

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0

Page 1

_S255SDUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

1 OPCOMDEF.SDL - system definition file for OPCOM internal structures

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52

```

( 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;
```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0

Page 2

_S255SDUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

53 /*
54 /* Define bit vector names for global status
55 /*
56 constant (

```

57          SHUTDOWN_PENDING,          /* OPCOM is doing an orderly shutdown
58          LOGGING_ENABLED,           /* Log events
59          LAST_LOG_FAILED,           /* Last log attempt failed
60          LOGFILE_CLOSED,           /* Logfile closed
61          OLD_FORMAT_MSG,            /* This is an old format msg
62          TIMESTAMP_PENDING,         /* Timestamp function pending
63          BUSY,                       /* OPCOM is busy
64          FLUSH_PENDING,             /* Must flush log file
65          IN_VAXcluster              /* Member of VAXcluster
66      ) equals 0 increment 1 prefix GBLSTS_;
67
68      /*
69      /* Define event flag numbers
70      /*
71      constant (
72          ASYNCH,                     /* Throwaway for asynch i/o
73          BRKTHRU,                    /* Break through service
74          MAILBOX,                    /* Mailbox reads
75          TIME_STAMP                   /* Timer ast
76      ) equals 1 increment 1 prefix EFN_;
77
78      /*
79      /* Define the common data structure header fields.
80      /* All of the data structure definitions that follow
81      /* this one have the first few fields in common.
82      /* However, some of the fields are used differently,
83      /* depending on the type of data structure.
84      /*
85      aggregate HDR_AGGREGATE structure prefix HDR fill;
86          FLINK      longword unsigned; /* Flink to next data structure
87          BLINK      longword unsigned; /* Blink to last data structure
88          SIZE       word unsigned;     /* Size of data structure
89          TYPE       byte unsigned;     /* Type of data structure
90          SCOPE      byte unsigned;     /* Scope of data structure
91          SEQNUM     longword unsigned; /* Sequence # of data structure
92          IDENT      longword unsigned; /* Cluster-wide ident (seqnum from original create)
93          CSID       longword unsigned; /* Cluster system id
94          NOD        longword unsigned; /* Pointer to NOD structure
95          SCS_ID union fill;
96              SYSTEMID character length 6; /* SCS System ID
97              SCS_ID_S structure fill;
98                  SYSTEMIDL longword unsigned; /* Low order longword
99                  SYSTEMIDH word unsigned; /* High order word
100          end SCS_ID_S;
101      end SCS_ID;
102          FILLER     word unsigned;     /* Spare
103          BPTR       longword unsigned; /* Back pointer to main data structure
104      /*
105      /* Define the common header status bits.
106      /* The low 16 bits are common for all data structures.
107      /* The high 16 bits are data structure specific.
108      /*
109      STATUS        structure longword unsigned;

```

```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

```

```

SDL V2.0
_$255$DUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

```

Page 3

```

110          LAL      bitfield mask; /* Block is from LAL
111          BRD      bitfield mask; /* Broadcast this message
112          LOG      bitfield mask; /* Log this message
113      end STATUS;
114
115      end HDR_AGGREGATE;
116

```

```

117 /*
118 /* A system mailbox message contains a header. Here we will define
119 /* the offsets to the fields inside that header.
120 /*
121 aggregate MSG_AGGREGATE structure prefix MSG_ fill;
122
123 MSGTYPE      word unsigned;      /* Message type code
124 REPLYMBX     word unsigned;      /* Reply mailbox unit number
125 PRIVMASK1   longword unsigned;   /* First LW of privilege mask
126 PRIVMASK2   longword unsigned;   /* Second LW of privilege mask
127 SENDERUIC   longword unsigned;   /* UIC of request sender
128 USERNAME    character length 12; /* Pointer to start of username
129 ACCOUNT     character length 8;   /* Pointer to start of account
130 BASEPRI     byte unsigned;        /* Sender's base priority
131 FILLER_2    byte unsigned fill;  /* ** 1 spare byte **
132
133 end MSG_AGGREGATE;
134

