

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 OPCSSTATUS
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 *
0012 0 * ALL RIGHTS RESERVED.
0013 0 *
0014 0 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0015 0 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0016 0 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0017 0 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0018 0 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0019 0 * TRANSFERRED.
0020 0 *
0021 0 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0022 0 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0023 0 * CORPORATION.
0024 0 *
0025 0 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0026 0 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
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 V03-002 CWH3001 CW Hobbs 30-Jul-1983
0054 0 Various and sundry things to make OPCOM distributed
0055 0 across the cluster.
0056 0
0057 0 V03-001 STJ3036 Steven T. Jeffreys, 07-Oct-1982

```

```

: 58      0058 0  |
: 59      0059 0  |
: 60      0060 0  |--
: 61      0061 0  |
: 62      0062 1  | BEGIN                | Start of STATUS
: 63      0063 1  |
: 64      0064 1  | LIBRARY 'SYSS$LIBRARY:LIB.L32';
: 65      0065 1  | LIBRARY 'LIBS:OPCOMLIB';
: 66      0066 1  |
: 67      0067 1  | FORWARD ROUTINE
: 68      0068 1  |     STATUS_HANDLER : NOVALUE;
: 69      0069 1  |
: 70      0070 1  | BUILTIN
: 71      0071 1  |
: 72      0072 1  |     INSQUE,           | Insert entry onto a queue
: 73      0073 1  |     REMQUE;          | Remove entry from a queue

```

```

75 0074 1 GLOBAL ROUTINE STATUS_HANDLER (BUFFER_DESC) : NOVALUE =
76 0075 1
77 0076 1 |**
78 0077 1 | Functional description:
79 0078 1 |
80 0079 1 |     This routine is the handler for all STATUS messages received by OPCOM.
81 0080 1 |
82 0081 1 |
83 0082 1 | Input:
84 0083 1 |
85 0084 1 |     BUFFER_DESC : The address of a quadword buffer descriptor that
86 0085 1 |                   describes the buffer containing the message.
87 0086 1 |
88 0087 1 | Implicit Input:
89 0088 1 |
90 0089 1 |     None.
91 0090 1 |
92 0091 1 | Output:
93 0092 1 |
94 0093 1 |     None.
95 0094 1 |
96 0095 1 | Implicit output:
97 0096 1 |
98 0097 1 |     Some accounting data will be updated
99 0098 1 |     to reflect the receipt of the message.
100 0099 1 |
101 0100 1 | Side effects:
102 0101 1 |
103 0102 1 |     None.
104 0103 1 |
105 0104 1 | Routine value:
106 0105 1 |
107 0106 1 |     None.
108 0107 1 | --
109 0108 1 |
110 0109 2 BEGIN                               ! Start of STATUS_HANDLER
111 0110 2
112 0111 2 MAP
113 0112 2
114 0113 2     BUFFER_DESC      : $ref_bblock;
115 0114 2
116 0115 2 EXTERNAL ROUTINE
117 0116 2     CHECK_REQUEST,           ! Common sanity checks
118 0117 2     DEALLOCATE_RQCB,        ! Dispose of an RQCB
119 0118 2     FIND_OPERATOR,          ! Locate a given operator
120 0119 2     FORMAT_MESSAGE,         ! Format an OPCOM message
121 0120 2     IMPLICITLY_CANCELED,    ! See if the request is dead
122 0121 2     IMPLIED_CANCEL,        ! Perform cancellation
123 0122 2     INTERPRET_MASK,        ! Interpret attention mask
124 0123 2     NOTIFY_OPERATOR,       ! Notify a given operator
125 0124 2     VALID_OPERATOR;      ! See if operator device is valid
126 0125 2
127 0126 2 LOCAL
128 0127 2     MESSAGE_VECTOR : VECTOR [5, LONG], ! Message info
129 0128 2     STATUS_DESC   : $desc_block,      ! Status message descriptor
130 0129 2     STATUS_BUF    : $bblock [OP($K_MAXREAD)], ! Status message buffer
131 0130 2     RQCB          : $ref_bblock,      ! RQCB data structure

```

