


```

000000  PPPPPPP  CCCCCCCC  000000  MM      MM  RRRRRRRR  PPPPPPPP  LL      YY      YY
000000  PPPPPPP  CCCCCCCC  000000  MM      MM  RRRRRRRR  PPPPPPPP  LL      YY      YY
00      00  PP      PP  CC      CC      00      00  MMMM  MMMM  RR      RR  PP      PP  LL      YY      YY
00      00  PP      PP  CC      CC      00      00  MMMM  MMMM  RR      RR  PP      PP  LL      YY      YY
00      00  PP      PP  CC      CC      00      00  MM   MM   MM   RR      RR  PP      PP  LL      YY      YY
00      00  PP      PP  CC      CC      00      00  MM   MM   MM   RR      RR  PP      PP  LL      YY      YY
00      00  PPPPPPPP  CC      CC      00      00  MM   MM   MM   RRRRRRRR  PPPPPPPP  LL      YY      YY
00      00  PPPPPPPP  CC      CC      00      00  MM   MM   MM   RRRRRRRR  PPPPPPPP  LL      YY      YY
00      00  PP      PP  CC      CC      00      00  MM   MM   MM   RR   RR  PP      PP  LL      YY      YY
00      00  PP      PP  CC      CC      00      00  MM   MM   MM   RR   RR  PP      PP  LL      YY      YY
00      00  PP      PP  CC      CC      00      00  MM   MM   MM   RR   RR  PP      PP  LL      YY      YY
00      00  PP      PP  CC      CC      00      00  MM   MM   MM   RR   RR  PP      PP  LL      YY      YY
000000  PP      PP  CC      CC      000000  MM   MM   RR   RR  PP      PP  LL      YY      YY
000000  PP      PP  CCCCCCCC  CCCCCCCC  000000  MM   MM   RR   RR  PP      PP  LL      YY      YY
000000  PP      PP  CCCCCCCC  CCCCCCCC  000000  MM   MM   RR   RR  PP      PP  LL      YY      YY

```

```

LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SSSSSS
LL      II     SSSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SS
LLLLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLLLL IIIIII  SSSSSSSS

```

1
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
53
54
55
56
57

```

0001 0 MODULE OPC$OPCOMPLY (
0002 0
0003 0     LANGUAGE (BLISS32),
0004 0     IDENT = 'V04-000'
0005 0 ) =
0006 0
0007 0 *****
0008 0 *
0009 0 *   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 0 *   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 0 *   ALL RIGHTS RESERVED.
0012 0 *
0013 0 *   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 0 *   ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 0 *   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 0 *   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 0 *   OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 0 *   TRANSFERRED.
0019 0 *
0020 0 *   THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 0 *   AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 0 *   CORPORATION.
0023 0 *
0024 0 *   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 0 *   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 0 *
0027 0 *****
0028 0
0029 0 ++
0030 0 FACILITY:
0031 0
0032 0     OPCOM
0033 0
0034 0 ABSTRACT:
0035 0
0036 0     This module contains the specialized logic to service
0037 0     a particular type of request sent by a user to OPCOM.
0038 0
0039 0 Environment:
0040 0
0041 0     VAX/VMS operating system.
0042 0
0043 0 Author:
0044 0
0045 0     Steven T. Jeffreys
0046 0
0047 0 Creation date:
0048 0
0049 0     March 10, 1981
0050 0
0051 0 Revision history:
0052 0
0053 0     V03-003 CWH3003          CW Hobbs          16-Sep-1983
0054 0     Use VM jacket routines.
0055 0
0056 0     V03-002 CWH3002          CW Hobbs          30-Jul-1983
0057 0     Various and sundry things to make OPCOM distributed

```

```

58      0058 0 | across the cluster.
59      0059 0 |
60      0060 0 | V03-001  STJ49918  Steven T. Jeffreys, 13-Oct-1982
61      0061 0 | Honor replies to a request even if the replier does not
62      0062 0 | have OPER privilege, as long as the reply originated from
63      0063 0 | an operator terminal attached to a process with the same
64      0064 0 | UIC as the request.
65      0065 0 |
66      0066 0 | V02-004  STJ0163  Steven T. Jeffreys, 08-Feb-1982
67      0067 0 | Make references to library routines use general addressing mode.
68      0068 0 |
69      0069 0 | V02-003  STJ0078  Steven T. Jeffreys, 01-Aug-1981
70      0070 0 | Added support for INITIALIZE and BLANK replies to the
71      0071 0 | mag. tape ACP's requests.
72      0072 0 |
73      0073 0 | V02-002  STJ0043  Steven T. Jeffreys, 23-May-1981
74      0074 0 | Changed code to use new RQSTCPLTE, RQSTABORT and RQSTPEND
75      0075 0 | message formats. Note corresponding change to OPCMSG.MSG.
76      0076 0 |
77      0077 0 | --
78      0078 0 |
79      0079 1 | BEGIN | Start of OPCOMRPLY
80      0080 1 |
81      0081 1 | LIBRARY 'SYSS$LIBRARY:LIB.L32';
82      0082 1 | LIBRARY 'LIBS:OPCOMLIB';
83      0083 1 |
84      0084 1 | FORWARD ROUTINE
85      0085 1 | REPLY_HANDLER : NOVALUE,
86      0086 1 | REPLY_CLM_HANDLER : NOVALUE;
87      0087 1 |
88      0088 1 | BUILTIN
89      0089 1 | INSQUE, | Insert entry onto a queue
90      0090 1 | REMQUE; | Remove entry from a queue
91      0091 1 |
92      0092 1 | EXTERNAL ROUTINE
93      0093 1 | CHECK_REQUEST, | Common sanity checks
94      0094 1 | CLUSMSG_CONV_CLM_RQCB, | Convert cluster message to RQCB
95      0095 1 | CLUSMSG_RQCB_SEND, | Send RQCB to remote nodes
96      0096 1 | DEALLOCATE_RQCB : NOVALUE, | Dispose of an RQCB
97      0097 1 | DUMP_LOG_FILE, | Format the message into the log file
98      0098 1 | FIND_OPERATOR, | Find a given operator
99      0099 1 | FORMAT_MESSAGE, | Format a given message
100     0100 1 | LOG_MESSAGE, | Write a message to a logfile
101     0101 1 | NOTIFY_OPERATOR, | Notify a single operator
102     0102 1 | NOTIFY_LISTED_OPERATORS, | Notify a list of operators
103     0103 1 | SEND_REPLY, | Send a reply to a request
104     0104 1 | VALID_OPERATOR; | Check operator device
105     0105 1 |
106     0106 1 | EXTERNAL LITERAL
107     0107 1 | MIN_SCOPE, | Minimum scope value
108     0108 1 | MAX_SCOPE; | Maximum scope value

