

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$CLUMBX (
0002 0     LANGUAGE (BLISS32),
0003 0     IDENT = 'V04-000'
0004 0 ) =
0005 0
0006 0 *****
0007 0 *
0008 0 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 0 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 0 *  ALL RIGHTS RESERVED.
0011 0 *
0012 0 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0013 0 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0014 0 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0015 0 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0016 0 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0017 0 *  TRANSFERRED.
0018 0 *
0019 0 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0020 0 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0021 0 *  CORPORATION.
0022 0 *
0023 0 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0024 0 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0025 0 *
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     V03-004 CWH3169      CW Hobbs      5-May-1984
0054 0     Second pass for cluster-wide OPCOM:
0055 0     - Check cluster configuration every time we get a message
0056 0     from the connection manager. This is a reasonable approx-
0057 0     imation of a configuration change AST.

```

```

58 0058 0 | - Add the NOQUORUM message, it was apparently omitted
59 0059 0 | up until now.
60 0060 0 |
61 0061 0 | V03-003 CWH3003 CW Hobbs 14-Apr-1984
62 0062 0 | Fix the case range to account for the new messages.
63 0063 0 |
64 0064 0 | V03-002 CWH3002 CW Hobbs 12-Apr-1984
65 0065 0 | Make a couple of new messages, delete two old ones.
66 0066 0 |
67 0067 0 | V03-001 CWH3001 CW Hobbs 15-Sep-1983
68 0068 0 | Cut and paste DEVICE.B32 to be CLUMBX.B32.
69 0069 0 | --
70 0070 0 |
71 0071 1 BEGIN ! Start of CLUMBX
72 0072 1 |
73 0073 1 LIBRARY 'SYSS$LIBRARY:LIB.L32';
74 0074 1 LIBRARY 'LIBS:OPCOMLIB';
75 0075 1 |
76 0076 1 FORWARD ROUTINE
77 0077 1 CLUMBX_HANDLER : NOVALUE;
78 0078 1 |
79 0079 1 EXTERNAL ROUTINE
80 0080 1 ALLOCATE_DS, ! Allocate a data structure
81 0081 1 CLUSMSG_RQCB_SEND, ! Send RQCB to remote nodes
82 0082 1 CLUSUTIC_CONFIGURE, ! Check to see who is in the club
83 0083 1 DEALLOCATE_RQCB : NOVALUE, ! Dispose of an RQCB
84 0084 1 DUMP_LOG_FILE, ! Put a random string in the log file
85 0085 1 FORMAT_MESSAGE, ! Format a message
86 0086 1 LOG_MESSAGE, ! Log an event
87 0087 1 NOTIFY_LISTED_OPERATORS; ! Notify interested operators
88 0088 1 |
89 0089 1 EXTERNAL
90 0090 1 OCD_VECTOR : VECTOR; ! OCD list heads
91 0091 1 |
92 0092 1 EXTERNAL LITERAL
93 0093 1 RQCB_K_TYPE, ! RQCB structure type
94 0094 1 MIN_SCOPE, ! Minimum scope value
95 0095 1 MAX_SCOPE; ! Maximum scope value

```