```

132 0131 OPER_RQCB      : $ref_bblock,      ! ditto (for known operator)
133 0132 RQST_RQCB    : $ref_bblock,      ! ditto (for request RQCB)
134 0133 OCD        : $ref_bblock,      ! OCD data structure
135 0134 MCB        : $ref_bblock,      ! MCB data structure
136 0135 RQST_COUNT : LONG;             ! Count of requests
137 0136 STATOS     : LONG;
138 0137
139 0138
140 0139          ! Check the minimum request size. If not
141 0140          ! enough data, then ignore the message.
142 0141
143 0142          IF .BUFFER_DESC [DSC$W_LENGTH] LSS (OPC$K_COMHDRSIZ + OPC$K_STATUS_MIN_SIZE)
144 0143          THEN
145 0144              RETURN;
146 0145
147 0146          ! Do some common sanity checking, and get an RQCB.
148 0147
149 0148          IF NOT CHECK_REQUEST (.BUFFER_DESC, RQCB)
150 0149          THEN
151 0150              RETURN;
152 0151
153 0152          ! Make sure this is a valid operator device.
154 0153
155 0154          IF NOT VALID_OPERATOR (.BUFFER_DESC, .RQCB)
156 0155          THEN
157 0156              BEGIN
158 0157                  DEALLOCATE_RQCB (.RQCB);          ! Dismiss the request
159 0158                  RETURN;
160 0159              END;
161 0160
162 0161          ! See if this is a valid operator. If not, then dismiss the request.
163 0162
164 0163          IF NOT FIND_OPERATOR (.RQCB, OPER_RQCB)
165 0164          THEN
166 0165              BEGIN
167 0166                  MESSAGE_VECTOR [0] = OPC$_ILLRQST;      ! Set message code
168 0167                  MESSAGE_VECTOR [1] = 0;                ! Use current time of day
169 0168                  FORMAT_MESSAGE (.RQCB, MESSAGE_VECTOR); ! Format the message
170 0169                  NOTIFY_OPERATOR (.RQCB);                ! Notify the requestor
171 0170                  DEALLOCATE_RQCB (.RQCB);                ! Dispose of the RQCB
172 0171                  RETURN;                                  ! Dismiss the request
173 0172              END;
174 0173
175 0174          ! The operator is known to opcom. Send the operator
176 0175          ! the standard status message.
177 0176
178 0177          STATUS_DESC [0,0,32,0] = OPC$K_MAXREAD;          ! Set buffer size
179 0178          STATUS_DESC [DSC$A_POINTER] = STATUS_BUF;      ! Set buffer address
180 0179          IF NOT INTERPRET_MASK (OPER_RQCB [RQCB_L_ATTNMASK1], STATUS_DESC, STATUS_DESC)
181 0180          THEN
182 0181              BEGIN
183 0182                  DEALLOCATE_RQCB (.RQCB);
184 0183                  RETURN;
185 0184              END
186 0185          ELSE
187 0186              BEGIN
188 0187                  MESSAGE_VECTOR [0] = OPC$_OPERSTS;      ! Set message code

```