```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0 Page 4
_S255SDUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

```

135 /*
136 /* Define Request Context Block (RQCB) fields.
137 /*
138
139 aggregate RQCB_AGGREGATE structure prefix RQCB_ fill;
140
141 /*
142 /* common header
143 /*
144 FLINK        longword unsigned;   /* Flink to next RQCB
145 BLINK        longword unsigned;   /* Blink to last RQCB
146 SIZE        word unsigned;        /* Size of data structure
147 TYPE        byte unsigned;        /* Type of data structure
148 FILLER_1    byte unsigned fill;   /* ** 1 spare byte **
149 SEQNUM      longword unsigned;    /* RQCB sequence number
150
151 /*
152 /* Data above this mark is owned by the RQCB allocation routines
153
154 OVERLAY     character length 0;    /* Address where OK to change RQCB
155 #overlay_mark = .;
156
157 IDENT       longword unsigned;    /* Cluster-wide ident (seqnum from original create)
158 CSID        longword unsigned;    /* Cluster system id
159 NOD         longword unsigned;    /* Pointer to NOD structure
160 SCS_ID union fill;
161   SYSTEMID  character length 6;    /* SCS System ID
162   SCS_ID_S  structure fill;
163     SYSTEMIDL longword unsigned;   /* Low order longword
164     SYSTEMIDH word unsigned;       /* High order word
165   end SCS_ID_S;
166 end SCS_ID;
167 FILLER     word unsigned;         /* Spare
168 OCD        longword unsigned;    /* Backpointer to OCD
169
170 /*
171 /* Define request status bits.
172 /*
173 STATUS     structure longword unsigned prefix RQSTS_;
174   FILLER_1  bitfield length 16 fill;
175   IMPCANCEL bitfield mask; /* Request implicitly canceled
176 end STATUS;
177
178 /*
179 /* Following 38 bytes are copied from the system mailbox message header

```

```

177 /*
178 MSGTYPE      word unsigned;      /* Message type code
179 REPLYMBX     word unsigned;      /* Reply mailbox unit number
180 PRIVMASK1   longword unsigned;  /* First LW of privilege mask
181 PRIVMASK2   longword unsigned;  /* Second LW of privilege mask
182 SENDERUIC   longword unsigned;  /* UIC of request sender
183 USERNAME    character length 12; /* Pointer to start of username
184 ACCOUNT     character length 8;  /* Pointer to start of account
185 BASEPRI     byte unsigned;       /* Sender's base priority
186 FILLER_2    byte unsigned fill; /* ** 1 spare byte **
187 RQSTCODE    byte unsigned;       /* Request type code
188 SCOPE       byte unsigned;       /* Scope of request
189 OPTIONS     longword unsigned;   /* Request independent options
190 RQ_OPTIONS  longword unsigned;   /* Request dependent options
191 ATTNMASK1   longword unsigned;   /* Attention mask part 1

```

15-SEP-1984 23:06:07.75

SDL V2.0

Page

5

15-SEP-1984 22:48:01

_S255\$DUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

```

192 ATTNMASK2   longword unsigned;   /* Attention mask part 2
193 RQSTID     longword unsigned;   /* User request identifier
194 UIC       longword unsigned;   /* Requestor UIC
195 /*
196 /* End of copied area, note that offset of MCB is used to determine end of copy
197 /*
198 MCB       longword unsigned;   /* Address of MCB
199 /*
200 RQSTNUM   longword unsigned;   /* Request number
201 USERNAMELEN word unsigned;     /* Length of username, blanks trimmed
202 ACCOUNTLEN word unsigned;     /* Length of account, blanks trimmed
203 /*
204 /* Define operator status bits.
205 /*
206 OPRSTS    structure word unsigned prefix OPRSTS_;
207     TRM     bitfield mask;      /* Terminal
208     REMTRM  bitfield mask;      /* Remote terminal
209     MBX     bitfield mask;      /* Mailbox
210     IMPDISABLE bitfield mask;  /* Operator implicitly disabled
211     end OPRSTS;
212
213 MBXSIZE   word unsigned;        /* Mailbox buffer size
214 OPER_LEN  longword unsigned;   /* Operator device name size
215 OPER_PTR  longword unsigned;   /* Operator device name pointer
216 TEXT_LEN  longword unsigned;   /* Optional text size
217 TEXT_PTR  longword unsigned;   /* Optional text pointer
218 DSBLFLINK longword unsigned;   /* Flink to next disabled oper
219 DSBLBLINK longword unsigned;   /* Blink to last disabled oper
220
221 constant SIZE equals .;        /* Size of RQCB in bytes
222 constant OVERLAY_SIZE equals .-#overlay_mark;
223
224 end RQCB_AGGREGATE;
225

```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0

Page

6

_S255\$DUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

```

226 /*
227 /* Define Request Control Block (RCB) fields.
228 /*
229 /*
230 aggregate RCB_AGGREGATE structure prefix RCB_ fill;
231 /*
232 /*
233 /* common header