```

```

110 0109 1 GLOBAL ROUTINE REPLY_HANDLER (BUFFER_DESC) : NOVALUE =
111 0110 1
112 0111 1 !++
113 0112 1 Functional description:
114 0113 1
115 0114 1 This routine is the handler for all REPLY messages received by OPCOM.
116 0115 1
117 0116 1
118 0117 1 Input:
119 0118 1
120 0119 1 BUFFER_DESC : The address of a quadword buffer descriptor that
121 0120 1 describes the bu fer containing the message.
122 0121 1
123 0122 1 Implicit Input:
124 0123 1
125 0124 1 None.
126 0125 1
127 0126 1 Output:
128 0127 1
129 0128 1 None.
130 0129 1
131 0130 1 Implicit output:
132 0131 1
133 0132 1 Some accounting data will be updated
134 0133 1 to reflect the receipt of the message.
135 0134 1
136 0135 1 Side effects:
137 0136 1
138 0137 1 None.
139 0138 1
140 0139 1 Routine value:
141 0140 1
142 0141 1 None.
143 0142 1 --
144 0143 1
145 0144 2 BEGIN ! Start of REPLY_HANDLER
146 0145 2
147 0146 2 MAP
148 0147 2 BUFFER_DESC : $ref_bblock;
149 0148 2
150 0149 2 LOCAL
151 0150 2 MESSAGE_VECTOR : VECTOR [5, LONG], ! Message info
152 0151 2 RQST : $ref_bblock, ! Request RQCB
153 0152 2 RQCB : $ref_bblock, ! Work RQCB
154 0153 2 OPER_RQCB : $ref_bblock, ! Operator RQCB
155 0154 2 MCB : $ref_bblock, ! MCB data structure
156 0155 2 OCD : $ref_bblock, ! OCD data structure
157 0156 2 MSG : $ref_bblock,
158 0157 2 MESSAGE : LONG, ! Holds message code
159 0158 2 RQST_COUNT : LONG, ! # of outstanding requests
160 0159 2 FOUND : LONG, ! Boolean
161 0160 2 STATUS : LONG;
162 0161 2
163 0162 2 !
164 0163 2 ! Check request size
165 0164 2
166 0165 3 IF .BUFFER_DESC [DSC$W_LENGTH] LSS (OPC$K_COMHDRSIZ + OPC$K_REPLY_MIN_SIZE)

```

```

167 0166 2 THEN
168 0167 2     RETURN;
169 0168 2
170 0169 2     Do some common sanity checks.
171 0170 2
172 0171 2 IF NOT CHECK_REQUEST (.BUFFER_DESC, RQCB)
173 0172 2 THEN
174 0173 2     RETURN;
175 0174 2
176 0175 2     Make sure this is a valid operator.  If not, ignore the request.
177 0176 2
178 0177 2 IF NOT VALID_OPERATOR (.BUFFER_DESC, .RQCB)
179 0178 2 THEN
180 0179 2     BEGIN
181 0180 2         DEALLOCATE_RQCB (.RQCB);
182 0181 2         RETURN;
183 0182 2     END;
184 0183 2
185 0184 2     Create a descriptor for the reply text, if any.
186 0185 2     MSG is pointer to the beginning of the ASCII text string.
187 0186 2     Note that the count is a word size.
188 0187 2
189 0188 2 MSG = .BUFFER_DESC [DSC$A_POINTER] + OPC$K_COMHDRSIZ;
190 0189 2 MSG = .BUFFER_DESC [DSC$A_POINTER] + (OPC$K_COMHDRSIZ + OPC$K_HDR_SIZE + .$bblock [MSG [OPC$T_OPENABLE_OPR]
191 0190 2 IF .MSG [0,0,16,0] GTR 0
192 0191 2 THEN
193 0192 2     BEGIN
194 0193 2         RQCB [RQCB_L_TEXT_LEN] = .MSG [0,0,16,0];
195 0194 2         IF NOT (STATUS = OPC$GET_VM (RQCB [RQCB_L_TEXT_LEN], RQCB [RQCB_L_TEXT_PTR]))
196 0195 2         THEN
197 0196 2             $signal stop (.STATUS);
198 0197 2             CH$MOVE (.RQCB [RQCB_L_TEXT_LEN], MSG [2,0,0,0], .RQCB [RQCB_L_TEXT_PTR]);
199 0198 2         END;
200 0199 2
201 0200 2     Perform some more checks.  These include:
202 0201 2     - The operator must be known to OPCOM
203 0202 2     - The reply must be valid
204 0203 2
205 0204 2 IF NOT FIND_OPERATOR (.RQCB, OPER RQCB)
206 0205 2 OR NOT (SELECTONE .RQCB [RQCB_L_RQ_OPTIONS] OF
207 0206 2     SET
208 0207 2         [OPC$_RQSTPEND]           : TRUE;           ! Valid reply status
209 0208 2         [OPC$_RQSTCPLTE]        : TRUE;           ! Valid reply status
210 0209 2         [OPC$_RQSTABORT]        : TRUE;           ! Valid reply status
211 0210 2         [OPC$_INITAPE]          : TRUE;           ! Valid reply status
212 0211 2         [OPC$_BLANKTAPE]        : . $bblock [RQCB [RQCB_L_PRIVMASK1], PRV$V_VOLPRO]; ! Valid if user has
213 0212 2         [OTHERWISE]             : FALSE;          ! All others are invalid
214 0213 2     TES)
215 0214 2 THEN
216 0215 2     BEGIN
217 0216 2         MESSAGE_VECTOR [0] = OPC$_ILLRQST;         ! Set message code
218 0217 2         MESSAGE_VECTOR [1] = 0;                   ! Set message NARGS
219 0218 2     END
220 0219 2 ELSE
221 0220 2     BEGIN
222 0221 2
223 0222 2     ! Search through the requests queued to this OCD for the