```

: 97      0096 1 GLOBAL ROUTINE CLUMBX_HANDLER (BUFFER_DESC : $ref_bblock) : NOVALUE =
: 98      0097 1
: 99      0098 1 !++
100     0099 1 ! Functional description:
101     0100 1
102     0101 1         This routine is the handler for all message of type MSG$CLUMBX.
103     0102 1         Note that this message has a special format, different from all other
104     0103 1         messages. This is because the message is sent directly from the
105     0104 1         connection manager instead of $SNDOPR.
106     0105 1
107     0106 1
108     0107 1
109     0108 1 Input:
110     0109 1
111     0110 1         BUFFER_DESC : The address of a quadword buffer descriptor that
112     0111 1         describes the buffer containing the message.
113     0112 1
114     0113 1 Implicit Input:
115     0114 1
116     0115 1         None.
117     0116 1
118     0117 1 Output:
119     0118 1
120     0119 1         None.
121     0120 1
122     0121 1 Routine value:
123     0122 1
124     0123 1         None.
125     0124 1 --
126     0125 1
127     0126 2 BEGIN                                     ! Start of CLUMBX_HANDLER
128     0127 2
129     0128 2 LOCAL
130     0129 2     MESSAGE_VECTOR : VECTOR [15, LONG],      ! Message info
131     0130 2     MSG              : $ref_bblock,          ! Pointer to message text
132     0131 2     RQCB             : $ref_bblock,          ! RQCB data structure
133     0132 2     OCD              : $ref_bblock,          ! OCD data structure
134     0133 2     OCD_COUNT        : LONG,                ! Count of OCDs in OCD list
135     0134 2     OCD_INDEX        : LONG,                ! Index into OCD_VECTOR
136     0135 2     OPER_COUNT       : LONG,                ! Count of operators in operator list
137     0136 2     STATOS           : LONG;
138     0137 2 !
139     0138 2 ! See if the message is correct
140     0139 2 !
141     0140 2 MSG = .BUFFER_DESC [DSC$A_POINTER];
142     0141 2 IF .BUFFER_DESC [DSC$W_LENGTH] NEQ CLUMBX$K_LENGTH
143     0142 2 OR
144     0143 2     .MSG [CLUMBX$B_DS_VERSION] NEQ CLUMBX$K_DS_VERSION
145     0144 2 OR
146     0145 2     .MSG [CLUMBX$W_LENGTH] NEQ CLUMBX$K_LENGTH
147     0146 2 THEN
148     0147 2     RETURN DUMP_LOG_FILE (.BUFFER_DESC, %ASCID 'CLUMBX message not valid');
149     0148 2 !
150     0149 2 ! Run a scan to see if the set of players visible through $GETSYI has changed
151     0150 2 !
152     0151 2 CLUSUTIL_CONFIGURE ();
153     0152 2 !

```