```

```

234 /*
235 FLINK      longword unsigned; /* Flink to next RCB
236 BLINK      longword unsigned; /* Blink to last RCB
237 SIZE       word unsigned;     /* Size of data structure
238 TYPE       byte unsigned;     /* Type of data structure
239 SCOPE      byte unsigned;     /* Scope of RCB
240 SEQNUM     longword unsigned; /* RCB sequence number
241 IDENT      longword unsigned; /* Cluster-wide ident (seqnum from original create)
242 CSID       longword unsigned; /* Cluster system id
243 NOD        longword unsigned; /* Pointer to NOD structure
244 SCS_ID union fill;
245     _SYSTEMID character length 6; /* SCS System ID
246     SCS_ID_S structure fill;
247     _SYSTEMIDL longword unsigned; /* Low order longword
248     _SYSTEMIDH word unsigned;     /* High order word
249     end SCS_ID_S;
250 end SCS_ID;
251 FILLER     word unsigned;     /* Spare
252 RQCB       longword unsigned; /* Pointer to RQCB
253 STATUS     longword unsigned; /* Status longword
254 /*
255 RQSTNUM    longword unsigned; /* Operator request number
256 TEXTLEN   longword unsigned; /* Length of request text
257 TEXTPTR   longword unsigned; /* Address of request text
258
259 constant SIZE equals .; /* Size of RCB in bytes
260 end RCB_AGGREGATE;
261

```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0 Page 7
_S255\$DUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

```

262 /*
263 /* Define Operator Control Block (OCB) fields
264 /*
265
266 aggregate OCB_AGGREGATE structure prefix OCB_ fill;
267
268 /*
269 /* common header
270 /*
271 FLINK      longword unsigned; /* Forward link to next OCB
272 BLINK      longword unsigned; /* Backward link to last OCB
273 SIZE       word unsigned;     /* Size of OCB
274 TYPE       byte unsigned;     /* Type of data structure
275 SCOPE      byte unsigned;     /* Scope of OCB
276 SEQNUM     longword unsigned; /* OCB sequence number
277 IDENT      longword unsigned; /* Cluster-wide ident (seqnum from original create)
278 CSID       longword unsigned; /* Cluster system id
279 NOD        longword unsigned; /* Pointer to NOD structure
280 SCS_ID union fill;
281     _SYSTEMID character length 6; /* SCS System ID
282     SCS_ID_S structure fill;
283     _SYSTEMIDL longword unsigned; /* Low order longword
284     _SYSTEMIDH word unsigned;     /* High order word
285     end SCS_ID_S;
286 end SCS_ID;
287 FILLER     word unsigned;     /* Spare
288 RQCB       longword unsigned; /* Address of RQCB
289 /*
290 /* Define OCB status bits.
291 /*
292 STATUS     structure longword unsigned;
293     FILLER_1 bitfield length 16 fill;

```

```

294          PRMOPR          bitfield mask; /* Operator is permanent
295          MAILBOX         bitfield mask; /* Coerator is a mailbox
296          end STATUS;
297 /*
298          ATTNMASK1        longword unsigned; /* Operator attention mask
299          ATTNMASK2        longword unsigned; /* Operator attention mask
300          DEVNAMLEN        longword unsigned; /* Operator device name length
301          DEVNAMPTR        longword unsigned; /* Operator device name string address
302          BUFSIZ           longword unsigned; /* Operator device buffer size
303
304          constant SIZE equals .; /* Size of OCB in bytes
305          end OCB_AGGREGATE;
306

```

```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

```

```

SDL V2.0 Page 8
_$255$DUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

