

Sym
--
ALL

ASC

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

000000000	PPPPPPPPPPP	CCCCCCCCCCCCC	000000000	MMM	MMM
000000000	PPPPPPPPPPP	CCCCCCCCCCCCC	000000000	MMM	MMM
000000000	PPPPPPPPPPP	CCCCCCCCCCCCC	000000000	MMM	MMM
000 000	PPP PPP	CCC	000 000	MMMMMM	MMMMMM
000 000	PPP PPP	CCC	000 000	MMMMMM	MMMMMM
000 000	PPP PPP	CCC	000 000	MMMMMM	MMMMMM
000 000	PPP PPP	CCC	000 000	MMM	MMM
000 000	PPP PPP	CCC	000 000	MMM	MMM
000 000	PPP PPP	CCC	000 000	MMM	MMM
000 000	PPP PPP	CCC	000 000	MMM	MMM
000 000	PPPPPPPPPPP	CCC	000 000	MMM	MMM
000 000	PPPPPPPPPPP	CCC	000 000	MMM	MMM
000 000	PPPPPPPPPPP	CCC	000 000	MMM	MMM
000 000	PPP	CCC	000 000	MMM	MMM
000 000	PPP	CCC	000 000	MMM	MMM
000 000	PPP	CCC	000 000	MMM	MMM
000 000	PPP	CCC	000 000	MMM	MMM
000 000	PPP	CCC	000 000	MMM	MMM
000 000	PPP	CCC	000 000	MMM	MMM
000 000	PPP	CCC	000 000	MMM	MMM
000 000	PPP	CCC	000 000	MMM	MMM
000000000	PPP	CCCCCCCCCCCCC	000000000	MMM	MMM
000000000	PPP	CCCCCCCCCCCCC	000000000	MMM	MMM
000000000	PPP	CCCCCCCCCCCCC	000000000	MMM	MMM

```

000000  PPPPPPPP  CCCCCCCC  000000  MM      MM  000000  LL      DDDDDDDD
000000  PPPPPPPP  CCCCCCCC  000000  MM      MM  000000  LL      DDDDDDDD
00      00  PP      PP  CC      00      00  MMMM  MMMM  00      00  LL      DD      DD
00      00  PP      PP  CC      00      00  MMMM  MMMM  00      00  LL      DD      DD
00      00  PP      PP  CC      00      00  MM      MM  00      00  LL      DD      DD
00      00  PP      PP  CC      00      00  MM      MM  00      00  LL      DD      DD
00      00  PPPPPPPP  CC      00      00  MM      MM  00      00  LL      DD      DD
00      00  PPPPPPPP  CC      00      00  MM      MM  00      00  LL      DD      DD
00      00  PP      CC      00      00  MM      MM  00      00  LL      DD      DD
00      00  PP      CC      00      00  MM      MM  00      00  LL      DD      DD
00      00  PP      CC      00      00  MM      MM  00      00  LL      DD      DD
00      00  PP      CC      00      00  MM      MM  00      00  LL      DD      DD
00      00  PP      CC      00      00  MM      MM  00      00  LL      DD      DD
000000  PP      CCCCCCCC  000000  MM      MM  000000  LLLLLLLLLL  DDDDDDDD
000000  PP      CCCCCCCC  000000  MM      MM  000000  LLLLLLLLLL  DDDDDDDD

```

```

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 0001
2 0002
3 0003
4 0004
5 0005
6 0006
7 0007
8 0008
9 0009
10 0010
11 0011
12 0012
13 0013
14 0014
15 0015
16 0016
17 0017
18 0018
19 0019
20 0020
21 0021
22 0022
23 0023
24 0024
25 0025
26 0026
27 0027
28 0028
29 0029
30 0030
31 0031
32 0032
33 0033
34 0034
35 0035
36 0036
37 0037
38 0038
39 0039
40 0040
41 0041
42 0042
43 0043
44 0044
45 0045
46 0046
47 0047
48 0048
49 0049
50 0050
51 0051
52 0052
53 0053
54 0054
55 0055
56 0056
57 0057

```
0 MODULE OPCSOPCOMOLD (
0 LANGUAGE (BLISS32),
0 IDENT = 'V04-000'
0 ) =
0
0 *****
0 *
0 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0 * ALL RIGHTS RESERVED.
0 *
0 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0 * TRANSFERRED.
0 *
0 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0 * CORPORATION.
0 *
0 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0 *
0 *****
0
0 **
0 FACILITY:
0
0 OPCOM
0
0 ABSTRACT:
0
0 This module contains the old format message handlers.
0 These routines merely reformat the request into the
0 new format, and call the correct handler to service
0 the request.
0
0 Environment:
0
0 VAX/VMS operating system.
0
0 Author:
0
0 Steven T. Jeffreys
0
0 Creation date:
0
0 March 10, 1981
0
0 Revision history:
0
0 V03-002 CWH3001 CW Hobbs 30-Jul-1983
0 Various and sundry things to make OPCOM distributed
0 across the cluster.
```

```
58 0058 0 :  
59 0059 0 : V03-001 STJ3033 Steven T. Jeffreys, 05-Oct-1982  
60 0060 0 : Make all operators permanent by default.  
61 0061 0 :  
62 0062 0 : V02-002 STJ0223 Steven T. Jeffreys, 17-Feb-1982  
63 0063 0 : Make all operators temporary by default.  
64 0064 0 :  
65 0065 0 :  
66 0066 0 : --  
67 0067 0 :  
68 0068 1 BEGIN ! Start of OPCOMOLD  
69 0069 1 :  
70 0070 1 LIBRARY 'SYSS$LIBRARY:LIB.L32';  
71 0071 1 LIBRARY 'LIB$:OPCOMLIB';  
72 0072 1 :  
73 0073 1 FORWARD ROUTINE  
74 0074 1 :  
75 0075 1 : Various message handlers for old format messages.  
76 0076 1 :  
77 0077 1 CNCL_HANDLER : NOVALUE, ! Cancel handler  
78 0078 1 LOGI_HANDLER : NOVALUE, ! Init logfile message handler  
79 0079 1 RPLY_HANDLER : NOVALUE, ! Reply handler  
80 0080 1 RQST_HANDLER : NOVALUE, ! Request handler  
81 0081 1 SECU_HANDLER : NOVALUE, ! Security handler  
82 0082 1 STS_HANDLER : NOVALUE, ! Status handler  
83 0083 1 TERME_HANDLER : NOVALUE, ! Enable operator message handler  
84 0084 1 :
```

```

86 0085 1 GLOBAL ROUTINE CNCL_HANDLER (BUFFER_DESC) : NOVALUE =
87 0086 1
88 0087 1 |++
89 0088 1 | Functional description:
90 0089 1 |
91 0090 1 |     This routine is the handler for all CNCL messages received by OPCOM.
92 0091 1 |     This message is in the old format, and must be converted to the new
93 0092 1 |     format before it can be processed. Once this is done, the new format
94 0093 1 |     message handler is called to process the reformatted request.
95 0094 1 |
96 0095 1 | Input:
97 0096 1 |
98 0097 1 |     BUFFER_DESC : The address of a quadword buffer descriptor that
99 0098 1 |     describes the buffer containing the message.
100 0099 1 |
101 0100 1 | Implicit Input:
102 0101 1 |
103 0102 1 |     None.
104 0103 1 |
105 0104 1 | Output:
106 0105 1 |
107 0106 1 |     None.
108 0107 1 |
109 0108 1 | Implicit output:
110 0109 1 |
111 0110 1 |     Some accounting data will be updated
112 0111 1 |     to reflect the receipt of the message.
113 0112 1 |
114 0113 1 | Side effects:
115 0114 1 |
116 0115 1 |     None.
117 0116 1 |
118 0117 1 | Routine value:
119 0118 1 |
120 0119 1 |     None.
121 0120 1 | --
122 0121 1 |
123 0122 2 BEGIN                               ! Start of CNCL_HANDLER
124 0123 2
125 0124 2 MAP
126 0125 2
127 0126 2     BUFFER_DESC      : $ref_bblock;
128 0127 2
129 0128 2 EXTERNAL
130 0129 2     GLOBAL_STATUS      : BITVECTOR;           ! Global status flags
131 0130 2
132 0131 2 EXTERNAL ROUTINE
133 0132 2     CANCEL_HANDLER      : NOVALUE;           ! New format msg handler
134 0133 2
135 0134 2 LOCAL
136 0135 2
137 0136 2     OLD_MSG              : $ref_bblock,       ! Pointer to start of old message
138 0137 2     NEW_MSG              : $ref_bblock,       ! Pointer to start of new message
139 0138 2     REFORMAT_BUFFER      : $bblock [OP($K_MAXREAD), ! Buffer to hold the reformatted message
140 0139 2     REFORMAT_DESC        : $desc_block;       ! Descriptor for the REFORMAT_BUFFER
141 0140 2
142 0141 2 !

```