```

```

: 224 0223 3 : specified request. This implies that an operator may
: 225 0224 3 : only influence requests of the same scope as the operator.
: 226 0225 3
: 227 0226 3
: 228 0227 3 FOUND = FALSE; : Assume not found
: 229 0228 3 MESSAGE_VECTOR [0] = OPCS_NOSUCHRQST; : Set message code
: 230 0229 3 MESSAGE_VECTOR [1] = 0; : Set message NARGS
: 231 0230 3 OCD = .OPER RQCB [RQCB_L_OCD]; : Get OCD address
: 232 0231 3 RQST_COUNT = .OCD [OCD_W-RQSTCOUNT]; : Get # of requests
: 233 0232 3 RQST = .OCD [OCD_L-RQSTFLINK]; : Get first request address
: 234 0233 3 WHILE (NOT .FOUND) AND (.RQST_COUNT GTR 0) DO
: 235 0234 3 IF .RQCB [RQCB_L-RQSTID] NEQ .RQST [RQCB_L-RQSTNUM]
: 236 0235 4 THEN
: 237 0236 4 BEGIN
: 238 0237 4 RQST_COUNT = .RQST_COUNT - 1; : Decrement request count
: 239 0238 4 RQST = .RQST [RQCB_L-FLINK]; : Get address of next request RQCB
: 240 0239 3 END
: 241 0240 4 ELSE
: 242 0241 4 BEGIN
: 243 0242 4 : We've found the request. If the operator has the privilege
: 244 0243 4 : to reply to this request, then format the reply text (if any)
: 245 0244 4 : and send the reply to the requestor.
: 246 0245 4
: 247 0246 4 FOUND = TRUE; : Note that we found it
: 248 0247 5 IF (.Sblock [RQCB [RQCB_L_PRIVMASK1], PRV$V OPER])
: 249 0248 5 OR ((.RQCB [RQCB_B SCOPE] EQL OPCS$K GROUP) AND (.Sblock [RQCB [RQCB_L_PRIVMASK1], PRV$V_GROUP])
: 250 0249 5 OR (.RQCB [RQCB_C_OIC] EQL .RQST [RQCB_L_UIC])
: 251 0250 4 THEN
: 252 0251 5 BEGIN
: 253 0252 5 MESSAGE_VECTOR [0] = .RQCB [RQCB_L_RQ_OPTIONS];
: 254 0253 5 MESSAGE_VECTOR [1] = RQCB [RQCB_C_TEXT_LEN];
: 255 0254 5 MESSAGE_VECTOR [2] = 0;
: 256 0255 5 MESSAGE_VECTOR [3] = .RQCB [RQCB_L_RQSTID];
: 257 0256 5 MESSAGE_VECTOR [4] = RQCB [RQCB_C_OPER_LEN];
: 258 0257 5 FORMAT_MESSAGE (.RQCB, MESSAGE_VECTOR);
: 259 0258 5
: 260 0259 5 : Attempt to notify the person
: 261 0260 5
: 262 0261 5 MCB = .RQST [RQCB_L_MCB]; : Save MCB address
: 263 0262 5 RQST [RQCB_L_MCB] = .RQCB [RQCB_L_MCB]; : Set new MCB
: 264 0263 5 IF NOT SEND_REPLY (.RQST) : Send the reply
: 265 0264 5 THEN
: 266 0265 6 BEGIN
: 267 0266 6 : The requestor could not be notified. This
: 268 0267 6 : means that the requestor has implicitly canceled
: 269 0268 6 : the request.
: 270 0269 6
: 271 0270 6
: 272 0271 6 MESSAGE_VECTOR [0] = OPCS_RQSTCAN; : Set message code
: 273 0272 6 MESSAGE_VECTOR [1] = 0; : Set message Nargs
: 274 0273 6 MESSAGE_VECTOR [2] = .RQCB [RQCB_L_RQSTID];
: 275 0274 6 FORMAT_MESSAGE (.RQCB, MESSAGE_VECTOR);
: 276 0275 6 END
: 277 0276 6
: 278 0277 6 : Able to reply, send it to everyone else in the cluster
: 279 0278 6
: 280 0279 5 ELSE

```

```

281 0280 5 CLUSMSG_RQCB SEND (-1, CLM_REPLY, .RQCB); ! Send it everywhere
282 0281 5 RQST [RQCB_C_MCB] = .MCB; ! Restore the MCB
283 0282 5 END
284 0283 4 ELSE
285 0284 5 BEGIN
286 0285 5
287 0286 5 The operator did not have the privilege to reply to this request.
288 0287 5 Set up the message so that he will be notified that his was an
289 0288 5 illegal request.
290 0289 5
291 0290 5 MESSAGE_VECTOR [0] = OPCS_ILLRQST; ! Set message code
292 0291 5 MESSAGE_VECTOR [1] = 0; ! Set message NARGS
293 0292 4 END;
294 0293 3 END;
295 0294 2
296 0295 1
297 0296 2 Finish processing. Decide what to do based on the message code.
298 0297 2
299 0298 2 SELECTONE .MESSAGE_VECTOR [0] OF
300 0299 2 SET
301 0300 2 [OPCS_ILLRQST,
302 0301 2 OPCS_NOSUCHRQST] : BEGIN
303 0302 2
304 0303 2 Inform the operator that the reply failed.
305 0304 2
306 0305 2 FORMAT_MESSAGE (.RQCB, MESSAGE_VECTOR);
307 0306 2 NOTIFY_OPERATOR (.RQCB);
308 0307 2 END;
309 0308 2
310 0309 2 [OPCS_RQSTPEND] : BEGIN
311 0310 2
312 0311 2 Log the message. Set the interest mask so that
313 0312 2 the message is logged properly.
314 0313 2
315 0314 2 RQCB [RQCB_L_ATTNUMASK1] = .RQST [RQCB_L_ATTNUMASK1];
316 0315 2 RQCB [RQCB_L_ATTNUMASK2] = .RQST [RQCB_L_ATTNUMASK2];
317 0316 2 LOG_MESSAGE (.RQCB);
318 0317 2 END;
319 0318 2
320 0319 2 [OPCS_RQSTCAN,
321 0320 2 OPCS_RQSTABORT,
322 0321 2 OPCS_INITAPE,
323 0322 2 OPCS_BLANKTAPE,
324 0323 2 OPCS_RQSTCMPLTE] : BEGIN
325 0324 2
326 0325 2 The request must be removed from the OCD's list, and the operators must be
327 0326 2 notified. Set the replier's interest mask so that the interested
328 0327 2 operators will be notified.
329 0328 2
330 0329 2 RQCB [RQCB_L_ATTNUMASK1] = .RQST [RQCB_L_ATTNUMASK1];
331 0330 2 RQCB [RQCB_L_ATTNUMASK2] = .RQST [RQCB_L_ATTNUMASK2];
332 0331 2 REMQUE (.RQST, RQST);
333 0332 2 OCD = .RQST [RQCB_L_OCD];
334 0333 2 OCD [OCD_W_RQSTCOUNT] = .OCD [OCD_W_RQSTCOUNT] - 1;
335 0334 2 DEALLOCATE_RQCB (.RQST);
336 0335 2
337 0336 2 ! Log and notify