```

```

307 /*
308 /* Define Operator Class Descriptor (OCD) fields.
309 /*
310
311 aggregate OCD_AGGREGATE structure prefix OCD_ fill;
312
313 /*
314 /* common header
315 /*
316          FLINK            longword unsigned; /* Flink to first OCB
317          BLINK            longword unsigned; /* Blink to last OCB
318          SIZE              word unsigned; /* OCD size
319          TYPE              byte unsigned; /* OCD type
320          SCOPE             byte unsigned; /* Scope of OCD
321          SEQNUM            longword unsigned; /* OCD sequence number
322          IDENT             longword unsigned; /* Cluster-wide ident (seqnum from original create)
323          CSID              longword unsigned; /* Cluster system id
324          NOD               longword unsigned; /* Pointer to NOD structure
325          SCS_ID union fill;
326          -SYSTEMID character length 6; /* SCS System ID
327          SCS_ID_S structure fill;
328          -SYSTEMIDL longword unsigned; /* Low order longword
329          SYSTEMIDH word unsigned; /* High order word
330          end SCS_ID_S;
331          end SCS_ID;
332          FILLER            word unsigned; /* Spare
333          UIC               longword unsigned; /* UIC associated w/ this OCD
334          /*
335          /* Define OCD status flags.
336          /*
337          STATUS            structure longword unsigned;
338          FILLER_1          bitfield length 16 fill;
339          IMPCANCEL         bitfield mask; /* At least one request was
340                                /* implicitly canceled.
341          end STATUS;
342 /*
343          NOTIFYMASK1       longword unsigned; /* Operator notification mask
344          NOTIFYMASK2       longword unsigned; /* Operator notification mask
345          LCB               longword unsigned; /* Pointer to logfile control block
346          FILLER_1          word unsigned fill; /* ** 2 spare bytes **
347          RQSTCOUNT         word unsigned; /* Number of outstanding requests
348          RQSTFLINK         longword unsigned; /* Flink to first request RQCB
349          RQSTBLINK         longword unsigned; /* Blink to last request RQCB
350          FILLER_2          word unsigned fill; /* ** 2 spare bytes **
351          OPERCOUNT         word unsigned; /* Count of operators
352          ATTNMASK1         longword unsigned; /* Operator attention mask1
353          ATTNMASK2         longword unsigned; /* Operator attention mask2

```



```

354 OPERFLINK longword unsigned; /* FLINK to first OCB
355 OPERBLINK longword unsigned; /* BLINK to last OCB
356 COUNTVECTOR character length 128; /* Count vector (64 words)
357
358 constant SIZE equals .; /* Size of OCD in bytes
359 end OCD_AGGREGATE;
360

```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0 Page 9
_S255SDUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

```

361 /*
362 /* Define the cluster node information block (NOD) offsets.
363 /*
364
365 aggregate NOD_AGGREGATE structure prefix NOD_ fill;
366
367 /*
368 /* common header
369 /*
370 FLINK longword unsigned; /* FLINK to next NOD
371 BLINK longword unsigned; /* BLINK to previous NOD
372 SIZE word unsigned; /* Size of data structure
373 TYPE byte unsigned; /* Type of data structure
374 FILL_1 byte unsigned; /*
375 SEQNUM longword unsigned; /* NOD sequence number
376 IDENT longword unsigned; /* Cluster-wide ident (seqnum from original create)
377 CSID longword unsigned; /* Cluster system id (for this host)
378 NOD longword unsigned; /* Pointer to NOD structure
379 SCS_ID union fill;
380 -SYSTEMID character length 6; /* SCS System ID
381 SCS_ID_S structure fill;
382 -SYSTEMIDL longword unsigned; /* Low order longword
383 -SYSTEMIDH word unsigned; /* High order word
384 end SCS_ID_S;
385 end SCS_ID;
386 STATE byte unsigned; /* State of the node
387 constant ( /* Values for NOD_B_STATE, in form NOD_K_STATE_XXX
388 STATE_LOCAL, /* This is the local node
389 STATE_START, /* Recognized by $GETSYI, but hasn't responded to messages
390 STATE_ACTIVE, /* Responded to messages, normally functioning partner
391 STATE_DEPARTED /* Has disappeared from $GETSYI
392 ) equals 1 increment 1;
393 constant STATE_MAX equals NOD_K_STATE_DEPARTED;
394 FILL_2 byte unsigned; /* Spare
395 FILL_3 longword unsigned; /*
396 STATUS structure longword unsigned; /* Status longword
397 FILLER_1 bitfield length 16 fill;
398 ACK_PEND bitfield mask; /* Waiting for acknowledgement from this node
399 ACK_ATTEMPTED bitfield mask; /* At least one ACK has been sent to this node
400 IOERR_DISPLAYED bitfield mask; /* We have displayed an i/o error message
401 NODE_LEAVING bitfield mask; /* Node is leaving the cluster, flush messages without signaling
402 end STATUS;
403 /*
404 NODE_CSID longword unsigned; /* CSID for the node (for this NOD block)
405 NAME_DESC structure quadword unsigned; /* Desc for node name
406 NAME_LEN longword unsigned; /* Name length
407 NAME_PTR longword unsigned; /* Name address
408 end NAME_DESC;
409 NAME_BUF character length 16; /* Buffer for actual name
410 SWINCARN quadword unsigned; /* S/W incarnation number
411 SCS_ID union fill;
412 -NODE_SYSTEMID character length 6; /* SCS System ID
413 SCS_ID_S structure fill;