```

143 0142 2  ! Make sure the message is big enough.  If not, it
144 0143 2  ! cannot possibly be a valid message, and OPCOM will
145 0144 2  ! simply ignore it.
146 0145 2  !
147 0146 2  IF .BUFFER_DESC [DSC$W_LENGTH] LSS (OPC$K_COMHDRSIZ + 8)
148 0147 2  THEN
149 0148 2  RETURN;
150 0149 2  !
151 0150 2  ! Copy the common header provided by $SNDOPR to the new buffer
152 0151 2  !
153 0152 2  CH$MOVE (OPC$K_COMHDRSIZ, .BUFFER_DESC [DSC$A_POINTER] , REFORMAT_BUFFER);
154 0153 2  !
155 0154 2  ! Zero the remainder of the REFORMAT_BUFFER.
156 0155 2  !
157 0156 2  !
158 0157 2  CH$FILL (0, (OPC$K_MAXREAD - OPC$K_COMHDRSIZ), (REFORMAT_BUFFER + OPC$K_COMHDRSIZ));
159 0158 2  !
160 0159 2  !
161 0160 2  ! Move the old message fields into their corresponding places in the new message format.
162 0161 2  !
163 0162 2  !
164 0163 2  OLD_MSG = .BUFFER_DESC [DSC$A_POINTER] + OPC$K_COMHDRSIZ;
165 0164 2  NEW_MSG = REFORMAT_BUFFER + OPC$K_COMHDRSIZ;
166 0165 2  NEW_MSG [OPC$B_RQSTCODE] = .OLD_MSG [OPC$B_MS_TYPE];
167 0166 2  NEW_MSG [OPC$B_SCOPE] = OPC$K_SYSTEM;
168 0167 2  NEW_MSG [OPC$L_ATTNUMASK1] = .OLD_MSG [$BYTEOFFSET (OPC$B_MS_TARGET),0,24,0];
169 0168 2  NEW_MSG [OPC$L_RQSTID] = .OLD_MSG [OPC$L_MS_RQSTID];
170 0169 2  !
171 0170 2  !
172 0171 2  ! Create a descriptor for the reformatted message.
173 0172 2  !
174 0173 2  REFORMAT_DESC [DSC$W_LENGTH] = OPC$K_COMHDRSIZ + OPC$K_HDR_SIZE;
175 0174 2  REFORMAT_DESC [DSC$B_DTYPE] = 0;
176 0175 2  REFORMAT_DESC [DSC$B_CLASS] = 0;
177 0176 2  REFORMAT_DESC [DSC$A_POINTER] = REFORMAT_BUFFER;
178 0177 2  !
179 0178 2  !
180 0179 2  ! Call the new-message handler to finish processing the message.
181 0180 2  !
182 0181 2  GLOBAL_STATUS [GBLSTS_K_OLD_FORMAT_MSG] = TRUE;          ! Mark this as an old format msg
183 0182 2  CANCEL_HANDLER (REFORMAT_DESC);
184 0183 2  GLOBAL_STATUS [GBLSTS_K_OLD_FORMAT_MSG] = FALSE;
185 0184 2  !
186 0185 1  END;

```

! End of CNCL_HANDLER

```

.TITLE OPCSOPCOMOLD
.IDENT \V04-000\

.EXTRN GLOBAL_STATUS, CANCEL_HANDLER

.PSECT $CODE$,NOWRT,2

.ENTRY CNCL_HANDLER, Save R2,R3,R4,R5,R6
MOVAB -2568(SP), SP
MOVL  BUFFER_DESC, R6
CMPW  (R6), #46

```

```

007C 00000
SE    F5F8 CE 9E 00002
56    04  AC D0 00007
2E    66  B1 0000B

```

: 0085
: 0146
:

09DA	8F	08	AE	04	BC	45	1F	0000E	BLSSU	1\$:	0153
			00		6E	26	28	00010	MOVCS	#38, @4(R6), REFORMAT_BUFFER	:	0158
						00	2C	00016	MOVCS	#C, (SP), #0, #2522, REFORMAT_BUFFER+38	:	
			51	04	A6	2E	AE	0001D			:	
					50	2E	AE	9E 00024	ADDL3	#38, 4(R6), OLD_MSG	:	0163
					60		61	90 00028	MOVAB	REFORMAT_BUFFER+38, NEW_MSG	:	0164
					6E		01	90 0002B	MOVAB	(OLD_MSG), (NEW_MSG)	:	0165
0A	A0	01	A1	01	A0		00	EF 0002F	MOVAB	#1, T(NEW_MSG)	:	0166
					18		00	EF 0002F	EXTZV	#0, #24, T(OLD_MSG), 10(NEW_MSG)	:	0167
					12	04	A1	D0 00036	MOVL	4(OLD_MSG), 18(NEW_MSG)	:	0168
					6E	40	8F	9A 0003B	MOVZBL	#64, REFORMAT_DESC	:	0173
					04	08	AE	9E 0003F	MOVAB	REFORMAT_BUFFER, REFORMAT_DESC+4	:	0176
				0000G	CF		10	88 00044	BISB2	#16, GLOBAL_STATUS	:	0181
							5E	DD 00049	PUSHL	SP	:	0182
				0000G	CF		01	FB 0004B	CALLS	#1, CANCEL_HANDLER	:	
				0000G	CF		10	8A 00050	BICB2	#16, GLOBAL_STATUS	:	0183
							04	00055 1\$:	RET		:	0185

; Routine Size: 86 bytes, Routine Base: \$CODE\$ + 0000

```

188 0186 1 GLOBAL ROUTINE LOGI_HANDLER (BUFFER_DESC) : NOVALUE =
189 0187 1
190 0188 1 ++
191 0189 1 Functional description:
192 0190 1
193 0191 1 This routine is the handler for all LOGI messages received by OPCOM.
194 0192 1 This message is in the old format, and must be converted to the new
195 0193 1 format before it can be processed. Once this is done, the new format
196 0194 1 message handler is called to process the reformatted request.
197 0195 1
198 0196 1 Input:
199 0197 1
200 0198 1 BUFFER_DESC : The address of a quadword buffer descriptor that
201 0199 1 describes the buffer containing the message.
202 0200 1
203 0201 1 Implicit Input:
204 0202 1
205 0203 1 None.
206 0204 1
207 0205 1 Output:
208 0206 1
209 0207 1 None.
210 0208 1
211 0209 1 Implicit output:
212 0210 1
213 0211 1 Some accounting data will be updated
214 0212 1 to reflect the receipt of the message.
215 0213 1
216 0214 1 Side effects:
217 0215 1
218 0216 1 None.
219 0217 1
220 0218 1 Routine value:
221 0219 1
222 0220 1 None.
223 0221 1 --
224 0222 1
225 0223 2 BEGIN ! Start of LOGI_HANDLER
226 0224 2
227 0225 2 MAP
228 0226 2
229 0227 2 BUFFER_DESC : $ref_bblock;
230 0228 2
231 0229 2 EXTERNAL
232 0230 2 GLOBAL_STATUS : BITVECTOR, ! Global status flags
233 0231 2 DEVICE_FAO : $bblock; ! FAO control string
234 0232 2
235 0233 2 EXTERNAL ROUTINE
236 0234 2 LOGFILE_HANDLER : NOVALUE; ! New format msg handler
237 0235 2
238 0236 2 LOCAL
239 0237 2
240 0238 2 OLD_MSG : $ref_bblock, ! Pointer to start of old message
241 0239 2 NEW_MSG : $ref_bblock, ! Pointer to start of new message
242 0240 2 OUT_LENGTH : WORD, ! Length of formatted operator device name
243 0241 2 DEV_DESC : $desc_block, ! Operator device name descriptor
244 0242 2 REFORMAT_BUFFER : $bblock [OP($K_MAXREAD), ! Buffer to hold the reformatted message

```