```



```

: 338      0337 3
: 339      0338 3
: 340      0339 3
: 341      0340 2
: 342      0341 2
: 343      0342 2
: 344      0343 2
: 345      0344 2
: 346      0345 2
: 347      0346 1

```

LOG MESSAGE (.RQCB);
NOTIFY_LISTED_OPERATORS (.RQCB);
END;
TES;
DEALLOCATE_RQCB (.RQCB);
RETURN;
END;

! End of REPLY_HANDLER

```

.TITLE OPCSOPCOMRPLY
.IDENT \V04-000\

.EXTRN CHECK_REQUEST, CLUSMSG_CONV_CLM_RQCB
.EXTRN CLUSMSG_RQCB_SEND
.EXTRN DEALLOCATE_RQCB
.EXTRN DUMP_LOG_FILE, FIND_OPERATOR
.EXTRN FORMAT_MESSAGE, LOG_MESSAGE
.EXTRN NOTIFY_OPERATOR
.EXTRN NOTIFY_LISTED_OPERATORS
.EXTRN SEND_REPLY, VALID_OPERATOR
.EXTRN MIN_SCOPE, MAX_SCOPE
.EXTRN OPCSGET_VM, LIB$STOP

```

.PSECT \$CODE\$,NOWRT,2

			01FC 00000	.ENTRY REPLY_HANDLER, Save R2,R3,R4,R5,R6,R7,R8	: 0109
	58	0000G	CF 9E 00002	MOVAB FORMAT_MESSAGE, R8	
	5E		1C C2 00007	SUBL2 #28, SP	
	53	04	AC D0 0000A	MOVL BUFFER_DESC, R3	: 0165
0046	8F		63 B1 0000E	CMPW (R3), #70	
			01 1E 00013	BGEQU 1\$	
			04 00015	RET	
		4008	8F BB 00016 1\$:	PUSHR #M<R3, SP>	: 0171
0000G	CF		02 FB 0001A	CALLS #2, CHECK_REQUEST	
	01		50 E8 0001F	BLBS R0, 2\$	
			04 00022	RET	
	56		6E D0 00023 2\$:	MOVL RQCB, R6	: 0177
0000G	CF	0048	8F BB 00026	PUSHR #M<R3, R6>	
	05		02 FB 0002A	CALLS #2, VALID_OPERATOR	
			50 E8 0002F	BLBS R0, 3\$	
			56 DD 00032	PUSHL R6	: 0180
			01D7 31 00034	BRW 23\$	
52	04	A3	26 C1 00037 3\$:	ADDL3 #38, 4(R3), MSG	: 0188
		1A	A2 9A 0003C	MOVZBL 26(MSG), R0	: 0189
53		04	A3 C1 00040	ADDL3 4(R3), R0, R3	
		41	A3 9E 00045	MOVAB 65(R3), MSG	
			62 B5 00049	TSTW (MSG)	: 0190
			27 13 0004B	BEQL 5\$	
	53	0084	C6 9E 0004D	MOVAB 132(R6), R3	: 0193
	63		62 3C 00052	MOVZWL (MSG), (R3)	
		0088	C6 9F 00055	PUSHAB 136(R6)	: 0194
			53 DD 00059	PUSHL R3	
0000G	CF		02 FB 0005B	CALLS #2, OPCSGET_VM	
	0A		50 E8 00060	BLBS STATUS, 4\$	