```

154 0153 2 ! The message must be one of the cluster mailbox messages. All others
155 0154 2 ! are ignored. Set the message code in the message vector to the
156 0155 2 ! proper value.
157 0156 2
158 0157 2 CASE .MSG [CLUMBX$W_SUBTYPE] FROM 1 TO CLUMBX$K_ST_QUORUM
159 0158 2 OF
160 0159 2 SET
161 0160 2 [ CLUMBX$K_ST_NEWSYS ] : MESSAGE_VECTOR [0] = OPC$_CLU_NEWSYS;
162 0161 2 [ CLUMBX$K_ST_CNX ] : MESSAGE_VECTOR [0] = OPC$_CLU_CNX;
163 0162 2 [ CLUMBX$K_ST_REC�X ] : MESSAGE_VECTOR [0] = OPC$_CLU_REC�X;
164 0163 2 [ CLUMBX$K_ST_LOSTCNX ] : MESSAGE_VECTOR [0] = OPC$_CLU_LOSTCNX;
165 0164 2 [ CLUMBX$K_ST_TIMCNX ] : MESSAGE_VECTOR [0] = OPC$_CLU_TIMCNX;
166 0165 2 [ CLUMBX$K_ST_INIFORM ] : MESSAGE_VECTOR [0] = OPC$_CLU_INIFORM;
167 0166 2 [ CLUMBX$K_ST_INIADD ] : MESSAGE_VECTOR [0] = OPC$_CLU_INIADD;
168 0167 2 [ CLUMBX$K_ST_INIRECNFIG ] : MESSAGE_VECTOR [0] = OPC$_CLU_INIRECNFIG;
169 0168 2 [ CLUMBX$K_ST_MEMREQ ] : MESSAGE_VECTOR [0] = OPC$_CLU_MEMREQ;
170 0169 2 [ CLUMBX$K_ST_ABORT ] : MESSAGE_VECTOR [0] = OPC$_CLU_ABORT;
171 0170 2 [ CLUMBX$K_ST_ADD ] : MESSAGE_VECTOR [0] = OPC$_CLU_ADD;
172 0171 2 [ CLUMBX$K_ST_NEWNODE ] : MESSAGE_VECTOR [0] = OPC$_CLU_ADD;
173 0172 2 [ CLUMBX$K_ST_DROPNODE ] : MESSAGE_VECTOR [0] = OPC$_CLU_DROPNODE;
174 0173 2 [ CLUMBX$K_ST_FORNCLUS ] : MESSAGE_VECTOR [0] = OPC$_CLU_FORNCLUS;
175 0174 2 [ CLUMBX$K_ST_INQUORUM ] : MESSAGE_VECTOR [0] = OPC$_CLU_INQUORUM;
176 0175 2 [ CLUMBX$K_ST_LOSTDISK ] : MESSAGE_VECTOR [0] = OPC$_CLU_LOSTDISK;
177 0176 2 [ CLUMBX$K_ST_GAINDISK ] : MESSAGE_VECTOR [0] = OPC$_CLU_GAINDISK;
178 0177 2 [ CLUMBX$K_ST_DISKRDERR ] : MESSAGE_VECTOR [0] = OPC$_CLU_DISKRDERR;
179 0178 2 [ CLUMBX$K_ST_DISKWRERR ] : MESSAGE_VECTOR [0] = OPC$_CLU_DISKWRERR;
180 0179 2 [ CLUMBX$K_ST_DISKINVDAT ] : MESSAGE_VECTOR [0] = OPC$_CLU_DISKINVDAT;
181 0180 2 [ CLUMBX$K_ST_DISKTIMEOUT ] : MESSAGE_VECTOR [0] = OPC$_CLU_DISKTIMEOUT;
182 0181 2 [ CLUMBX$K_ST_LOSTMSG ] : MESSAGE_VECTOR [0] = OPC$_CLU_LOSTMSG;
183 0182 2 [ CLUMBX$K_ST_NOQUORUM ] : MESSAGE_VECTOR [0] = OPC$_CLU_NOQUORUM;
184 0183 2 [ CLUMBX$K_ST_FORNDISK ] : MESSAGE_VECTOR [0] = OPC$_CLU_FORNDISK;
185 0184 2 [ CLUMBX$K_ST_COMPLETE ] : MESSAGE_VECTOR [0] = OPC$_CLU_COMPLETE;
186 0185 2 [ CLUMBX$K_ST_QUORUM ] : MESSAGE_VECTOR [0] = OPC$_CLU_QUORUM;
187 0186 2 [ INRANGE, OUTRANGE ] : RETURN DUMP_LOG_FILE (.BUFFER_DESC, %ASCID 'CLUMBX message subtype not v
188 0187 2 TES;
189 0188 2
190 0189 2 ! To put additional stress on the system, request acks from every node. Make
191 0190 2 ! all this noise switchable by defining the logical name OPC$ACK (presence of
192 0191 2 ! the name is significant, not its value).
193 0192 2
194 L 0193 2 %IF %VARIANT GTR 0 ! Only include if /VARIANT specified
195 U 0194 2 %THEN
196 U 0195 2 BEGIN
197 U 0196 2 LOCAL lcl_buf : $bvector [16], out_dsc : VECTOR [2, LONG];
198 U 0197 2 EXTERNAL lcl_csid, nod_head : vector [2, long];
199 U 0198 2 EXTERNAL ROUTINE clu$msg_ack_please;
200 U 0199 2 out_dsc [0] = 16; out_dsc [1] = lcl_buf;
201 U 0200 2 IF $strniog (lognam=%ASCID 'OPC$ACK', rslLen=out_dsc, rslbuf=out_dsc) EQL ss$_normal
202 U 0201 2 THEN
203 U 0202 2 BEGIN
204 U 0203 2 LOCAL nod : $ref_bblock;
205 U 0204 2 nod = .nod_head [0];
206 U 0205 2 WHILE .nod NEQ .nod_head [0]
207 U 0206 2 DO
208 U 0207 2 BEGIN
209 U 0208 2 IF .nod [nod_l_node_csid] NEQ .lcl_csid
210 U 0209 2 THEN

```