```

```

414         NODE_SYSTEMIDL longword unsigned; /* Low order longword
415         NODE_SYSTEMIDH word unsigned; /* High order woru
416     end SCS_ID_S;
417 end SCS_ID;

```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0 Page 10
_S255SDUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

```

418
419     constant SIZE equals .; /* Size of NOD in bytes
420     end NOD_AGGREGATE;
421

```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0 Page 11
_S255SDUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

```

422 /*
423 /* Define the Message control block (MCB) offsets.
424 /*

```

```

425 aggregate MCB_AGGREGATE structure prefix MCB_ fill;

```

```

426 /*
427 /* common header
428 /*

```

```

429     FLINK      longword unsigned; /* FLINK to next MCB
430     BLINK      longword unsigned; /* BLINK to previous MCB
431     SIZE       word unsigned; /* Size of data structure
432     TYPE       byte unsigned; /* Type of data structure
433     SCOPE      byte unsigned; /* Scope of MCB
434     SEQNUM     longword unsigned; /* MCB sequence number
435     IDENT      longword unsigned; /* Cluster-wide ident (seqnum from original create)
436     CSID       longword unsigned; /* Cluster system id
437     NOD        longword unsigned; /* Pointer to NOD structure
438     SCS_ID union fill;
439     SYSTEMID character length 6; /* SCS System ID
440     SCS_ID_S structure fill;
441     SYSTEMIDL longword unsigned; /* Low order longword
442     SYSTEMIDH word unsigned; /* High order word
443     end SCS_ID_S;
444     end SCS_ID;
445     FILLER     word unsigned; /* Spare
446     RQCB       longword unsigned; /* Pointer to RQCB
447     STATUS     longword unsigned; /* Status longword
448 /*
449     MSGID      longword unsigned; /* Message Identifier
450     TEXTLEN    longword unsigned; /* Message text length
451     TEXTPTR    longword unsigned; /* Message text address
452     IOSB       longword unsigned; /* I/O status block

```

```

453     constant SIZE equals .; /* Size of MCB in bytes
454     end MCB_AGGREGATE;
455

```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0 Page 12
_S255SDUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

```

456 /*
457 /* Define the BRKTHRU Output Descriptor (BOD) offsets.
458 /*

```

```

459 aggregate BOD_AGGREGATE structure prefix BOD_ fill;

```

```

460 /*
461 /* common header
462 /*

```

```

463

```

```

468 FLINK      longword unsigned; /* FLINK to next BOD
469 BLINK      longword unsigned; /* BLINK to previous BOD
470 SIZE       word unsigned; /* Size of data structure
471 TYPE       byte unsigned; /* Type of data structure
472 FILL_B     byte unsigned; /*
473 /*
474 STATUS structure longword unsigned; /* Status longword
475     DEAD      bitfield mask; /* BRKTHRU was very slow
476     WAIT      bitfield mask; /* A REPLY /WAIT is being executed
477     LOCAL_NODE bitfield mask; /* Target is on the local node
478     SHORT_TIMEOUT bitfield mask; /* Use short timeout period
479     end STATUS;
480 /*
481 COMPLETION_ROUTINE address; /* Address of I/O completion routine
482 CSID        longword unsigned; /* CSID of remote node
483 NODDSC structure quadword unsigned; /* Node name descriptor
484     NODLEN    longword unsigned; /* Node name length
485     NODPTR    address; /* Node name address
486     end NODDSC;
487 TRMDSC structure quadword unsigned; /* Terminal name descriptor
488     TRMLEN    longword unsigned; /* Terminal name length
489     TRMPTR    address; /* Terminal name address
490     end TRMDSC;
491 /*
492 /* Items for parameter list for actual call to $BRKTHRU
493 /*
494 MSGBUF structure quadword unsigned; /* MSGBUF parameter for call
495     MSGLEN    longword unsigned; /* Message text length
496     MSGPTR    address; /* Message text address
497     end MSGBUF;
498 SENDTO structure quadword unsigned; /* SENDTO parameter for call
499     SENLEN    longword unsigned; /* Device name length
500     SENPTR    address; /* Device name address
501     end MSGBUF;
502 SNDTYP      longword unsigned; /* Type code for send
503 IOSB structure quadword unsigned; /* I/O status block
504     IOSB0     word unsigned;
505     IOSB1     word unsigned;
506     IOSB2     word unsigned;
507     IOSB3     word unsigned;
508     end IOSB;
509 CARCON      longword unsigned; /* Carriage control
510 FLAGS       longword unsigned; /* Flags longword
511 REQID       longword unsigned; /* Requestor id
512 /*
513 QUETIME     quadword unsigned; /* Time that $BRKTHRU issued
514 SENBUF      character length 64; /* Send name buffer
515 NODBUF      character length 16; /* Node name buffer

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

516 TRMBUF     character length 20; /* Terminal name buffer
517 /*
518 constant SIZE equals .; /* Size of BOD in bytes
519 end BOD_AGGREGATE;
520

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

521 /*
522 /* Define the Cluster Output Descriptor (COD) offsets.
523 /*
524