```
245 0243 2 REFORMAT_DESC : $desc_block; ! Descriptor for the REFORMAT_BUFFER
246 0244 2
247 0245 2
248 0246 2 : Make sure the message is big enough. If not, it
249 0247 2 : cannot possibly be a valid message, and OPCOM will
250 0248 2 : simply ignore it.
251 0249 2
252 0250 2 IF .BUFFER_DESC [DSC$W_LENGTH] LSS (OPC$K_COMHDRSIZ + 13)
253 0251 2 THEN
254 0252 2 RETURN;
255 0253 2
256 0254 2 : Copy the common header provided by $SENDPR to the new buffer
257 0255 2
258 0256 2 CH$MOVE (OPC$K_COMHDRSIZ, .BUFFER_DESC [DSC$A_POINTER], REFORMAT_BUFFER);
259 0257 2
260 0258 2
261 0259 2 : Zero the remainder of the REFORMAT_BUFFER.
262 0260 2
263 0261 2
264 0262 2 CH$FILL (0, (OPC$K_MAXREAD - OPC$K_COMHDRSIZ), (REFORMAT_BUFFER + OPC$K_COMHDRSIZ));
265 0263 2
266 0264 2
267 0265 2 : Move the old message fields into their corresponding places in the new message format.
268 0266 2
269 0267 2 OLD_MSG = .BUFFER_DESC [DSC$A_POINTER] + OPC$K_COMHDRSIZ;
270 0268 2 NEW_MSG = REFORMAT_BUFFER + OPC$K_COMHDRSIZ;
271 0269 2 NEW_MSG [OPC$B_RQSTCODE] = .OLD_MSG [OPC$B_MS_TYPE]; ! Set request type
272 0270 2 NEW_MSG [OPC$B_SCOPE] = OPC$K_SYSTEM; ! Force SYSTEM request
273 0271 2 IF .OLD_MSG [OPC$L_MS_RQSTID] EQL 0
274 0272 2 THEN
275 0273 2 $bblock [NEW_MSG [OPC$L_RQ_OPTIONS], OPC$V_INITLOG] = TRUE ! INITLOG function
276 0274 2 ELSE
277 0275 2 $bblock [NEW_MSG [OPC$L_RQ_OPTIONS], OPC$V_CLOSELOG] = TRUE; ! CLOSELOG function
278 0276 2
279 0277 2 : Build the operator device name from the ASCII device string
280 0278 2 : and the device unit number. Build the FAO OUTBUF descriptor
281 0279 2 : to point to the correct spot within NEW_MSG to save a copy.
282 0280 2
283 0281 2 DEV_DESC [0,0,32,0] = 20; ! Allow for a large device name
284 0282 2 DEV_DESC [DSC$A_POINTER] = .NEW_MSG + $BYTEOFFSET (OPC$T_LOGFILE_OPR) + 1;
285 0283 2 $FAO (DEVICE FAO, OUT_LENGTH, DEV_DESC, OLD_MSG [OPC$T_MS_ONAME], .OLD_MSG [OPC$W_MS_OUNIT]);
286 0284 2 NEW_MSG [$BYTEOFFSET (OPC$T_LOGFILE_OPR),0,8,0] = .OUT_LENGTH;
287 0285 2
288 0286 2
289 0287 2 : Create a descriptor for the reformatted message.
290 0288 2
291 0289 2 REFORMAT_DESC [DSC$W_LENGTH] = OPC$K_COMHDRSIZ + OPC$K_HDR_SIZE + .OUT_LENGTH + 1;
292 0290 2 REFORMAT_DESC [DSC$B_DTYPE] = 0;
293 0291 2 REFORMAT_DESC [DSC$B_CLASS] = 0;
294 0292 2 REFORMAT_DESC [DSC$A_POINTER] = REFORMAT_BUFFER;
295 0293 2
296 0294 2
297 0295 2 : Call the new-message handler to finish processing the message.
298 0296 2
299 0297 2 GLOBAL STATUS [GBLSTS_K_OLD_FORMAT_MSG] = TRUE; ! Mark this as an old format msg
300 0298 2 LOGFILE_HANDLER (REFORMAT_DESC);
301 0299 2 GLOBAL STATUS [GBLSTS_K_OLD_FORMAT_MSG] = FALSE;
```



```

305 0302 1 GLOBAL ROUTINE RPLY_HANDLER (BUFFER_DESC) : NOVALUE =
306 0303 1
307 0304 1 +-
308 0305 1 Functional description:
309 0306 1
310 0307 1 This routine is the handler for all RPLY messages received by OPCOM.
311 0308 1 This message is in the old format, and must be converted to the new
312 0309 1 format before it can be processed. Once this is done, the new format
313 0310 1 message handler is called to process the reformatted request.
314 0311 1
315 0312 1 Input:
316 0313 1
317 0314 1 BUFFER_DESC : The address of a quadword buffer descriptor that
318 0315 1 describes the buffer containing the message.
319 0316 1
320 0317 1 Implicit Input:
321 0318 1
322 0319 1 None.
323 0320 1
324 0321 1 Output:
325 0322 1
326 0323 1 None.
327 0324 1
328 0325 1 Implicit output:
329 0326 1
330 0327 1 Some accounting data will be updated
331 0328 1 to reflect the receipt of the message.
332 0329 1
333 0330 1 Side effects:
334 0331 1
335 0332 1 None.
336 0333 1
337 0334 1 Routine value:
338 0335 1
339 0336 1 None.
340 0337 1 --
341 0338 1
342 0339 2 BEGIN ! Start of RPLY_HANDLER
343 0340 2
344 0341 2 MAP
345 0342 2
346 0343 2 BUFFER_DESC : $ref_bblock;
347 0344 2
348 0345 2 EXTERNAL
349 0346 2 GLOBAL_STATUS : BITVECTOR, ! Global status flags
350 0347 2 DEVICE_FAO : $bblock; ! FAO control string
351 0348 2
352 0349 2 EXTERNAL ROUTINE
353 0350 2 REPLY_HANDLER : NOVALUE; ! New format msg handler
354 0351 2
355 0352 2 LOCAL
356 0353 2
357 0354 2 OLD_MSG_LEN : LONG, ! Length of old message
358 0355 2 NEW_MSG_LEN : LONG, ! Length of new message
359 0356 2 OLD_MSG : $ref_bblock, ! Pointer to start of old message
360 0357 2 NEW_MSG : $ref_bblock, ! Pointer to start of new message
361 0358 2 OUT_LENGTH : WORD, ! Length of formatted operator device name

```

```

362 0359 2          DEV_DESC      : $desc_block,          ! Operator device name descriptor
363 0360 2          REFORMAT_BUFFER : $bblock [OPC$K_MAXREAD], ! Buffer to hold the reformatted message
364 0361 2          REFORMAT_DESC  : $desc_block;         ! Descriptor for the REFORMAT_BUFFER
365 0362 2
366 0363 2
367 0364 2          ! Make sure the message is big enough. If not, it
368 0365 2          ! cannot possibly be a valid message, and OPCOM will
369 0366 2          ! simply ignore it.
370 0367 2
371 0368 2          IF .BUFFER_DESC [DSC$W_LENGTH] LSS (OPC$K_COMHDRSIZ + 13)
372 0369 2          THEN
373 0370 2              RETURN;
374 0371 2
375 0372 2          ! Copy the common header provided by $SENDPR to the new buffer
376 0373 2          !
377 0374 2          CHSMOVE (OPC$K_COMHDRSIZ, .BUFFER_DESC [DSC$A_POINTER], REFORMAT_BUFFER);
378 0375 2
379 0376 2
380 0377 2
381 0378 2          ! Zero the remainder of the REFORMAT_BUFFER.
382 0379 2
383 0380 2          CHSFILL (0, (OPC$K_MAXREAD - OPC$K_COMHDRSIZ), (REFORMAT_BUFFER + OPC$K_COMHDRSIZ));
384 0381 2
385 0382 2
386 0383 2          ! Move the old message fields into their corresponding places in the new message format.
387 0384 2
388 0385 2          OLD_MSG = .BUFFER_DESC [DSC$A_POINTER] + OPC$K_COMHDRSIZ; ! Set pointer to request text
389 0386 2          NEW_MSG = REFORMAT_BUFFER + OPC$K_COMHDRSIZ; ! Set pointer to start of new message.
390 0387 2          NEW_MSG [OPC$B_RQSTCODE] = .OLD_MSG [OPC$B_MS_TYPE]; ! Set message type
391 0388 2          NEW_MSG [OPC$B_SCOPE] = OPC$K_SYSTEM; ! Force to SYSTEM reply
392 0389 2          NEW_MSG [OPC$L_RQSTID] = .OLD_MSG [OPC$L_MS_RPLYID]; ! Set reply number
393 0390 2          NEW_MSG [OPC$L_RQ_OPTIONS] = .OLD_MSG [OPC$W_MS_STATUS] + OPC$FACILITY*16; ! Set reply status
394 0391 2
395 0392 2          ! Build the operator device name from the ASCII device string
396 0393 2          ! and the device unit number. Build the FAO OUTBUF descriptor
397 0394 2          ! to point to the correct spot within NEW_MSG to save a copy.
398 0395 2
399 0396 2          DEV_DESC [0,0,32,0] = 20; ! Allow for a large device name
400 0397 2          DEV_DESC [DSC$A_POINTER] = .NEW_MSG + $BYTEOFFSET (OPC$T_REPLY_OPR) + 1;
401 0398 2          $FAO (DEVICE FAO, OUT_LENGTH, DEV_DESC, OLD_MSG [OPC$T_MS_ONAME], .OLD_MSG [OPC$W_MS_OUNIT]);
402 0399 2          NEW_MSG [$BYTEOFFSET (OPC$T_REPLY_OPR),0,8,0] = .OUT_LENGTH;
403 0400 2          OLD_MSG_LEN = OPC$K_COMHDRSIZ + $BYTEOFFSET (OPC$L_MS_OTEXT);
404 0401 2          NEW_MSG_LEN = $BYTEOFFSET (OPC$T_REPLY_OPR) + .OUT_LENGTH + 1;
405 0402 2
406 0403 2          ! Check for the presence of some REPLY text.
407 0404 2
408 0405 2          IF .BUFFER_DESC [DSC$W_LENGTH] GTR .OLD_MSG_LEN
409 0406 2          THEN
410 0407 2              BEGIN
411 0408 2
412 0409 2              ! There is some reply text present. Copy it to
413 0410 2              ! the new message buffer.
414 0411 2
415 0412 2              CHSMOVE ( (.BUFFER_DESC [DSC$W_LENGTH] - .OLD_MSG_LEN),
416 0413 2                  OLD_MSG [$BYTEOFFSET (OPC$L_MS_OTEXT),0,0,0],
417 0414 2                  NEW_MSG [.NEW_MSG_LEN + 2,0,0,0]
418 0415 2              );

```