				50	DD	00063		PUSHL	STATUS			0196
				01	FB	00065		CALLS	#1, LIB\$STOP			
						04	0006C	RET				
0088	D6	02	A2	63	28	0006D	4\$:	MOVCL	(R3), 2(MSG), @136(R6)			0197
				04	AE	00074	5\$:	PUSHAB	OPER_RQCB			0204
						56	00077	PUSHL	R6			
						02	00079	CALLS	#2, FIND_OPERATOR			
						50	0007E	BLBC	R0, 6\$			
						58	00081	MOVL	88(R6), R0			0205
						50	00085	CML	R0, #360481			0207
						37	0008C	BEQL	8\$			
						50	0008E	CML	R0, #360489			0208
						2E	00095	BEQL	8\$			
						50	00097	CML	R0, #360476			0209
						25	0009E	BEQL	8\$			
						50	000A0	CML	R0, #360915			0210
						1C	000A7	BEQL	8\$			
						50	000A9	CML	R0, #360931			0211
						05	000B0	BNEQ	6\$			
						05	000B2	BBS	#5, 50(R6), 8\$			
OE						08	000B7	6\$:	MOVL	#360572, MESSAGE_VECTOR		0216
						0C	000BF	CLRL	MESSAGE_VECTOR+4			0217
						00	000C2	7\$:	BRW	17\$		0204
						53	000C5	8\$:	CLRL	FOUND		0226
						08	000C7	MOVL	#360516, MESSAGE_VECTOR			0227
						0C	000CF	CLRL	MESSAGE_VECTOR+4			0228
						50	000D2	MOVL	OPER_RQCB, R0			0229
						54	000D6	MOVL	36(R0), OCD			
						55	000DA	MOVZWL	58(OCD), RQST_COUNT			0230
						52	000DE	MOVL	60(OCD), RQST			0231
						DD	000E2	9\$:	BLBS	FOUND, 7\$		0232
						55	000E5	TSTL	RQST_COUNT			
						D9	000E7	BLEQ	7\$			
						70	000E9	CML	100(R6), 112(RQST)			0233
						07	000EE	BEQL	11\$			
						55	000F0	DECL	RQST_COUNT			0236
						52	000F2	MOVL	(RQST), RQST			0237
						EB	000F5	10\$:	BRB	9\$		0233
						01	000F7	11\$:	MOVL	#1, FOUND		0246
						02	000FA	BBS	#2, 50(R6), 13\$			0247
11						02	000FF	CMPB	83(R6), #2			0248
						04	00103	BNEQ	12\$			
						07	00105	BLBS	49(R6), 13\$			
						68	00109	12\$:	CML	104(R6), 104(RQST)		0249
						5F	0010E	BNEQ	16\$			
						08	00110	13\$:	MOVL	88(R6), MESSAGE_VECTOR		0252
						0C	00115	MOVAB	132(R6), MESSAGE_VECTOR+4			0253
						10	0011B	CLRL	MESSAGE_VECTOR+8			0254
						14	0011E	MOVL	100(R6), MESSAGE_VECTOR+12			0255
						18	00123	MOVAB	124(R6), MESSAGE_VECTOR+16			0256
						08	00128	PUSHAB	MESSAGE_VECTOR			0257
						56	0012B	PUSHL	R6			
						02	0012D	CALLS	#2, FORMAT_MESSAGE			
						68	00130	MOVL	108(RQST), MCB			0261
						57	00134	MOVL	108(R6), 108(RQST)			0262
						6C	00139	PUSHL	RQST			0263
						6C	0013B	CALLS	#1, SEND_REPLY			
						01	00138					
						0000G	CF					

	1A		50	E8	00140	BLBS	R0, 14\$		
08	AE	00058084	8F	DO	00143	MOVL	#360580, MESSAGE_VECTOR	0271	
		OC	AE	D4	0014B	CLRL	MESSAGE_VECTOR+4	0272	
10	AE	64	A6	DO	0014E	MOVL	100(R6), MESSAGE_VECTOR+8	0273	
		08	AE	9F	00153	PUSHAB	MESSAGE_VECTOR	0274	
			56	DD	00156	PUSHL	R6		
	68		02	FB	00158	CALLS	#2, FORMAT_MESSAGE		
			OC	11	0015B	BRB	15\$	0265	
			56	DD	0015D	PUSHL	R6	0280	
			OC	DD	0015F	PUSHL	#12		
	7E		01	CE	00161	MNEGL	#1, -(SP)		
0000G	CF		03	FB	00164	CALLS	#3, CLUSMSG_RQCB_SEND		
6C	A2		57	DO	00169	MOVL	MCB, 108(RQST)	0281	
			86	11	0016D	BRB	10\$	0247	
08	AE	0005807C	8F	DO	0016F	MOVL	#360572, MESSAGE_VECTOR	0290	
		OC	AE	D4	00177	CLRL	MESSAGE_VECTOR+4	0291	
			FF65	31	0017A	BRW	9\$	0233	
00058044	53	08	AE	DO	0017D	MOVL	MESSAGE_VECTOR, R3	0298	
	8F		53	D1	00181	CML	R3, #360516	0300	
0005807C	8F		09	13	00188	BEQL	18\$		
			53	D1	0018A	CML	R3, #360572		
			11	12	00191	BNEQ	19\$		
		08	AE	9F	00193	PUSHAB	MESSAGE_VECTOR	0305	
			56	DD	00196	PUSHL	R6		
	68		02	FB	00198	CALLS	#2, FORMAT_MESSAGE		
			56	DD	0019B	PUSHL	R6	0306	
0000G	CF		01	FB	0019D	CALLS	#1, NOTIFY_OPERATOR		
			68	11	001A2	BRB	22\$	0298	
00058021	8F		53	D1	001A4	CMFL	R3, #360481	0309	
			0E	12	001AB	BNEQ	20\$		
5C	A6	5C	A2	7D	001AD	MOVQ	92(RQST), 92(R6)	0314	
			56	DD	001B2	PUSHL	R6	0316	
0000G	CF		01	FB	001B4	CALLS	#1, LOG_MESSAGE		
			51	11	001B9	BRB	22\$	0298	
0005801C	8F		53	D1	001BB	CML	R3, #360476	0319	
			24	13	001C2	BEQL	21\$		
00058029	8F		53	D1	001C4	CML	R3, #360489		
			1B	13	001CB	BEQL	21\$		
00058084	8F		53	D1	001CD	CML	R3, #360580		
			12	13	001D4	BEQL	21\$		
000581D3	8F		53	D1	001D6	CML	R3, #360915		
			09	13	001DD	BEQL	21\$		
000581E3	8F		53	D1	001DF	CML	R3, #360931		
			24	12	001E6	BNEQ	22\$		
5C	A6	5C	A2	7D	001E8	MOVQ	92(RQST), 92(R6)	0329	
	52		62	0F	001ED	REMQUE	(RQST), RQST	0331	
	54	24	A2	DO	001F0	MOVL	36(RQST), OCD	0332	
		3A	A4	B7	001F4	DECW	58(OCD)	0333	
			52	DD	001F7	PUSHL	RQST	0334	
0000G	CF		01	FB	001F9	CALLS	#1, DEALLOCATE_RQCB		
			6E	DD	001FE	PUSHL	RQCB	0338	
0000G	CF		01	FB	00200	CALLS	#1, LOG_MESSAGE		
			6E	DD	00205	PUSHL	RQCB	0339	
0000G	CF		01	FB	00207	CALLS	#1, NOTIFY_LISTED_OPERATORS		
			6E	DD	0020C	PUSHL	RQCB	0343	
0000G	CF		01	FB	0020E	CALLS	#1, DEALLOCATE_RQCB		
			04	00213	RET			0346	