```

211 U 0210 2 BEGIN
212 U 0211 2     nod [nod_v_ack_pend] = false;
213 U 0212 2     CLUSMSG_ACR_PLEASE (.nod);
214 U 0213 2     END;
215 U 0214 2     nod = .nod [nod_l_flink];
216 U 0215 2     END;
217 U 0216 2     END;
218 U 0217 2 END;
219 U 0218 2 XFI
220 U 0219 2 ! End of conditionally compiled message codes.
221 U 0220 2 ! Allocate an RQCB. This is necessary to format and later issue the message.
222 U 0221 2
223 U 0222 2 IF NOT ALLOCATE_DS (RQCB_K_TYPE, RQCB)
224 U 0223 2 THEN
225 U 0224 2     RETURN;
226 U 0225 2
227 U 0226 2 ! Set the operator interest mask so that cluster operators will find it
228 U 0227 2
229 U 0228 2 RQCB [RQCB_L_ATTNUMASK1] = OPCSM_NM_CLUSTER;
230 U 0229 2
231 U 0230 2 ! Format the message, then send it to everyone else in the cluster
232 U 0231 2
233 U 0232 2 MESSAGE_VECTOR [1] = 0; ! Use current system time for header
234 U 0233 2 MESSAGE_VECTOR [2] = MSG [CLUMBX$Q_TIME]; ! Put the event time in the message
235 U 0234 2
236 U 0235 2 ! Move information for local system
237 U 0236 2
238 U 0237 2 MESSAGE_VECTOR [3] = .(MSG [CLUMBX$T_NODENAME_L])<0,8,0>; ! Get ASCII length of nodename
239 U 0238 2 MESSAGE_VECTOR [4] = MSG [CLUMBX$T_NODENAME_L] + 1; ! Pass address of nodename
240 U 0239 2 MESSAGE_VECTOR [5] = .(MSG [CLUMBX$B_SYSTEMID_L] + 4)<0,16,0>; ! High word of system id
241 U 0240 2 MESSAGE_VECTOR [6] = .(MSG [CLUMBX$B_SYSTEMID_L]); ! Low longword of system id
242 U 0241 2 MESSAGE_VECTOR [7] = .MSG [CLUMBX$L_CSID_L]; ! Cluster system id of system
243 U 0242 2
244 U 0243 2 ! Move information for remote system
245 U 0244 2
246 U 0245 2 MESSAGE_VECTOR [8] = .(MSG [CLUMBX$T_NODENAME_R])<0,8,0>; ! Get ASCII length of nodename
247 U 0246 2 MESSAGE_VECTOR [9] = MSG [CLUMBX$T_NODENAME_R] + 1; ! Pass address of nodename
248 U 0247 2 MESSAGE_VECTOR [10] = .(MSG [CLUMBX$B_SYSTEMID_R] + 4)<0,16,0>; ! High word of system id
249 U 0248 2 MESSAGE_VECTOR [11] = .(MSG [CLUMBX$B_SYSTEMID_R]); ! Low longword of system id
250 U 0249 2 MESSAGE_VECTOR [12] = .MSG [CLUMBX$L_CSID_R]; ! Cluster system id of system
251 U 0250 2
252 U 0251 2 ! Now actually format it
253 U 0252 2
254 U 0253 2 FORMAT_MESSAGE (.RQCB, MESSAGE_VECTOR);
255 U 0254 2
256 U 0255 2 ! If desired, send it to remote nodes
257 U 0256 2
258 U 0257 2 IF .MSG [CLUMBX$V_BRDCST]
259 U 0258 2 THEN
260 U 0259 2     CLUSMSG_RQCB_SEND (-1, CLM_CLUMBX, .RQCB); ! Send it everywhere
261 U 0260 2
262 U 0261 2 ! Log the message, and send it to all interested operators.
263 U 0262 2 ! Every operator in the data base is a candidate for the mes age.
264 U 0263 2
265 U 0264 2 OCD_INDEX = MAX SCOPE;
266 U 0265 2 WHILE (.OCD_INDEX GEQ MIN_SCOPE) DO
267 U 0266 3 BEGIN

```