```
419 0416 NEW_MSG [NEW_MSG_LEN,0,16,0] = .BUFFER_DESC [DSCSW_LENGTH] - .OLD_MSG_LEN;  
420 0417 NEW_MSG_LEN = .NEW_MSG_LEN + .NEW_MSG [NEW_MSG_LEN,0,16,0];  
421 0418 END;  
422 0419 NEW_MSG_LEN = .NEW_MSG_LEN + 2;  
423 0420  
424 0421  
425 0422 Create a descriptor for the reformatted message.  
426 0423  
427 0424 REFORMAT_DESC [DSCSW_LENGTH] = .NEW_MSG_LEN + OPC$K_COMHDRSIZ;  
428 0425 REFORMAT_DESC [DSCSB_DTYPE] = 0;  
429 0426 REFORMAT_DESC [DSCSB_CLASS] = 0;  
430 0427 REFORMAT_DESC [DSCSA_POINTER] = REFORMAT_BUFFER;  
431 0428  
432 0429  
433 0430 Call the new-message handler to finish processing the message.  
434 0431  
435 0432 GLOBAL_STATUS [GBLSTS_K_OLD_FORMAT_MSG] = TRUE; ! Mark this as an old format msg  
436 0433 REPLY_HANDLER (REFORMAT_DESC);  
437 0434 GLOBAL_STATUS [GBLSTS_K_OLD_FORMAT_MSG] = FALSE;  
438 0435  
439 0436 1 END; ! End of RPLY_HANDLER
```

.EXTRN REPLY_HANDLER

09DA	8F	OC	AE	04	B8	26	28	00011	18:	.ENTRY RPLY_HANDLER, Save R2,R3,R4,R5,R6,R7,R8	0302
			00	6E	6E	00	2C	00017		MOVAB -2580(SP), SP	
			52	04	A8	32	AE	0001E		MOVVL BUFFER_DESC, R8	0368
					56	32	AE	9E 00025		MPW (R8), #51	
				01	A6	01	90	0002C		BGEQU 1\$	
			12	A6	04	A2	D0	00030		RET	
			06	A6	02	A2	3C	00035		MOVCL #38, 24(R8), REFORMAT_BUFFER	0375
			08	A6		05	A0	0003A		MOVCL #0, (SP), #0, #2522, REFORMAT_BUFFER+38	0380
			F8	AD	1B	A6	9E	00042		ADDL3 #38, 4(R8), OLD_MSG	0385
			FC	AD	08	A2	3C	00047		MOVAB REFORMAT_BUFFER+38, NEW_MSG	0386
					0A	A2	9F	0004B		MOVBL (OLD_MSG), (NEW_MSG)	0387
					F8	AD	9F	0004E		MOVBL #1, T(NEW_MSG)	0388
					OC	AE	9F	00051		MOVL 4(OLD_MSG), 18(NEW_MSG)	0389
			00000000G	00	CF	9F	00054			MOVZWL 2(OLD_MSG), 6(NEW_MSG)	0390
			1A	A6	05	FB	00058			ADDW2 #5, 8(NEW_MSG)	
				50	6E	90	0005F			MOVL #20, DEV_DESC	0396
				57	8F	9A	00063			MOVAB 27(R6), DEV_DESC+4	0397
				57	6E	3C	00067			MOVZWL 8(OLD_MSG), -(SP)	0398
50		68		10	1B	C0	0006A			PUSHAB 10(OLD_MSG)	
					00	ED	0006D			PUSHAB DEV_DESC	
					1C	15	00072			PUSHAB OUT_LENGTH	
					68	3C	00074			PUSHAB DEVICE FAO	
										CALLS #5, SYSSFAO	
										MOVBL OUT_LENGTH, 26(NEW_MSG)	0399
										MOVZBL #64, OLD_MSG_LEN	0400
										MOVZWL OUT_LENGTH, NEW_MSG_LEN	0401
										ADDL2 #27, NEW_MSG_LEN	
										CMPZV #0, #16, (R8), OLD_MSG_LEN	0405
										BLEQ 2\$	
										MOVZWL (R8), R8	0412

02	A746	1A	58	50	C2	00077	SUBL2	OLD_MSG_LEN, R8	:	0414
			A2	58	28	0007A	MOVCS	R8, 26(OLD_MSG), 2(NEW_MSG_LEN)[NEW_MSG]	:	0416
			9E	6746	9F	00081	PUSHAB	(NEW_MSG_LEN)[NEW_MSG]	:	0417
			50	6746	B0	00084	MOVW	R8, 4(SPT)+	:	
			57		9F	00087	PUSHAB	(NEW_MSG_LEN)[NEW_MSG]	:	
			57		3C	0008A	MOVZWL	@(SPT+, R0	:	
			57		50	0008D	ADDL2	R0, NEW_MSG_LEN	:	
04	AE		57		02	00090	ADDL2	#2, NEW_MSG_LEN	:	0419
					A1	00093	ADDW3	#38, NEW_MSG_LEN, REFORMAT_DESC	:	0424
				06	AE	00098	CLRW	REFORMAT_DESC+2	:	0425
		08	AE	0C	AE	0009B	MOVAB	REFORMAT_BUFFER, REFORMAT_DESC+4	:	0427
		0000G	CF		10	000A0	BISB2	#16, GLOBAL_STATUS	:	0432
				04	AE	000A5	PUSHAB	REFORMAT_DESC	:	0433
		0000G	CF		01	000A8	CALLS	#1, REPLY_HANDLER	:	
		0000G	CF		10	000AD	BICB2	#16, GLOBAL_STATUS	:	0434
					04	000B2	RET		:	0436

; Routine Size: 179 bytes, Routine Base: \$CODE\$ + 00DB