OPC\$OPCOMRPLY
V04-000

F 12
16-Sep-1984 01:35:42
14-Sep-1984 12:50:50

VAX-11 Bliss-32 V4.0-742
[OPCOM.SRC]OPCOMRPLY.B32;1

Page 10
(2)

OPC
V04

; Routine Size: 532 bytes, Routine Base: \$CODE\$ + 0000

.....

```

349 0347 1 GLOBAL ROUTINE REPLY_CLM_HANDLER (BUFFER_DESC : $ref_bblock, CLM : $ref_bblock, LEN) : NOVALUE =
350 0348 1
351 0349 1 +-
352 0350 1 Functional description:
353 0351 1
354 0352 1 This routine is the handler for all REPLY messages received by OPCOM from remote nodes.
355 0353 1
356 0354 1
357 0355 1 Input:
358 0356 1
359 0357 1 BUFFER_DESC - pointer to message from remote node, including $SENDPR header
360 0358 1 CLM - pointer to CLMRQCB structure
361 0359 1 LEN - length of LEN
362 0360 1
363 0361 1 Implicit Input:
364 0362 1
365 0363 1 None.
366 0364 1
367 0365 1 Output:
368 0366 1
369 0367 1 None.
370 0368 1
371 0369 1 Implicit output:
372 0370 1
373 0371 1 Some accounting data will be updated
374 0372 1 to reflect the receipt of the message.
375 0373 1
376 0374 1 Side effects:
377 0375 1
378 0376 1 None.
379 0377 1
380 0378 1 Routine value:
381 0379 1
382 0380 1 None.
383 0381 1 --
384 0382 1
385 0383 2 BEGIN ! Start of REPLY_CLM_HANDLER
386 0384 2
387 0385 2 LOCAL
388 0386 2 RQST : $ref_bblock, ! Request RQCB
389 0387 2 RQCB : $ref_bblock, ! Work RQCB
390 0388 2 OPER_RQCB : $ref_bblock, ! Operator RQCB
391 0389 2 MCB : $ref_bblock, ! MCB data structure
392 0390 2 OCD : $ref_bblock, ! OCD data structure
393 0391 2 MSG : $ref_bblock,
394 0392 2 MESSAGE : LONG, ! Holds message code
395 0393 2 RQST_COUNT : LONG, ! # of outstanding requests
396 0394 2 STATOS : LONG;
397 0395 2
398 0396 2
399 0397 2 ! Check the version number of the message. If the message is from any other version,
400 0398 2 ! simply ignore it.
401 0399 2
402 0400 2 IF .CLM [CLM_B_DS_VERSION] NEQ CLMRQCB_K_DS_VERSION
403 0401 2 THEN
404 0402 2 RETURN DUMP_LOG_FILE (.BUFFER_DESC, %ASCID 'CLM_x mismatch');
405 0403 2 !

```

```
406 0404 2 : Allocate an RQCB and convert the message RQCB into the new RQCB
407 0405 2
408 0406 2 IF NOT CLUSMSG_CONV_CLM_RQCB (.CLM, RQCB)
409 0407 2 THEN
410 0408 2     RETURN DUMP_LOG_FILE (.BUFFER_DESC, ascid_INVALIDRQCB);
411 0409 2
412 0410 2 : Perform some more checks. These include:
413 0411 2 - The operator must be known to OPCOM
414 0412 2 - The reply must be valid
415 0413 2
416 0414 2 IF NOT FIND_OPERATOR (.RQCB, OPER_RQCB)
417 0415 2 THEN
418 0416 2     BEGIN
419 0417 2     DUMP LOG FILE (.BUFFER_DESC, %ASCID 'reply operator not found');
420 0418 2     DEALLOCATE_RQCB (.RQCB);
421 0419 2     RETURN;
422 0420 2     END;
423 0421 2
424 0422 2 : Search through the requests queued to this OCD for the
425 0423 2 specified request. This implies that an operator may
426 0424 2 only influence requests of the same scope as the operator.
427 0425 2
428 0426 2 OCD = .OPER_RQCB [RQCB_L_OCD];           ! Get OCD address
429 0427 2 RQST = .OCD [OCD_L_RQSTFLINK];         ! Get first request address
430 0428 2 WHILE .RQST NEQ OCD [OCD_L_RQSTFLINK] DO
431 0429 2     BEGIN
432 0430 2     IF .RQCB [RQCB_L_RQSTID] NEQ .RQST [RQCB_L_RQSTNUM]
433 0431 2     THEN
434 0432 2         RQST = .RQST [RQCB_L_FLINK]           ! Get address of next request RQCB
435 0433 2     ELSE
436 0434 2         BEGIN
437 0435 2         :
438 0436 2         : We've found the request. If the operator has the privilege
439 0437 2         : to reply to this request, then format the reply text (if any)
440 0438 2         : and send the reply to the requestor.
441 0439 2         :
442 0440 2         MCB = .RQST [RQCB_L_MCB];           ! Save MCB address
443 0441 2         RQST [RQCB_L_MCB] = .RQCB [RQCB_L_MCB]; ! Set new MCB
444 0442 2         SEND_REPLY (.RQST);                 ! Send the reply
445 0443 2         RQST [RQCB_L_MCB] = .MCB;           ! Restore the MCB
446 0444 2         EXITLOOP;
447 0445 2         END;
448 0446 2     END;
449 0447 2
450 0448 2 : Request not found, oh well
451 0449 2
452 0450 2 IF .RQST EQL OCD [OCD_L_RQSTFLINK]
453 0451 2 THEN
454 0452 2     BEGIN
455 0453 2     DUMP LOG FILE (.BUFFER_DESC, %ASCID 'reply request not found');
456 0454 2     DEALLOCATE_RQCB (.RQCB);
457 0455 2     RETURN;
458 0456 2     END;
459 0457 2
460 0458 2 : Finish processing. Decide what to do based on the message code.
461 0459 2
462 0460 2 IF (MCB = .RQCB [RQCB_L_MCB]) NEQ 0
```

```

: 463 0461 2 THEN
: 464 0462 3 BEGIN
: 465 0463 3 SELECTONE .MCB [MCB_L_MSGID] OF
: 466 0464 3 SET
: 467 0465 4 [OPCS_RQSTPEND] : BEGIN
: 468 0466 4
: 469 0467 4 Log the message. Set the interest mask so that
: 470 0468 4 the message is logged properly.
: 471 0469 4
: 472 0470 4 RQCB [RQCB_L_ATTNUMASK1] = .RQST [RQCB_L_ATTNUMASK1];
: 473 0471 4 RQCB [RQCB_L_ATTNUMASK2] = .RQST [RQCB_L_ATTNUMASK2];
: 474 0472 4 LOG MESSAGE (.RQCB);
: 475 0473 3 END;
: 476 0474 3
: 477 0475 3 [OPCS_RQSTCAN,
: 478 0476 3 OPCS_RQSTABORT,
: 479 0477 3 OPCS_INITAPE,
: 480 0478 3 OPCS_BLANKTAPE,
: 481 0479 4 OPCS_RQSTCPLTE] : BEGIN
: 482 0480 4
: 483 0481 4 The request must be removed from the OCD's list,
: 484 0482 4 and the operators must be notified. Set the
: 485 0483 4 replier's interest mask so that the interested
: 486 0484 4 operators will be notified.
: 487 0485 4
: 488 0486 4 RQCB [RQCB_L_ATTNUMASK1] = .RQST [RQCB_L_ATTNUMASK1];
: 489 0487 4 RQCB [RQCB_L_ATTNUMASK2] = .RQST [RQCB_L_ATTNUMASK2];
: 490 0488 4 REMQUE (.RQST, RQST);
: 491 0489 4 OCD = .RQST [RQCB_L_OCD];
: 492 0490 4 OCD [OCD_W_RQSTCOUNT] = .OCD [OCD_W_RQSTCOUNT] - 1;
: 493 0491 4 DEALLOCATE_RQCB (.RQST);
: 494 0492 4 LOG MESSAGE (.RQCB);
: 495 0493 4 NOTIFY_LISTED_OPERATORS (.RQCB);
: 496 0494 3 END;
: 497 0495 3 [OTHERWISE] : DUMP_LOG_FILE (.BUFFER_DESC, %ASCID 'unexpected status for reply');
: 498 0496 3 TES;
: 499 0497 3 END
: 500 0498 3 !
: 501 0499 3 ! What, no MCB on the message! Bogus
: 502 0500 3 !
: 503 0501 2 ELSE
: 504 0502 2 DUMP_LOG_FILE (.BUFFER_DESC, %ASCID 'no MCB on reply message');
: 505 0503 2
: 506 0504 2 DEALLOCATE_RQCB (.RQCB);
: 507 0505 2
: 508 0506 2 RETURN;
: 509 0507 1 END;