```

: 268 0267 3 |
: 269 0268 3 | Scan the OCD list for each class of operator.
: 270 0269 3 |
: 271 0270 3 |
: 272 0271 3 | OCD_COUNT = .OCD VECTOR [(OCD_INDEX - 1) * 2 + 1];
: 273 0272 3 | OCD = .OCD VECTOR [(OCD_INDEX - 1) * 2];
: 274 0273 4 | WHILE (.OCD_COUNT GTR 0) DO
: 275 0274 4 | BEGIN
: 276 0275 4 | | Notify every operator in the OCD's operator list.
: 277 0276 4 | | Also log the message for each OCD.
: 278 0277 4 |
: 279 0278 4 | RQCB [RQCB L OCD] = .OCD; | Set OCD address
: 280 0279 4 | LOG MESSAGE T.RQCB); | Log the message
: 281 0280 4 | NOTIFY LISTED OPERATORS (.RQCB); | Inform the operators
: 282 0281 4 | OCD_COUNT = .OCD_COUNT - 1; | Decrement operator count
: 283 0282 4 | OCD = .OCD [OCD[_FLINK]]; | Get next OCD address
: 284 0283 3 | END;
: 285 0284 3 | OCD_INDEX = .OCD_INDEX - 1;
: 286 0285 2 | END;
: 287 0286 2 |
: 288 0287 2 | | Free the rqcb
: 289 0288 2 |
: 290 0289 2 | DEALLOCATE_RQCB (.RQCB);
: 291 0290 2 | RETURN;
: 292 0291 2 |
: 293 0292 1 | END; | End of CLUMBX_HANDLER

```