```

441 0437 1 GLOBAL ROUTINE RQST_HANDLER (BUFFER_DESC) : NOVALUE =
442 0438 1
443 0439 1 ++
444 0440 1 Functional description:
445 0441 1
446 0442 1 This routine is the handler for all RQST messages received by OPCOM.
447 0443 1 This message is in the old format, and must be converted to the new
448 0444 1 format before it can be processed. Once this is done, the new format
449 0445 1 message handler is called to process the reformatted request.
450 0446 1
451 0447 1 Input:
452 0448 1
453 0449 1 BUFFER_DESC : The address of a quadword buffer descriptor that
454 0450 1 describes the buffer containing the message.
455 0451 1
456 0452 1 Implicit Input:
457 0453 1
458 0454 1 None.
459 0455 1
460 0456 1 Output:
461 0457 1
462 0458 1 None.
463 0459 1
464 0460 1 Implicit output:
465 0461 1
466 0462 1 Some accounting data will be updated
467 0463 1 to reflect the receipt of the message.
468 0464 1
469 0465 1 Side effects:
470 0466 1
471 0467 1 None.
472 0468 1
473 0469 1 Routine value:
474 0470 1
475 0471 1 None.
476 0472 1 --
477 0473 1
478 0474 2 BEGIN ! Start of RQST_HANDLER
479 0475 2
480 0476 2 MAP
481 0477 2
482 0478 2 BUFFER_DESC : $ref_bblock;
483 0479 2
484 0480 2 EXTERNAL
485 0481 2 GLOBAL_STATUS : BITVECTOR; ! Global status flags
486 0482 2
487 0483 2 EXTERNAL ROUTINE
488 0484 2 REQUEST_HANDLER : NOVALUE; ! New format msg handler
489 0485 2
490 0486 2 LOCAL
491 0487 2
492 0488 2 OLD_MSG_LEN : LONG, ! Length of old message
493 0489 2 OLD_MSG : $ref_bblock, ! Pointer to start of old message
494 0490 2 NEW_MSG : $ref_bblock, ! Pointer to start of new message
495 0491 2 REFORMAT_BUFFER : $bblock [OP($K_MAXREAD)], ! Buffer to hold the reformatted message
496 0492 2 REFORMAT_DESC : $desc_block; ! Descriptor for the REFORMAT_BUFFER
497 0493 2

```


				.EXTRN REQUEST_HANDLER						
								.ENTRY	RQST_HANDLER, Save R2,R3,R4,R5,R6,R7	: 0437
								MOVAB	-2568(SP), SP	: 0499
								MOVL	BUFFER_DESC, R7	: 0506
								CMPW	(R7), #46	: 0511
								BLSSU	2\$: 0516
09DA	8F	08	AE	04	B7			MOVC3	#38, @4(R7), REFORMAT_BUFFER	: 0517
			00		6E			MOVC5	#0, (SP), #0, #2522, REFORMAT_BUFFER+38	: 0518
										: 0519
			50	04	A7			ADDL3	#38, 4(R7), OLD_MSG	: 0520
								MOVAB	REFORMAT_BUFFER+38, NEW_MSG	: 0521
								MOVW	(OLD_MSG), (NEW_MSG)	: 0522
								MOVW	#1, T(NEW_MSG)	: 0526
0A	A6	01	A0	01	A6			EXTZV	#0, #24, T(OLD_MSG), 10(NEW_MSG)	: 0529
								MOVL	4(OLD_MSG), 18(NEW_MSG)	: 0530
								MOVL	#46, OLD_MSG_LEN	: 0536
								CMPZV	#0, #16, (R7), OLD_MSG_LEN	: 0537
								BLEQ	1\$: 0539
								SUBW3	OLD_MSG_LEN, (R7), 26(NEW_MSG)	: 0544
								MOVC3	26(NEW_MSG), 8(OLD_MSG), 28(NEW_MSG)	: 0545
								ADDW3	#66, 28(NEW_MSG), REFORMAT_DESC	: 0546
								CLRW	REFORMAT_DESC+2	: 0548
								MOVAB	REFORMAT_BUFFER, REFORMAT_DESC+4	: 0548
								BISB2	#16, GLOBAL_STATUS	: 0548
								PUSHL	SP	: 0548
								CALLS	#1, REQUEST_HANDLER	: 0548
								BICB2	#16, GLOBAL_STATUS	: 0548
								RET		: 0548
										: 0548

; Routine Size: 114 bytes, Routine Base: \$CODE\$ + 018E

```

554 0549 1 GLOBAL ROUTINE SECU_HANDLER (BUFFER_DESC) : NOVALUE =
555 0550 1
556 0551 1 +-
557 0552 1 Functional description:
558 0553 1
559 0554 1 This routine is the handler for all SECURITY messages received by OPCOM.
560 0555 1 This message is in the old format, and must be converted to the new
561 0556 1 format before it can be processed. Once this is done, the new format
562 0557 1 message handler is called to process the reformatted request.
563 0558 1
564 0559 1 Input:
565 0560 1
566 0561 1 BUFFER_DESC : The address of a quadword buffer descriptor that
567 0562 1 describes the buffer containing the message.
568 0563 1
569 0564 1 Implicit Input:
570 0565 1
571 0566 1 None.
572 0567 1
573 0568 1 Output:
574 0569 1
575 0570 1 None.
576 0571 1
577 0572 1 Implicit output:
578 0573 1
579 0574 1 Some accounting data will be updated
580 0575 1 to reflect the receipt of the message.
581 0576 1
582 0577 1 Side effects:
583 0578 1
584 0579 1 None.
585 0580 1
586 0581 1 Routine value:
587 0582 1
588 0583 1 None.
589 0584 1 --
590 0585 1
591 0586 2 BEGIN ! Start of SECU_HANDLER
592 0587 2
593 0588 2 MAP
594 0589 2
595 0590 2 BUFFER_DESC : $ref_bblock;
596 0591 2
597 0592 2 EXTERNAL
598 0593 2 GLOBAL_STATUS : BITVECTOR; ! Global status flags
599 0594 2
600 0595 2 EXTERNAL ROUTINE
601 0596 2 SECURITY_HANDLER : NOVALUE; ! New format msg handler
602 0597 2
603 0598 2 LOCAL
604 0599 2
605 0600 2 OLD_MSG_LEN : LONG, ! Length of old message
606 0601 2 OLD_MSG : $ref_bblock, ! Pointer to start of old message
607 0602 2 NEW_MSG : $ref_bblock, ! Pointer to start of new message
608 0603 2 REFORMAT_BUFFER : $bblock [OP($K_MAXREAD)], ! Buffer to hold the reformatted message
609 0604 2 REFORMAT_DESC : $desc_block; ! Descriptor for the REFORMAT_BUFFER
610 0605 2

```