```

! End of REPLY_CLM_HANDLER

.PSECT \$SPLITS,NOWRT,NOEXE,2

```

68 63 74 61 6D 73 69 6D 20 78 5F 5F 4D 4C 43 0000 P.AAB: .ASCII \CLM_x mismatch\<>
: 00 000F
: 010E000F 00010 P.AAA: .LONG 17694735
: 00000000 00014 .ADDRESS P.AAB
20 72 6F 74 61 72 65 70 6F 20 79 6C 70 65 72 00018 P.AAD: .ASCII \reply operator not found\

```

```

        64  6E  75  6F  66  20  74  6F  6E  00027
        010E0018  00030 P.AAC:  .LONG  17694744
        00000000'  00034 .ADDRESS P.AAD
6E  20  74  73  65  75  71  65  72  20  79  6C  70  65  72  00038 P.AAF:  .ASCII  \reply request not found\<0>
        00  64  6E  75  6F  66  20  74  6F  00047
        010E0017  00050 P.AAE:  .LONG  17694743
        00000000'  00054 .ADDRESS P.AAF
74  61  74  73  20  64  65  74  63  65  70  78  65  6E  75  00058 P.AAH:  .ASCII  \unexpected status for reply\<0>
        00  79  6C  70  65  72  20  72  6F  66  20  73  75  00067
        010E001B  00074 P.AAG:  .LONG  17694747
        C0000000'  00078 .ADDRESS P.AAH
79  6C  70  65  72  20  6E  6F  20  42  43  4D  20  6F  6E  0007C P.AAJ:  .ASCII  \no MCB on reply message\<0>
        00  65  67  61  73  73  65  6D  20  0008B
        C10E0017  00094 P.AAI:  .LONG  17694743
        00000000'  00098 .ADDRESS P.AAJ
    
```

```

.PSECT  $CODE$,NOWRT,2
        .EXTRN  ASCID_INVALIDRQCB
        .ENTRY  REPLY_CLM_HANDLER, Save R2,R3,R4,R5,R6
    
```

```

        56      0000G  CF  007C  00000 .ENTRY  REPLY_CLM_HANDLER, Save R2,R3,R4,R5,R6 : 0347
        5E      0000G  CF  9E  00002 MOVAB   DUMP_LOG_FILE, R6
        52      08    AC  D0  0000A 00000G  CF  08    AC  D0  0000A   MOVL    CLM, R2   : 0400
        02      02    A2  91  0000E   CMPB   2(R2), #2
        0000'  06    13  00012   BEQL   1$
        4004    0000'  CF  9F  00014   PUSHAB P.AAA   : 0402
        10    11  00018   BRB    2$
        0000G  CF  8F  BB  0001A 1$:   PUSHR  #*M<R2, SP> : 0406
        08    02    FB  0001E   CALLS  #2, CLUSMSG_CONV_CLM_RQCB
        50    E8  00023   BLBS   R0, 3$
        0000G  CF  9F  00026   PUSHAB ASCID_INVALIDRQCB : 0408
        04    AC  DD  0002A 2$:   PUSHL  BUFFER_DESC
        66      02    FB  0002D   CALLS  #2, DUMP_LOG_FILE
        04    AE  9F  00031 3$:   RET
        53      04    AE  D0  00034   PUSHAB OPER_RQCB : 0414
        53    DD  00038   MOVL   RQCB, R3
        0000G  CF  02    FB  0003A   PUSHL  R3
        06      50    E8  0003F   CALLS  #2, FIND_OPERATOR
        0000'  CF  9F  00042   BLBS   R0, 4$
        42    11  00046   PUSHAB P.AAC : 0417
        50      04    AE  D0  00048 4$:   BRB    8$
        54      24    A0  D0  0004C   MOVL   OPER_RQCB, R0 : 0426
        52      3C    A4  D0  00050   MOVL   36(R0), OCD
        50      3C    A4  9E  00054 5$:   MOVL   60(OCD), RQST : 0427
        50      52    D1  00058   MOVAB  60(OCD), R0 : 0428
        20      20    13  0005B   CMPL  RQST, R0
        70      A2      64    A3  D1  0005D   BEQL   7$
        05      13  00062   CMPL  100(R3), 112(RQST) : 0430
        52      62    D0  00064   BEQL   6$
        EB    11  00067   MOVL  (RQST), RQST : 0432
        55      6C    A2  D0  00069 6$:   BRB    5$
        6C      A2      6C    A3  D0  0006D   MOVL   108(RQST), MCB : 0440
        52      DD  00072   MOVL  108(R3), 108(RQST) : 0441
        0000G  CF  01    FB  00074   PUSHL  RQST : 0442
        6C      A2      55    D0  00079   CALLS  #1, SEND_REPLY
        6C      A2      55    D0  00079   MOVL  MCB, 108(RQST) : 0443
    
```