```

SDL V2.0 Page 13
 _\$255\$DUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

SDL V2.0 Page 14
 _\$255\$DUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

```

525 aggregate COD_AGGREGATE structure prefix COD_ fill;
526
527 /*
528 /* common header
529 /*
530     FLINK      longword unsigned;    /* FLINK to next COD
531     BLINK      longword unsigned;    /* BLINK to previous COD
532     SIZE       word unsigned;        /* Size of data structure
533     TYPE       byte unsigned;        /* Type of data structure
534     FILL_B     byte unsigned;        /*
535 /*
536     STATUS structure longword unsigned; /* Status longword
537     DEAD       bitfield mask;        /* EXECSP_CALL was very slow
538     end STATUS;
539 /*
540     CSID       longword unsigned;    /* CSID of remote node
541     CSD        address;              /* CSD address
542     NOD        address;              /* Address of the nod for the system
543     ERRSTAT    longword unsigned;    /* Routine status code for error signal
544     MSGBUF structure quadword unsigned; /* Message for remote node
545     MSGLEN     longword unsigned;    /* Message text length
546     MSGPTR     address;              /* Message text address
547     end MSGBUF;
548 /*
549     QUETIME    quadword unsigned;    /* Time that EXECSP_CALL issued
550 /*
551     constant SIZE equals .;          /* Size of COD in bytes
552     end COD_AGGREGATE;
553

```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0 Page 15
_S255SDUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

```

554 /*
555 /* Define Structure Control Block (SCB) fields.
556 /* This control block contains information about all
557 /* data structures, and is used by the create and
558 /* delete data structure routines. Each data structure
559 /* is represented in the SCB table by an entry of the
560 /* following form. (LAL = Look Aside List)
561 /*

```

```

562
563 aggregate SCB_AGGREGATE structure prefix SCB_ fill;
564
565     SIZE       word unsigned;        /* Size of data structure
566     LAL_COUNT  word unsigned;        /* # of Look-aside list entries
567     SEQNUM     longword unsigned;    /* Count of blocks created
568     FLINK      longword unsigned;    /* Flink to first LAL entry
569     BLINK      longword unsigned;    /* Blink to last LAL entry
570
571     constant SIZE equals .;          /* Size of SCB in bytes
572     end SCB_AGGREGATE;
573

```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0 Page 16
_S255SDUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

```

574 /*
575 /* Define the Request Descriptor Block (RDB) fields. Each
576 /* RDB structure contains some control information on a
577 /* particular request. Each type of request that is known
578 /* by OPCOM is represented by an RDB.
579 /*

```

```

580
581 aggregate RDB_AGGREGATE structure prefix RDB_ fill;

```

```

582
583 HANDLER      longword unsigned;    /* Address of request handler
584 COUNT       longword unsigned;    /* Count of requests received
585 OPTIONS     longword unsigned;    /* Options bit mask
586 ATTNMASK1   longword unsigned;    /* Operator attention mask
587 ATTNMASK2   longword unsigned;    /* Operator attention mask
588
589 constant SIZE equals .;           /* Size of RDB in bytes
590 end RDB_AGGREGATE;
591

```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0

Page 17

_S255SDUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

```

592 /*+
593 /* Cluster communications messages. These messages are sent between OPCOMs on
594 /* different nodes to implement cluster operations. These messages have a short
595 /* header which is identical for all messages.
596 /*-
597
598 /*
599 /* Define header for all cluster messages. This fills the first part of the message
600 /*
601 aggregate CLUSTER_MESSAGE structure prefix CLM_ fill;
602
603     RQSTCODE      byte unsigned;      /* Same as OPC$B_RQSTCODE (OPC$_X_CLUSMSG)
604
605     CLM_CODE      byte unsigned;      /* Cluster-specific request code
606
607     DS_VERSION    byte unsigned;      /* Version of data structure
608
609     SW_VERSION    byte unsigned;      /* Version of OPCOM software
610
611     LENGTH        word unsigned;     /* Size of structure
612
613     fill_1        word unsigned;     /* spare word */
614
615     CSID          longword unsigned;  /* CSID of sender
616
617
618 constant SIZE equals .;
619
620
621 end CLUSTER_MESSAGE;
622
623