```

: 611 0606 2 |
: 612 0607 2 | Make sure the message is big enough. If not, it
: 613 0608 2 | cannot possibly be a valid message, and OPCOM will
: 614 0609 2 | simply ignore it.
: 615 0610 2 |
: 616 0611 2 | IF .BUFFER_DESC [DSC$W_LENGTH] LSS (OPC$K_COMHDRSIZ + 8)
: 617 0612 2 | THEN
: 618 0613 2 |     RETURN;
: 619 0614 2 |
: 620 0615 2 |
: 621 0616 2 |     Copy the commom header provided by $SENDPR to the new buffer
: 622 0617 2 |
: 623 0618 2 | CH$MOVE (OPC$K_COMHDRSIZ, .BUFFER_DESC [DSC$A_POINTER], REFORMAT_BUFFER);
: 624 0619 2 |
: 625 0620 2 |
: 626 0621 2 |     Zero the remainder of the REFORMAT_BUFFER.
: 627 0622 2 |
: 628 0623 2 | CH$FILL (0, (OPC$K_MAXREAD - OPC$K_COMHDRSIZ), (REFORMAT_BUFFER + OPC$K_COMHDRSIZ));
: 629 0624 2 |
: 630 0625 2 |
: 631 0626 2 |     Move the old message fields into their corresponding places in the new message format.
: 632 0627 2 |
: 633 0628 2 | OLD_MSG = .BUFFER_DESC [DSC$A_POINTER] + OPC$K_COMHDRSIZ;
: 634 0629 2 | NEW_MSG = REFORMAT_BUFFER + OPC$K_COMHDRSIZ;
: 635 0630 2 | NEW_MSG [OPC$B_RQSTCODE] = .OLD_MSG [OPC$B_MS_TYPE];           ! Set message code
: 636 0631 2 | NEW_MSG [OPC$B_SCOPE] = OPC$K_SYSTEM;                         ! Force SYSTEM request
: 637 0632 2 | NEW_MSG [OPC$L_ATTNUMASK1] = .OLD_MSG [$BYTEOFFSET (OPC$B_MS_TARGET), 0, 24, 0];
: 638 0633 2 | NEW_MSG [OPC$L_RQSTID] = .OLD_MSG [OPC$L_MS_RQSTID];
: 639 0634 2 | OLD_MSG_LEN = OPC$K_COMHDRSIZ + $BYTEOFFSET (OPC$L_MS_TEXT);
: 640 0635 2 |
: 641 0636 2 |     Copy the request text, if any, to the new message buffer.
: 642 0637 2 |
: 643 0638 2 | IF .BUFFER_DESC [DSC$W_LENGTH] GTR .OLD_MSG_LEN
: 644 0639 2 | THEN
: 645 0640 2 |     BEGIN
: 646 0641 2 |     NEW_MSG [OPC$W_REQUEST_LENGTH] = .BUFFER_DESC [DSC$W_LENGTH] - .OLD_MSG_LEN;
: 647 0642 2 |     CH$MOVE (.NEW_MSG [OPC$W_REQUEST_LENGTH], OLD_MSG [OPC$L_MS_TEXT], NEW_MSG [OPC$T_REQUEST_TEXT]);
: 648 0643 2 |     END;
: 649 0644 2 |
: 650 0645 2 |
: 651 0646 2 |     Create a descriptor for the reformatted message.
: 652 0647 2 |
: 653 0648 2 | REFORMAT_DESC [DSC$W_LENGTH] = OPC$K_COMHDRSIZ + OPC$K_HDR_SIZE + .NEW_MSG [OPC$W_REQUEST_LENGTH] + 2;
: 654 0649 2 | REFORMAT_DESC [DSC$B_DTYPE] = 0;
: 655 0650 2 | REFORMAT_DESC [DSC$B_CLASS] = 0;
: 656 0651 2 | REFORMAT_DESC [DSC$A_POINTER] = REFORMAT_BUFFER;
: 657 0652 2 |
: 658 0653 2 |
: 659 0654 2 |     Call the new-message handler to finish processing the message.
: 660 0655 2 |
: 661 0656 2 | GLOBAL_STATUS [GBLSTS_K_OLD_FORMAT_MSG] = TRUE;           ! Mark this as an old format msg
: 662 0657 2 | SECURITY_HANDLER (REFORMAT_DESC);
: 663 0658 2 | GLOBAL_STATUS [GBLSTS_K_OLD_FORMAT_MSG] = FALSE;
: 664 0659 2 |
: 665 0660 1 | END;

```

! End of SECU_HANDLER

				.EXTRN SECURITY_HANDLER			
				00FC 00000	.ENTRY	SECU_HANDLER, Save R2,R3,R4,R5,R6,R7	: 0549
			5E	F5F8 CE 9E 00002	MOVAB	-2568(SP), SP	: 0611
			57	04 AC D0 00007	MCVL	BUFFER_DESC, R7	: 0618
			2E	67 B1 0000B	CMPW	(R7), #46	: 0623
				61 1F 0000E	BLSSU	2\$: 0628
09DA	8F	08	AE	26 28 00010	MOVAB	#38, @4(R7), REFORMAT_BUFFER	: 0629
			00	00 2C 00016	MOVAB	#0, (SP), #0, #2522, REFORMAT_BUFFER+38	: 0630
			6E	2E AE 0001D	ADDL3	#38, 4(R7), OLD_MSG	: 0631
		50	04	26 C1 0001F	MOVAB	REFORMAT_BUFFER+38, NEW_MSG	: 0632
				2E AE 9E 00024	MOVAB	(OLD_MSG), (NEW_MSG)	: 0633
			66	60 90 00028	MOVAB	#1, T(NEW_MSG)	: 0634
			01	01 90 0002B	EXTZV	#0, #24, T(OLD_MSG), 10(NEW_MSG)	: 0641
0A	A6	01	A0	00 EF 0002F	MOVL	4(OLD_MSG), 18(NEW_MSG)	: 0642
			18	04 A0 D0 00036	MOVL	#46, OLD_MSG_LEN	: 0648
			12	2E D0 0003B	CMPZV	#0, #16, -(R7), OLD_MSG_LEN	: 0649
			51	00 ED 0003E	BLEQ	1\$: 0651
			10	0C 15 00043	SUBW3	OLD_MSG_LEN, (R7), 26(NEW_MSG)	: 0656
		1A	A6	51 A3 00045	MOVAB	26(NEW_MSG), 8(OLD_MSG), 28(NEW_MSG)	: 0657
		1C	A6	8F A1 00051 1\$:	ADDW3	#66, 28(NEW_MSG), REFORMAT_DESC	: 0660
			6E	02 AE B4 00058	CLRW	REFORMAT_DESC+2	: 0660
				08 AE 9E 0005B	MOVAB	REFORMAT_BUFFER, REFORMAT_DESC+4	: 0660
			04	10 88 00060	BISB2	#16, GLOBAL_STATUS	: 0660
			AE	5E DD 00065	PUSHL	SP	: 0660
			CF	01 FB 00067	CALLS	#1, SECURITY_HANDLER	: 0660
			0000G	10 8A 0006C	BICB2	#16, GLOBAL_STATUS	: 0660
			CF	04 00071 2\$:	RET		: 0660
			0000G				
			CF				

; Routine Size: 114 bytes, Routine Base: \$CODE\$ + 0200

```

667 0661 1 GLOBAL ROUTINE STS_HANDLER (BUFFER_DESC) : NOVALUE =
668 0662 1
669 0663 1 !++
670 0664 1 Functional description:
671 0665 1
672 0666 1 This routine is the handler for all STS messages received by OPCOM.
673 0667 1 This message is in the old format, and must be converted to the new
674 0668 1 format before it can be processed. Once this is done, the new format
675 0669 1 message handler is called to process the reformatted request.
676 0670 1
677 0671 1 Input:
678 0672 1
679 0673 1 BUFFER_DESC : The address of a quadword buffer descriptor that
680 0674 1 describes the buffer containing the message.
681 0675 1
682 0676 1 Implicit Input:
683 0677 1
684 0678 1 None.
685 0679 1
686 0680 1 Output:
687 0681 1
688 0682 1 None.
689 0683 1
690 0684 1 Implicit output:
691 0685 1
692 0686 1 Some accounting data will be updated
693 0687 1 to reflect the receipt of the message.
694 0688 1
695 0689 1 Side effects:
696 0690 1
697 0691 1 None.
698 0692 1
699 0693 1 Routine value:
700 0694 1
701 0695 1 None.
702 0696 1 --
703 0697 1
704 0698 2 BEGIN ! Start of STS_HANDLER
705 0699 2
706 0700 2 MAP
707 0701 2
708 0702 2 BUFFER_DESC : $ref_bblock;
709 0703 2
710 0704 2 EXTERNAL
711 0705 2 GLOBAL_STATUS : BITVECTOR, ! Global status flags
712 0706 2 DEVICE_FAO : $bblock; ! FAO control string
713 0707 2
714 0708 2 EXTERNAL ROUTINE
715 0709 2 STATUS_HANDLER : NOVALUE; ! New format msg handler
716 0710 2
717 0711 2 LOCAL
718 0712 2
719 0713 2 OLD_MSG : $ref_bblock, ! Pointer to start of old message
720 0714 2 NEW_MSG : $ref_bblock, ! Pointer to start of new message
721 0715 2 OUT_LENGTH : WORD, ! Length of formatted operator device name
722 0716 2 DEV_DESC : $desc_block, ! Operator device name descriptor
723 0717 2 REFORMAT_BUFFER : $bblock [OP($K_MAXREAD), ! Buffer to hold the reformatted message

```