50	3C	A4	9E	0007D	7\$:	MOVAB	60(OCD), R0	0450
50		52	D1	00081		CMPL	RQST, R0	
		0F	12	00084		BNEQ	9\$	
	0000'	CF	9F	00086		PUSHAB	P.AAE	0453
	04	AC	DD	0008A	8\$:	PUSHL	BUFFER_DESC	
66		02	FB	0008D		CALLS	#2, DUMP_LOG_FILE	
		53	DD	00090		PUSHL	R3	0454
		0086	31	00092		BRW	16\$	
55	6C	A3	D0	00095	9\$:	MOVL	108(R3), MCB	0460
		74	13	00099		BEQL	13\$	
00058021	50	2C	A5	D0	0009B	MOVL	44(MCB), R0	0463
	8F		50	D1	0009F	CMPL	R0, #360481	0465
	5C	A3	5C	0E	12	BNEQ	10\$	
				A2	7D	MOVQ	92(RQST), 92(R3)	0470
	0000G	CF		53	DD	PUSHL	R3	0472
				01	FB	CALLS	#1, LOG_MESSAGE	
0005801C	8F			63	11	BRB	15\$	0463
				50	D1	CMPL	R0, #360476	0475
00058029	8F			24	13	BEQL	11\$	
				50	D1	CMPL	R0, #360489	
00058084	8F			1B	13	BEQL	11\$	
				50	D1	CMPL	R0, #360580	
000581D3	8F			12	13	BEQL	11\$	
				50	D1	CMPL	R0, #360915	
000581E3	8F			09	13	BEQL	11\$	
				50	D1	CMPL	R0, #360931	
	5C	A3	5C	26	12	BNEQ	12\$	
		52		A2	7D	MOVQ	92(RQST), 92(R3)	0486
		54		62	0F	REMQUE	(RQST), RQST	0488
				24	A2	MOVL	36(RQST), OCD	0489
				3A	A4	DECW	58(OCD)	0490
				52	DD	PUSHL	RQST	0491
0000G	CF			01	FB	CALLS	#1, DEALLOCATE_RQCB	
				6E	DD	PUSHL	RQCB	0492
0000G	CF			01	FB	CALLS	#1, LOG_MESSAGE	
				6E	DD	PUSHL	RQCB	0493
0000G	CF			01	FB	CALLS	#1, NOTIFY_LISTED_OPERATORS	
				10	11	BRB	15\$	0463
		0000'		CF	9F	PUSHAB	P.AAG	0495
				04	11	BRB	14\$	
		0000'		CF	9F	PUSHAB	P.AAI	0502
		04		AC	DD	PUSHL	BUFFER_DESC	
	66			02	FB	CALLS	#2, DUMP_LOG_FILE	
				6E	DD	PUSHL	RQCB	0504
0000G	CF			01	FB	CALLS	#1, DEALLOCATE_RQCB	
				04	00120	RET		0507

; Routine Size: 289 bytes, Routine Base: \$CODE\$ + 0214

: 511 0508 1 END
: 512 0509 0 ELUDOM

! End of OPCOMRPLY

PSECT SUMMARY

Name	Bytes	Attributes
\$CODES	821	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$SPLITS	156	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Symbols -----		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	14 0	1000	00:01.9
_\$255\$DUA28:[OPCOM.OBJ]OPCOMLIB.L32;1	633	31 4	43	00:01.0

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:OPCOMRPLY/OBJ=OBJ\$:OPCOMRPLY MSRC\$:OPCOMRPLY/UPDATE=(ENHS:OPCOMRPLY)

: Size: 821 code + 156 data bytes
: Run Time: 00:17.5
: Elapsed Time: 00:46.5
: Lines/CPU Min: 1741
: Lexemes/CPU-Min: 15075
: Memory Used: 168 pages
: Compilation Complete

0290 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

A grid of 180 terminal window screenshots, arranged in 15 rows and 12 columns. Each window displays technical data, including system logs, configuration files, and diagnostic reports. Several windows are highlighted with larger, bolded text labels:

- LOGFILE LIS
- OPCOMDEF LIS
- OPCOMDATA LIS
- OPCOMINI LIS
- OPCOMLIB LIS
- OPCOMDEFTMP LIS
- OPCCRASH LIS
- OPCOMUTIL LIS
- OPCOMMAIN LIS
- OPCOMOLD LIS
- OPCOMRPLY LIS
- OPCOMRST LIS
- OPERUTIL LIS

The screenshots show various system parameters, error messages, and operational status indicators. The text is small and dense, typical of early computer terminal displays.