```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0

Page 18

_S255SDUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

```

624 /*
625 /* Define RPYBRD (REPLY broadcast) message fields. This message is sent from REPLY command
626 /* to OPCOM. OPCOM then sends the same packet to other nodes with a new request code.
627 /*
628
629 aggregate RPYBRD_MESSAGE structure prefix RPYBRD_ fill;
630
631     CLM_HEADER character length CLM_K_SIZE fill;
632     constant DS_VERSION equals 6;
633     /*
634     OPTIONS      structure word unsigned;
635         ALL      bitfield mask;      /* /ALL involved
636         BELL     bitfield mask;      /* /BELLS involved
637         NODE     bitfield mask;      /* /NODE
638         NOTIFY   bitfield mask;      /* Notify of any actions

```

```

639          SHUTDOWN          bitfield mask;          /* /SHUTDOWN
640          TERMINAL          bitfield mask;          /* /TERM involved
641          URGENT          bitfield mask;          /* /URGENT
642          USERNAME          bitfield mask;          /* /USER involved
643          WAIT          bitfield mask;          /* /WAIT, do it locally
644          BROAD_LOCAL          bitfield mask;          /* Broadcast is going to local node
645          BROAD_REMOTEALL          bitfield mask;          /* Broadcast is going to all remotes
646          BROAD_REMOTEALL          bitfield mask;          /* List of nodes (rpybrd_w_targ_node_len <> 0)
647          LOCAL_NODE          bitfield mask;          /* Command originated on the local node
648          end OPTIONS;
649          fill_2          word unsigned;
650          /*
651          /* Length fields point into the text area at the end of the block. Text
652          /* fields are concatenated at the end, in the same order as the length
653          /* fields.
654          /*
655          SEND_CSID          longword unsigned;          /* CSID for sending node
656          SEND_TERM_LEN          word unsigned;          /* Terminal name of sender
657          SEND_USER_LEN          word unsigned;          /* User name of sender
658          SEND_NODE_LEN          word unsigned;          /* Name of sending node
659          MESSAGE_LEN          word unsigned;          /* Length of message text
660          OPTIONAL_OFF          word unsigned;          /* Offset to start of optional items
661          TARG_TERM_LEN          word unsigned;          /* Terminal name(s) of target terms
662          TARG_USER_LEN          word unsigned;          /* User name of target user(s)
663          TARG_NODE_LEN          word unsigned;          /* Name of target node(s)
664          TARG_NODE_OFF          word unsigned;          /* Offset to first node
665          fill_3          word unsigned;
666          /*
667          /* Formatted message buffer
668          /*
669          FORMAT_DESC          structure quadword unsigned;
670          FORMAT_LEN          longword unsigned;
671          FORMAT_PTR          address;
672          end FORMAT_DESC;
673
674          constant MIN_SIZE          equals .;          /* Min message size
675
676          TEXT          character length 0;          /* Text
677
678          end RPYBRD_MESSAGE;
679

```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0
_S255SDUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

Page 19

```

680          /*
681          /* Define RPYNOT (REPLY broadcast notification) message fields. This message is sent from OPCOM
682          /* back to the node where a reply originated.
683          /*
684
685          aggregate RPYNOT_MESSAGE          structure prefix RPYNOT_ fill;
686
687          CLM_HEADER          character length CLM_K_SIZE fill;
688          constant DS_VERSION          equals 2;
689          /*
690          /* Length fields point into the text area at the end of the block. Text
691          /* fields are concatenated at the end, in the same order as the length
692          /* fields.
693          /*
694          TERM_LEN          word unsigned;          /* Terminal name of sender
695          MESSAGE_LEN          word unsigned;          /* Length of message text
696
697          constant MIN_SIZE          equals .;          /* Min message size
698