```

: 724      0718      2      REFORMAT_DESC      : $desc_block;      ! Descriptor for the REFORMAT_BUFFER
: 725      0719      2
: 726      0720      2
: 727      0721      2      ! Make sure the message is big enough. If not, it
: 728      0722      2      ! cannot possibly be a valid message, and OPCOM will
: 729      0723      2      ! simply ignore it.
: 730      0724      2
: 731      0725      2      IF .BUFFER_DESC [DSC$W_LENGTH] LSS (OPC$K_COMHDR$SIZ + 7)
: 732      0726      2      THEN
: 733      0727      2          RETURN;
: 734      0728      2
: 735      0729      2      ! Copy the common header provided by $SENDPR to the new buffer
: 736      0730      2
: 737      0731      2
: 738      0732      2      CH$MOVE (OPC$K_COMHDR$SIZ, .BUFFER_DESC [DSC$A_POINTER], REFORMAT_BUFFER);
: 739      0733      2
: 740      0734      2
: 741      0735      2      ! Zero the remainder of the REFORMAT_BUFFER.
: 742      0736      2
: 743      0737      2      CH$FILL (0, (OPC$K_MAXREAD - OPC$K_COMHDR$SIZ), (REFORMAT_BUFFER + OPC$K_COMHDR$SIZ));
: 744      0738      2
: 745      0739      2
: 746      0740      2      ! Move the old message fields into their corresponding places in the new message format.
: 747      0741      2
: 748      0742      2      OLD_MSG = .BUFFER_DESC [DSC$A_POINTER] + OPC$K_COMHDR$SIZ;
: 749      0743      2      NEW_MSG = REFORMAT_BUFFER + OPC$K_COMHDR$SIZ;
: 750      0744      2      NEW_MSG [OPC$B_RQSTCODE] = .OLD_MSG [OPC$B_MS_TYPE];
: 751      0745      2      NEW_MSG [OPC$B_SCOPE] = OPC$K_SYSTEM;
: 752      0746      2
: 753      0747      2      ! Build the operator device name from the ASCII device string
: 754      0748      2      ! and the device unit number. Build the FAO OUTBUF descriptor
: 755      0749      2      ! to point to the correct spot within NEW_MSG to save a copy.
: 756      0750      2
: 757      0751      2      DEV_DESC [0,0,32,0] = 20;      ! Allow for a large device name
: 758      0752      2      DEV_DESC [DSC$A_POINTER] = .NEW_MSG + $BYTEOFFSET (OPC$T_STATUS_OPR) + 1;
: 759      0753      2      $FAO (DEVICE FAO, OUT_LENGTH, DEV_DESC, OLD_MSG [OPC$T_MS_ONAME], .OLD_MSG [OPC$W_MS_OUNIT]);
: 760      0754      2      NEW_MSG [$BYTEOFFSET (OPC$T_STATUS_OPR),0,8,0] = .OUT_LENGTH;
: 761      0755      2
: 762      0756      2
: 763      0757      2
: 764      0758      2      ! Create a descriptor for the reformatted message.
: 765      0759      2
: 766      0760      2      REFORMAT_DESC [DSC$W_LENGTH] = OPC$K_COMHDR$SIZ + OPC$K_HDR_SIZE + .OUT_LENGTH + 1;
: 767      0761      2      REFORMAT_DESC [DSC$B_DTYPE] = 0;
: 768      0762      2      REFORMAT_DESC [DSC$B_CLASS] = 0;
: 769      0763      2      REFORMAT_DESC [DSC$A_POINTER] = REFORMAT_BUFFER;
: 770      0764      2
: 771      0765      2
: 772      0766      2      ! Call the new-message handler to finish processing the message.
: 773      0767      2
: 774      0768      2      GLOBAL_STATUS [GBLSTS_K_OLD_FORMAT_MSG] = TRUE;      ! Mark this as an old format msg
: 775      0769      2      STATUS_HANDLER (REFORMAT_DESC);
: 776      0770      2      GLOBAL_STATUS [GBLSTS_K_OLD_FORMAT_MSG] = FALSE;
: 777      0771      2
: 778      0772      1      END;      ! End of STS_HANDLER
```

.EXTRN STATUS_HANDLER

```

.ENTRY STS_HANDLER, Save R2,R3,R4,R5,R6      ; 0661
MOVAB  -2580(SP), SP                          ;
MOVL   BUFFER_DESC, R6                       ; 0725
CMPW   (R6), #45                             ;
BLSSU  1$                                     ;
MOVCS  #38, 24(R6), REFORMAT_BUFFER         ; 0732
MOVCS  #0, (SP), #0, #2522, REFORMAT_BUFFER+38 ; 0737

ADDL3  #38, 4(R6), OLD_MSG                   ; 0742
MOVAB  REFORMAT_BUFFER+38, NEW_MSG          ; 0743
MOVB   (OLD_MSG), (NEW_MSG)                 ; 0744
MOVB   #1, T(NEW_MSG)                       ; 0745
MOVL   #20, DEV_DESC                         ; 0751
MOVAB  27(R2), DEV_DESC+4                   ; 0752
MOVZWL 8(OLD_MSG), -(SP)                     ; 0753
PUSHAB 10(OLD_MSG)                           ;
PUSHAB DEV_DESC                             ;
PUSHAB OUT_LENGTH                           ;
PUSHAB DEVICE_FAO                           ;
CALLS  #5, SYSSFAO                           ;
MOVB   OUT_LENGTH, 26(NEW_MSG)               ; 0754
ADDW3  #65, OUT_LENGTH, REFORMAT_DESC       ; 0760
CLRW   REFORMAT_DESC+2                      ; 0761
MOVAB  REFORMAT_BUFFER, REFORMAT_DESC+4     ; 0763
BISB2  #16, GLOBAL_STATUS                   ; 0768
PUSHAB REFORMAT_DESC                         ; 0769
CALLS  #1, STATUS_HANDLER                   ;
BICB2  #16, GLOBAL_STATUS                   ; 0770
RET                                          ; 0772

```

```

007C 00000
5E    F5EC CE 9E 00002
56    04  AC D0 00007
2D    66 B1 0000B
      65 1F 0000E
09DA 8F  OC AE 04 B6      26 28 00010
      00 2C 00016
      32  AE 0001D
      50  04 A6      26 C1 0001F
      52  32  AE 9E 00024
      62  60 90 00028
      01  A2 01 90 0002B
      F8  AD 14 D0 0002F
      FC  AD 1B A2 9E 00033
      7E  08 A0 3C 00038
      0A A0 9F 0003C
      F8 AD 9F 0003F
      0C AE 9F 00042
      0000G CF 9F 00045
      00000000G 00 05 FB 00049
      1A A2 6E 90 00050
04 AE 6E 8F A1 00054
      06 AE B4 0005B
      08 AE 9E 0005E
      0000G CF 10 88 00063
      04 AE 9F 00068
      0000G CF 01 FB 0006B
      0000G CF 10 8A 00070
      04 00075 1$:

```

; Routine Size: 118 bytes, Routine Base: \$CODE\$ + 0272

```

0773 1 GLOBAL ROUTINE TERME_HANDLER (BUFFER_DESC) : NOVALUE =
0774 1
0775 1 |++
0776 1 | Functional description:
0777 1 |
0778 1 |     This routine is the handler for all TERME messages received by OPCOM.
0779 1 |     This message is in the old format, and must be converted to the new
0780 1 |     format before it can be processed. Once this is done, the new format
0781 1 |     message handler is called to process the reformatted request.
0782 1 |
0783 1 | Input:
0784 1 |
0785 1 |     BUFFER_DESC : The address of a quadword buffer descriptor that
0786 1 |     describes the buffer containing the message.
0787 1 |
0788 1 | Implicit Input:
0789 1 |
0790 1 |     None.
0791 1 |
0792 1 | Output:
0793 1 |
0794 1 |     None.
0795 1 |
0796 1 | Implicit output:
0797 1 |
0798 1 |     Some accounting data will be updated
0799 1 |     to reflect the receipt of the message.
0800 1 |
0801 1 | Side effects:
0802 1 |
0803 1 |     None.
0804 1 |
0805 1 | Routine value:
0806 1 |
0807 1 |     None.
0808 1 | --
0809 1
0810 2 BFGIN                                ! Start of TERME_HANDLER
0811 2
0812 2 MAP
0813 2
0814 2     BUFFER_DESC      : $ref_bblock;
0815 2
0816 2 EXTERNAL ROUTINE
0817 2     OPRENABLE_HANDLER : NOVALUE;           ! New format message handler
0818 2
0819 2 EXTERNAL
0820 2     GLOBAL_STATUS    : BITVECTOR,        ! Global status flags
0821 2     DEVICE_FAO       : $bblock;         ! FAO control string
0822 2
0823 2 LOCAL
0824 2
0825 2     OLD_MSG          : $ref_bblock,      ! Pointer to start of old message
0826 2     NEW_MSG          : $ref_bblock,      ! Pointer to start of new message
0827 2     OUT_LENGTH       : WORD,            ! Length of formatted operator device name
0828 2     DEV_DESC         : $desc_block,     ! Operator device name descriptor
0829 2     REFORMAT_BUFFER : $bblock [OP($K_MAXREAD), ! Buffer to hold the reformatted message
0830 2
0831 2
0832 2
0833 2
0834 2
0835 2
0836 2

```



