIONAL_OFF        word unsigned;          /* Offset to start of optional items
    TARG_TERM_LEN       word unsigned;          /* Terminal name(s) of target terms
    TARG_USER_LEN       word unsigned;          /* User name of target user(s)
    TARG_NODE_LEN       word unsigned;          /* Name of target node(s)
    TARG_NODE_OFF       word unsigned;          /* Offset to first node
    fill_3              word unsigned;
    /*
    /* Formatted message buffer
    /*
    FORMAT_DESC structure quadword unsigned;
        FORMAT_LEN      longword unsigned;
        FORMAT_PTR      address;
        end FORMAT_DESC;

    constant MIN_SIZE equals .;             /* Min message size

    TEXT                character length 0;     /* Text

    end RPYBRD_MESSAGE;
```

```
/*
/* Define RPYNOT (REPLY broadcast notification) message fields.  This message is sent from OPCOM
/* back to the node where a reply originated.
/*

aggregate RPYNOT_MESSAGE structure prefix RPYNOT_ fill;

    CLM_HEADER  character length CLM_K_SIZE fill;
    constant DS_VERSION equals 2;
    /*
    /* Length fields point into the text area at the end of the block.  Text
    /* fields are concatenated at the end, in the same order as the length
    /* fields.
    /*
    TERM_LEN            word unsigned;          /* Terminal name of sender
    MESSAGE_LEN         word unsigned;          /* Length of message text

    constant MIN_SIZE equals .;                 /* Min message size

    TEXT                character length 0;     /* Text

    end RPYNOT_MESSAGE;


/*
/* Define SHUT message fields.  This message is sent from OPCOM to remote nodes to
/* shut down opcom.
/*

aggregate SHUT_MESSAGE structure prefix SHUT_ fill;

    CLM_HEADER  character length CLM_K_SIZE fill;
    constant DS_VERSION equals 3;

    constant MIN_SIZE equals .;                 /* Min message size

    end SHUT_MESSAGE;
```

```
/*
/* Define CLMRQCB message fields.  This message is sent between OPCOMs.  The message
/* is essentially a copy of the OPCOM RQCB structure, except that all text strings
/* are concatenated to the end of the message (after the CLMRQCB_T_TEXT field).
/*

aggregate CLMRQCB_MESSAGE structure prefix CLMRQCB_ fill;

    /*
    /* The front part consists of the CLM header, then an entire RQCB block.
    /*
    CLM_HEADER  character length CLM_K_SIZE fill;
    constant DS_VERSION equals 2;
    RQCB_OVERLAY        character length RQCB_K_SIZE;
    /*
    /* Other information necessary to ship RQCBs between nodes
    /*
    MCB_MSGID           longword unsigned;      /* Message code from MCB
    MCB_STATUS          longword unsigned;      /* Status code from MCB
    /*
    /* Length fields point into the text area at the end of the block.  Text
    /* fields are concatenated at the end, in the same order as the length
    /* fields.
    /*

    constant MIN_SIZE equals .;                 /* Min message size

    TEXT                character length 0;     /* Text

    end CLMRQCB_MESSAGE;


/*
/* Define CLMACK message fields.  This message is sent between OPCOMs to acknowledge
/* each other.
/*

aggregate CLMACK_MESSAGE structure prefix CLMACK_ fill;

    /*
    CLM_HEADER  character length CLM_K_SIZE fill;
    constant DS_VERSION equals 2;
    /*
    CSID        longword unsigned;      /* Cluster system id
    SCS_ID union fill;
        SYSTEMID character length 6;    /* SCS System ID
        SCS_ID_S structure fill;
            SYSTEMIDL longword unsigned;        /* Low order longword
            SYSTEMIDH word unsigned;            /* High order word
            end SCS_ID_S;
        end SCS_ID;

    constant SIZE equals .;

    end CLMACK_MESSAGE;
```

```
/*+
/*  Connection manager messages.  These messages are sent from cluster
/* connection management to the OPCOM process on the local node.
/*-

/*
/* Define header for all CNXMAN messages.
/*

aggregate CNXMAN_MESSAGE structure prefix CNM_ fill;

    RQSTCODE    byte unsigned;          /* OPC$B_RQSTCODE (value OPC$_X_CNXMAN)
    CNM_CODE    byte unsigned;          /* CNXMAN-specific request code
    DS_VERSION  byte unsigned;          /* Version of data structure
    SW_VERSION  byte unsigned;          /* Version of CNXMAN software
    LENGTH      word unsigned;          /* Total length (including extensions)
    fill_1      word unsigned;          /* spare word */
    CSID        longword unsigned;      /* CSID associated with message
    SCS_ID union fill;
        SYSTEMID        character length 6;     /* 48 bit SCS System ID
        SCS_ID_S structure fill;
            SYSTEMIDL   longword unsigned;      /* Low order longword
            SYSTEMIDH   word unsigned;          /* High order word
            end SCS_ID_S;
        end SCS_ID;
    SCSNODE structure quadword unsigned;        /* Quadword buffer for node name
        SCSNODEL        longword unsigned;      /* Low order name
        SCSNODEH        longword unsigned;      /* High order name
        end SCSNODE;

constant SIZE equals .;
end CNXMAN_MESSAGE;
```

```
/*
/* Define message fields for a particular connection manager message.  The
/* CNM_B_CNM_CODE field in the header implies the appropriate message
/* extension.
/*

aggregate xxx_MESSAGE structure prefix xxx_ fill;

    CNM_HEADER  character length CNM_K_SIZE fill;
    constant DS_VERSION equals 1;                   /* Gives us xxx_K_DS_VERSION
    /*

    constant MIN_SIZE equals .;                     /* Min message size

    constant SIZE equals .;                         /* Total message size

    end xxx_MESSAGE;
```

```
end_module OPCOMDEF;
```

NPARSE
LIS

REQUEST
MAP

OPCOM

REPLY
MAP

NMLXFER
LIS

OPCCRASH
MAP

NMLV2STA
LIS

OPCOMLIB
B32

ALLOCATE
LIS

NMLZERO
LIS

OPCOMDEF
SDL

OPCOM
MAP

OPCDEFTMP
SDL

CANCEL
LIS