```

189 0188 3 MESSAGE_VECTOR [1] = 0; ! Use current date and time
190 0189 3 MESSAGE_VECTOR [2] = RQCB [RQCB_L_OPER_LEN]; ! Set operator device name
191 0190 3 MESSAGE_VECTOR [3] = STATUS_DESC; ! Set status message descriptor
192 0191 3 FORMAT_MESSAGE (.RQCB, MESSAGE_VECTOR); ! Format the message
193 0192 3 NOTIFY_OPERATOR (.RQCB); ! Send status message to the requestor
194 0193 3 END;
195 0194 2 !
196 0195 2 ! Scan through the list of request queued for this OCD.
197 0196 2 ! If the requestor is enabled to handle the request, inform
198 0197 2 ! the operator of the request.
199 0198 2 !
200 0199 2 OCD = .OPER_RQCB [RQCB_L_OCD]; ! Get OCD address
201 0200 2 RQST_COUNT = .OCD [OCD_W-RQSTCOUNT]; ! Get count of requests
202 0201 2 RQST_RQCB = .OCD [OCD [RQSTFLINK]]; ! Get address of first RQCB
203 0202 2 WHILE (.RQST_COUNT GTR 0) DO
204 0203 3 BEGIN
205 0204 3 !
206 0205 3 ! Compare the attention masks of the requests against
207 0206 3 ! the attention masks of the operator. If any bits are
208 0207 3 ! in common, then notify the operator of the request.
209 0208 3 !
210 0209 4 IF ((.OPER_RQCB [RQCB_L_ATTNMASK1] AND .RQST_RQCB [RQCB_L_ATTNMASK1]) NEQ 0)
211 0210 4 OR ((.OPER_RQCB [RQCB_L_ATTNMASK2] AND .RQST_RQCB [RQCB_L_ATTNMASK2]) NEQ 0)
212 0211 3 THEN
213 0212 3 IF NOT IMPLICITLY_CANCELED (.RQST_RQCB)
214 0213 3 THEN
215 0214 4 BEGIN
216 0215 4 MCB = .RQCB [RQCB_L_MCB]; ! Save MCB address
217 0216 4 RQCB [RQCB_L_MCB] = .RQST_RQCB [RQCB_L_MCB]; ! Inform the operator
218 0217 4 NOTIFY_OPERATOR (.RQCB); ! Restore MCB address
219 0218 4 RQCB [RQCB_L_MCB] = .MCB;
220 0219 3 END;
221 0220 3 RQST_COUNT = .RQST_COUNT - 1; ! Decrement the request count
222 0221 3 RQST_RQCB = .RQST_RQCB [RQCB_L_FLINK]; ! Get address of next request
223 0222 2 END;
224 0223 2 !
225 0224 2 IMPLIED_CANCEL (); ! Process any implicit cancellations
226 0225 2 DEALLOCATE_RQCB (.RQCB); ! Dismiss the request.
227 0226 2 !
228 0227 1 END; ! End of STATUS_HANDLER

```

```

.TITLE OPCSSTATUS
.IDENT \V04-000\

```

```

.EXTRN CHECK_REQUEST, DEALLOCATE_RQCB
.EXTRN FIND_OPERATOR, FORMAT_MESSAGE
.EXTRN IMPLICITLY_CANCELED
.EXTRN IMPLIED_CANCEL, INTERPRET_MASK
.EXTRN NOTIFY_OPERATOR
.EXTRN VALID_OPERATOR

```