```

: 837      0830      2      REFORMAT_DESC      : $desc_block;      ! Descriptor for the REFORMAT_BUFFER
: 838      0831      2
: 839      0832      2
: 840      0833      2      ! Make sure the message is big enough.  If not, it
: 841      0834      2      ! cannot possibly be a valid message, and OPCOM will
: 842      0835      2      ! simply ignore it.
: 843      0836      2
: 844      0837      2      IF .BUFFER_DESC [DSC$W_LENGTH] LSS (OPC$K_COMHDRSIZ + 13)
: 845      0838      2      THEN
: 846      0839      2      RETURN;
: 847      0840      2
: 848      0841      2      ! Copy the common header provided by $SENDPR to the new buffer
: 849      0842      2
: 850      0843      2
: 851      0844      2      CH$MOVE (OPC$K_COMHDRSIZ, .BUFFER_DESC [DSC$A_POINTER], REFORMAT_BUFFER);
: 852      0845      2
: 853      0846      2      ! Zero the remainder of the REFORMAT_BUFFER.
: 854      0847      2
: 855      0848      2
: 856      0849      2      CH$FILL (0, (OPC$K_MAXREAD - OPC$K_COMHDRSIZ), (REFORMAT_BUFFER + OPC$K_COMHDRSIZ));
: 857      0850      2
: 858      0851      2
: 859      0852      2      ! Move the old message fields into their corresponding places in the new message format.
: 860      0853      2
: 861      0854      2      OLD_MSG = .BUFFER_DESC [DSC$A_POINTER] + OPC$K_COMHDRSIZ;      ! Set pointer to request text
: 862      0855      2      NEW_MSG = REFORMAT_BUFFER + OPC$K_COMHDRSIZ;      ! Set pointer to start of new message.
: 863      0856      2      NEW_MSG [OPC$B_RQSTCODE] = .OLD_MSG [OPC$B_MS_TYPE];      ! Set message type code
: 864      0857      2      NEW_MSG [OPC$B_SCOPE] = OPC$K_SYSTEM;      ! Force this to a SYSTEM enable
: 865      0858      2      $bblock [NEW_MSG [OPC$L_RQ_OPTIONS], OPC$V_PERMOPER] = TRUE;      ! Force this to a PERMANENT enable
: 866      0859      2      IF .OLD_MSG [$BYTEOFFSET (OPC$B_MS_ENAB),0,24,0] EQL 0
: 867      0860      2      THEN
: 868      0861      2      BEGIN
: 869      0862      3      $bblock [NEW_MSG [OPC$L_RQ_OPTIONS], OPC$V_DISABLE] = TRUE; ! This is a DISABLE request
: 870      0863      3      $bblock [NEW_MSG [OPC$L_RQ_OPTIONS], OPC$V_PERMOPER] = FALSE; ! Clear the PERMOPR bit
: 871      0864      2      END;
: 872      0865      2      NEW_MSG [OPC$L_ATTNMASK1] = .OLD_MSG [OPC$L_MS_MASK];      ! Set the enable/disable mask.
: 873      0866      2
: 874      0867      2      ! Create an operator device name from the device name ASCII
: 875      0868      2      ! string and the device unit #.  Just assume that FAO succeeded.
: 876      0869      2      ! Set up the OUTBUF descriptor so it points to the correct spot
: 877      0870      2      ! in NEW_MSG.  This will save a copy operation.
: 878      0871      2
: 879      0872      2      DEV_DESC [0,0,16,0] = 20;      ! Allow for a large device name
: 880      0873      2      DEV_DESC [DSC$A_POINTER] = .NEW_MSG + $BYTEOFFSET (OPC$T_OPENABLE_OPR) + 1;
: 881      0874      2      $FAO (DEVICE FAO, OUT_LENGTH, DEV_DESC, OLD_MSG [OPC$T_MS_ONAME], .OLD_MSG [OPC$W_MS_OUNIT]);
: 882      0875      2      NEW_MSG [$BYTEOFFSET (OPC$T_OPENABLE_OPR),0,8,0] = .OUT_LENGTH;
: 883      0876      2
: 884      0877      2
: 885      0878      2      ! Create a descriptor for the reformatted message.
: 886      0879      2
: 887      0880      2      REFORMAT_DESC [DSC$W_LENGTH] = OPC$K_COMHDRSIZ + OPC$K_HDR_SIZE + .OUT_LENGTH + 1;
: 888      0881      2      REFORMAT_DESC [DSC$B_DTYPE] = 0;
: 889      0882      2      REFORMAT_DESC [DSC$B_CLASS] = 0;
: 890      0883      2      REFORMAT_DESC [DSC$A_POINTER] = REFORMAT_BUFFER;
: 891      0884      2
: 892      0885      2
: 893      0886      2      ! Call the new-message handler to finish processing the message.

```

```

: 894      0887 2 !
: 895      0888 2 GLOBAL_STATUS [GBLSTS_K_OLD_FORMAT_MSG] = TRUE;      ! Mark this as an old format msg
: 896      0889 2 OPRENABLE_HANDLER (REFORMAT_DESC);
: 897      0890 2 GLOBAL_STATUS [GBLSTS_K_OLD_FORMAT_MSG] = FALSE;
: 898      0891 2
: 899      0892 1 END;

```

! End of TERME_HANDLER

.EXTRN OPRENABLE_HANDLER

```

.ENTRY TERME_HANDLER, Save R2,R3,R4,R5,R6      : 0773
MOVAB  -2580(SP), SP
MOVL   BUFFER_DESC, R6                        : 0837
CMPW   (R6), #51
BLSSU  2$
MOVCS  #38, @4(?), REFORMAT_BUFFER           : 0844
MOVCS  #0, (SP #0, #2522, REFORMAT_BUFFER+38 : 0849

ADDL3  #38, 4(R0), OLD_MSG                    : 0854
MOVAB  REFORMAT_BUFFER+38, NEW_MSG            : 0855
MOVB   (OLD_MSG), (NEW_MSG)                  : 0856
MOVB   #1, T(NEW_MSG)                         : 0857
BISB2  #2, 6(NEW_MSG)                         : 0858
CMPZV  #0, #24, T(OLD_MSG), #0               : 0859
BNEQ   1$
BISB2  #1, 6(NEW_MSG)                         : 0862
BICB2  #2, 6(NEW_MSG)                         : 0863
MOVL   4(OLD_MSG), 10(NEW_MSG)                : 0865
MOVW   #20, DEV_DESC                          : 0872
MOVAB  27(R2), DEV_DESC+4                     : 0873
MOVZWL 8(OLD_MSG), -(SP)                      : 0874
PUSHAB 10(OLD_MSG)
PUSHAB DEV_DESC
PUSHAB OUT_LENGTH
PUSHAB DEVICE_FAO
CALLS  #5, SY$FAO
MOVB   OUT_LENGTH, 26(NEW_MSG)                : 0875
ADDW3  #65, OUT_LENGTH, REFORMAT_DESC         : 0880
CLRW   REFORMAT_DESC+2                       : 0881
MOVAB  REFORMAT_BUFFER, REFORMAT_DESC+4      : 0883
BISB2  #16, GLOBAL_STATUS                    : 0888
PUSHAB REFORMAT_DESC                         : 0889
CALLS  #1, OPRENABLE_HANDLER
BICB2  #16, GLOBAL_STATUS                    : 0890
RET                                          : 0892

```

```

: 09DA 8F 0C AE 04 B6 5E FSEC CE 007C 00000
: 56 04 AC D0 00002
: 33 66 B1 0000B
: 7E 1F 0000E
: 26 28 00010
: 00 2C 00016
: 32 AE 0001D
: 26 C1 0001F
: 32 AE 9E 00024
: 62 60 90 00028
: 01 A2 01 90 0002B
: 06 A2 02 88 0002F
: 18 00 ED 00033
: 08 12 00039
: 06 A2 01 88 0003B
: 06 A2 02 8A 0003F
: 0A A2 04 A0 D0 00043 1$:
: FB AD 14 B0 00048
: FC AD 1B A2 9E 0004C
: 08 A0 3C 00051
: 0A A0 9F 00055
: FB AD 9F 00058
: 0C AE 9F 0005B
: 0000G CF 9F 0005E
: 0000000G 00 05 FB 00062
: 1A A2 6E 90 00069
: 04 AE 6E 0041 8F A1 0006D
: 06 AE B4 00074
: 08 AE 9E 00077
: 0000G CF 10 88 0007C
: 04 AE 9F 00081
: 0000G CF 01 FB 00084
: 0000G CF 10 8A 00089
: 04 0008E 2$:

```

; Routine Size: 143 bytes. Routine Base: \$CODE\$ + 02E8

```

: 900      0893 1
: 901      0894 1 END
: 902      0895 0 ELUDOM

```

! End of OPCOMOLD

PSECT SUMMARY

```
Name          Bytes          Attributes
SCODE$        887 NOVEC, NOWRT, RD, EXE, 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	20 0	1000	00:01.9
-\$255\$DUA28:[OPCOM.OBJ]OPCOMLIB.L32;1	633	24 3	43	00:00.9

COMMAND QUALIFIERS

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

: Size: 887 code + 0 data bytes
: Run Time: 00:22.0
: Elapsed Time: 01:12.2
: Lines/CPU Min: 2440
: Lexemes/CPU-Min: 20228
: Memory Used: 102 pages
: Compilation Complete