```

```
699     TEXT                character length 0;    /* Text      C 6
700
701     end RPYNOT_MESSAGE;
702
703
704 /*
705 /* Define SHUT message fields. This message is sent from OPCOM to remote nodes to
706 /* shut down opcom.
707 /*
708
709 aggregate SHUT_MESSAGE structure prefix SHUT_ fill;
710
711     CLM_HEADER character length CLM_K_SIZE fill;
712     constant DS_VERSION equals 3;
713
714     constant MIN_SIZE equals .;                /* Min message size
715
716     end SHUT_MESSAGE;
717
718 /*
719 /* Define CLMRQCB message fields. This message is sent between OPCOMs. The message
720 /* is essentially a copy of the OPCOM RQCB structure, except that all text strings
721 /* are concatenated to the end of the message (after the CLMRQCB_T_TEXT field).
722 /*
723
724 aggregate CLMRQCB_MESSAGE structure prefix CLMRQCB_ fill;
725
726 /*
727 /* The front part consists of the CLM header, then an entire RQCB block.
728 /*
729     CLM_HEADER character length CLM_K_SIZE fill;
730     constant DS_VERSION equals 2;
731     RQCB_OVERLAY character length RQCB_K_SIZE;
732 /*
733 /* Other information necessary to ship RQCBs between nodes
734 /*
735     MCB_MSGID          longword unsigned;      /* Message code from MCB
736     MCB_STATUS         longword unsigned;      /* Status code from MCB
737 /*
738 /* Length fields point into the text area at the end of the block. Text
739 /* fields are concatenated at the end, in the same order as the length
740 /* fields.
741 /*
742
743     constant MIN_SIZE equals .;                /* Min message size
744
745     TEXT                character length 0;    /* Text
746
747     end CLMRQCB_MESSAGE;
748
749
750 /*
751 /* Define CLMACK message fields. This message is sent between OPCOMs to acknowledge
752 /* each other.
753 /*
754
755 aggregate CLMACK_MESSAGE structure prefix CLMACK_ fill;
756
757 /*
758     CLM_HEADER character length CLM_K_SIZE fill;
```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0
_S255SDUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

Page 20

```

759 constant DS_VERSION equals 2;
760 /*
761 CSID      longword unsigned;      /* Cluster system id
762 SCS_ID union fill;
763   SYSTEMID character length 6;    /* SCS System ID
764   SCS_ID_S structure fill;
765   SYSTEMIDL longword unsigned;   /* Low order longword
766   SYSTEMIDH word unsigned;      /* High order word
767   end SCS_ID_S;
768 end SCS_ID;
769
770 constant SIZE equals .;
771
772 end CLMACK_MESSAGE;
773

```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0 Page 21
_S255SDUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

```

774 /*+
775 /* Connection manager messages. These messages are sent from cluster
776 /* connection management to the OPCOM process on the local node.
777 /*-
778
779 /*
780 /* Define header for all CNXMAN messages.
781 /*
782
783 aggregate CNXMAN_MESSAGE structure prefix CNM_ fill;
784
785 RQSTCODE  byte unsigned;          /* OPC$B RQSTCODE (value OPC$ X_CNXMAN)
786 CNM_CODE  byte unsigned;          /* CNXMAN-specific request code
787 DS_VERSION byte unsigned;          /* Version of data structure
788 SW_VERSION byte unsigned;          /* Version of CNXMAN software
789 LENGTH    word unsigned;          /* Total length (including extensions)
790 fill_1    word unsigned;          /* spare word */
791 CSID      longword unsigned;      /* CSID associated with message
792 SCS_ID union fill;
793   SYSTEMID character length 6;    /* 48 bit SCS System ID
794   SCS_ID_S structure fill;
795   SYSTEMIDL longword unsigned;   /* Low order longword
796   SYSTEMIDH word unsigned;      /* High order word
797   end SCS_ID_S;
798 end SCS_ID;
799 SCSNODE structure quadword unsigned; /* Quadword buffer for node name
800   SCSNODEL longword unsigned;     /* Low order name
801   SCSNODEH longword unsigned;     /* High order name
802 end SCSNODE;
803
804 constant SIZE equals .;
805
806 end CNXMAN_MESSAGE;
807

```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0 Page 22
_S255SDUA28:[OPCOM.SRC]OPCOMDEF.SDL;1

```

808 /*
809 /* Define message fields for a particular connection manager message. The
810 /* CNM_B_CNМ_CODE field in the header implies the appropriate message
811 /* extension.
812 /*
813
814 aggregate xxx_MESSAGE structure prefix xxx_ fill;
815

```



```

816 CNM_HEADER character length CNM_K_SIZE fill;
817 constant DS_VERSION equals 1; /* Gives us xxx_K_DS_VERSION
818 /*
819
820 constant MIN_SIZE equals .; /* Min message size
821
822 constant SIZE equals .; /* Total message size
823
824 end xxx_MESSAGE;
825

```

15-SEP-1984 23:06:07.75
15-SEP-1984 22:48:01

SDL V2.0 Page 23
_\$255\$DUA28:LOPCOM.SRC\OPCOMDEF.SDL;1

```

826 end_module OPCOMDEF;
827

```