```

.PSECT $CODE$,NOWRT,2

```

```

.ENTRY STATUS_HANDLER, Save R2,R3,R4,R5,R6,R7 ; 0074
MOVAB NOTIFY_OPERATOR, R7 ;
MOVAB -2596(SP), SP ;

```

```

00FC 0000
S7 0000G CF 9E 00002
SE F5DC CE 9E 00007

```

0044	8F	04	BC	B1	0000C	CMPW	@BUFFER_DESC, #68	0142
			01	1E	00012	BGEQU	1\$	
				04	00014	RET		
			5E	DD	00015	1\$:	PUSHL SP	0148
0000G	CF	04	AC	DD	00017		PUSHL BUFFER_DESC	
	01		02	FB	0001A		CALLS #2, CHECK_REQUEST	
			50	E8	0001F		BLBS R0, 2\$	
	53			04	00022		RET	
			6E	DO	00023	2\$:	MOVL R0CB, R3	0154
			53	DD	00026		PUSHL R3	
0000G	CF	04	AC	DD	00028		PUSHL BUFFER_DESC	
	27		02	FB	0002B		CALLS #2, VACID_OPERATOR	
			50	E9	00030		BLBC R0, 3\$	
		04	AE	9F	00033		PUSHAB OPER_R0CB	0163
			53	DD	00036		PUSHL R3	
0000G	CF		02	FB	00038		CALLS #2, FIND_OPERATOR	
	1D		50	E8	0003D		BLBS R0, 4\$	
EC	AD	0005807C	8F	DO	00040		MOVL #360572, MESSAGE_VECTOR	0166
			AD	D4	00048		CLRL MESSAGE_VECTOR+4	0167
			AD	9F	0004B		PUSHAB MESSAGE_VECTOR	0168
			53	DD	0004E		PUSHL R3	
0000G	CF		02	FB	00050		CALLS #2, FORMAT_MESSAGE	
	67		53	DD	00055		PUSHL R3	0169
			01	FB	00057		CALLS #1, NOTIFY_OPERATOR	
		008A	31	0005A	3\$:	BRW 9\$		0170
E4	AD	0A00	8F	3C	0005D	4\$:	MOVZWL #2560, STATUS_DESC	0177
E8	AD	08	AE	9E	00063		MOVAB STATUS_BUF, STATUS_DESC+4	0178
		E4	AD	9F	00068		PUSHAB STATUS_DESC	0179
		E4	AD	9F	0006B		PUSHAB STATUS_DESC	
	54		OC	AE	DO	0006E	MOVL OPER_R0CB, R4	
			5C	A4	9F	00072	PUSHAB 92(R4)	
0000G	CF		03	FB	00075		CALLS #3, INTERPRET_MASK	
	6A		50	E9	0007A		BLBC R0, 9\$	
EC	AD	000580C3	8F	DO	0007D		MOVL #360643, MESSAGE_VECTOR	0187
			AD	D4	00085		CLRL MESSAGE_VECTOR+4	0188
F4	AD	7C	A3	9E	00088		MOVAB 124(R3), MESSAGE_VECTOR+8	0189
F8	AD	E4	AD	9E	0008D		MOVAB STATUS_DESC, MESSAGE_VECTOR+12	0190
		EC	AD	9F	00092		PUSHAB MESSAGE_VECTOR	0191
			53	DD	00095		PUSHL R3	
0000G	CF		02	FB	00097		CALLS #2, FORMAT_MESSAGE	
	67		53	DD	0009C		PUSHL R3	0192
			01	FB	0009E		CALLS #1, NOTIFY_OPERATOR	
	50	24	A4	DO	000A1		MOVL 36(R4), OCD	0199
	55	3A	A0	3C	000A5		MOVZWL 58(OCD), R0ST_COUNT	0200
	52	3C	A0	DO	000A9		MOVL 60(OCD), R0ST_R0CB	0201
			55	D5	000AD	5\$:	TSTL R0ST_COUNT	0202
			31	15	000AF		BLEQ 8\$	
5C	A2	5C	A4	D3	000B1		BITL 92(R4), 92(R0ST_R0CB)	0209
			07	12	000B6		BNEQ 6\$	
60	A2	60	A4	D3	000B8		BITL 96(R4), 96(R0ST_R0CB)	0210
			1C	13	000BD		BEQL 7\$	
			52	DD	000BF	6\$:	PUSHL R0ST_R0CB	0212
0000G	CF		01	FB	000C1		CALLS #1, IMPLICITLY_CANCELED	
	12		50	E8	000C6		BLBS R0, 7\$	
	56	6C	A3	DO	000C9		MOVL 108(R3), MCB	0215
6C	A3	6C	A2	DO	000CD		MOVL 108(R0ST_R0CB), 108(R3)	0216
			53	DD	000D2		PUSHL R3	0217


```

        6C 67      01 FB 000D4      CALLS #1, NOTIFY_OPERATOR
        A3      56 DO 000D7      MOVL MCB, 108(R3)
        52      55 D7 000DB 7$:  DECL RQST_COUNT
        0000G CF 62 DO 000DD      MOVL (RQST_RQCB), RQST_RQCB
        0000G CF CB 11 000E0      BRB 5$
        0000G CF 00 FB 000E2 8$:  CALLS #0, IMPLIED_CANCEL
        0000G CF 53 DD 000E7 9$:  PUSHL R3
        01 FB 000E9      CALLS #1, DEALLOCATE_RQCB
        04 000EE      RET
    
```

; Routine Size: 239 bytes, Routine Base: \$CODE\$ + 0000

```

: 229      0228 1
: 230      0229 1 END
: 231      0230 0 ELUDOM
! End of STATUS
    
```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	239	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	6	0	1000	00:01.8
_\$255\$DUA28:[OPCOM.OBJ]OPCOMLIB.L32;1	633	13	2	43	00:00.8

COMMAND QUALIFIERS

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

```

: Size:      239 code + 0 data bytes
: Run Time:   00:07.6
: Elapsed Time: 00:30.1
: Lines/CPU Min: 1820
: Lexemes/CPU-Min: 10195
: Memory Used: 103 pages
: Compilation Complete
    
```