```

: .TITLE OPC$CLUMBX
: .IDENT \V04-000\
: .PSECT $SPLITS,NOWRT,NOEXE,2
20 65 67 61 73 73 65 6D 20 58 42 4D 55 4C 43 0000 P.AAB: .ASCII \CLUMBX message not valid\
6C 61 76 20 74 6F 6E 20 6C 61 76 20 74 6F 6E 000F
: 010E0018 00018 P.AAA: .LONG 17694744
: 00000000' 0001C .ADDRESS P.AAB
20 65 67 61 73 73 65 6D 20 58 42 4D 55 4C 43 00020 P.AAD: .ASCII \CLUMBX message subtype not valid\
6C 61 76 20 74 6F 6E 20 65 70 79 74 62 75 73 0002F
: 64 69 0003E
: 010E0020 00040 P.AAC: .LONG 17694752
: 00000000' 00044 .ADDRESS P.AAD
: .EXTRN ALLOCATE_DS, CLUSMSG_RQCB_SEND
: .EXTRN CLUSUTIL_CONFIGURE
: .EXTRN DEALLOCATE_RQCB
: .EXTRN DUMP_LOG_FILE, FORMAT_MESSAGE
: .EXTRN LOG_MESSAGE, NOTIFY_LISTED_OPERATORS
: .EXTRN OCD_VECTOR, RQCB_K_TYPE
: .EXTRN MIN_SCOPE, MAX_SCOPE
: .PSECT $CODE$,NOWRT,2
: .ENTRY CLUMBX_HANDLER, Save R2,R3,R4,R5 : 0096
: MOVAB -64(SP), SP : 0140
: MOVL BUFFER_DESC, R3
: MOVL 4(R3), MSG

```


0062
008A
00A8
00BC
00E4
010C
0134

1B
0058
008C
0038
00B2
00DA
0102
012A

0048	8F		63	B1	0000E		CMPW	(R3), #72		0141
			0E	12	00013		BNEQ	1\$		
	01	04	A2	91	00015		CMPB	4(MSG), #1		0143
			08	12	00019		BNEQ	1\$		
0048	8F		A2	B1	0001B		CMPW	6(MSG), #72		0145
			06	13	00021		BEQL	2\$		
		0000'	CF	9F	00023	1\$:	PUSHAB	P.AAA		0147
			46	11	00027		BRB	5\$		
0000G	CF		00	FB	00029	2\$:	CALLS	#0, CLUSUTIL CONFIGURE		0151
	01	02	A2	AF	0002E		CASEW	2(MSG), #1, #27		0157
	004E		0044		00033	3\$:	.WORD	6\$-3\$,-		
	0076		006C		0003B			7\$-3\$,-		
	009E		0094		00043			8\$-3\$,-		
	00A8		0038		0004B			9\$-3\$,-		
	00D0		00C6		00053			10\$-3\$,-		
	00F8		00EE		0005B			11\$-3\$,-		
	0120		0116		00063			12\$-3\$,-		
								13\$-3\$,-		
								14\$-3\$,-		
								15\$-3\$,-		
								4\$-3\$,-		
								16\$-3\$,-		
								4\$-3\$,-		
								16\$-3\$,-		
								17\$-3\$,-		
								18\$-3\$,-		
								20\$-3\$,-		
								22\$-3\$,-		
								24\$-3\$,-		
								26\$-3\$,-		
								28\$-3\$,-		
								30\$-3\$,-		
								32\$-3\$,-		
								34\$-3\$,-		
								36\$-3\$,-		
								38\$-3\$,-		
								40\$-3\$,-		
								42\$-3\$,-		
		0000'	CF	9F	0006B	4\$:	PUSHAB	P.AAC		0186
			53	DD	0006F	5\$:	PUSHL	R3		
0000G	CF		02	FB	00071		CALLS	#2, DUMP_LOG_FILE		
				04	00076		RET			
04	AE	000582DB	8F	D0	00077	6\$:	MOVL	#361179, MESSAGE_VECTOR		0160
			76	11	0007F		BRB	19\$		
04	AE	000582E3	8F	D0	00081	7\$:	MOVL	#361187, MESSAGE_VECTOR		0161
			76	11	00089		BRB	21\$		
04	AE	000582EB	8F	D0	0008B	8\$:	MOVL	#361195, MESSAGE_VECTOR		0162
			76	11	00093		BRB	23\$		
04	AE	000582F3	8F	D0	00095	9\$:	MOVL	#361203, MESSAGE_VECTOR		0163
			76	11	0009D		BRB	25\$		
04	AE	000582FB	8F	D0	0009F	10\$:	MOVL	#361211, MESSAGE_VECTOR		0164
			76	11	000A7		BRB	27\$		
04	AE	00058303	8F	D0	000A9	11\$:	MOVL	#361219, MESSAGE_VECTOR		0165
			76	11	000B1		BRB	29\$		
04	AE	0005830B	8F	D0	000B3	12\$:	MOVL	#361227, MESSAGE_VECTOR		0166
			76	11	000BB		BRB	31\$		
04	AE	00058313	8F	D0	000BD	13\$:	MOVL	#361235, MESSAGE_VECTOR		0167

04	AE	0005831B	76	11	000C5	BRB	33\$				
			8F	DO	000C7	14\$:	MOVL	#361243,	MESSAGE_VECTOR		0168
04	AE	00058323	76	11	000CF	BRB	35\$				
			8F	DO	000D1	15\$:	MOVL	#361251,	MESSAGE_VECTOR		0169
04	AE	0005832B	76	11	000D9	BRB	37\$				
			8F	DO	000DB	16\$:	MOVL	#361259,	MESSAGE_VECTOR		0171
04	AE	00058333	76	11	000E3	BRB	39\$				
			8F	DO	000E5	17\$:	MOVL	#361267,	MESSAGE_VECTOR		0172
04	AE	0005833B	76	11	000ED	BRB	41\$				
			8F	DO	000EF	18\$:	MOVL	#361275,	MESSAGE_VECTOR		0173
04	AE	00058343	76	11	000F7	BRB	43\$				
			8F	DO	000F9	20\$:	MOVL	#361283,	MESSAGE_VECTOR		0174
04	AE	0005834B	6C	11	00101	BRB	43\$				
			8F	DO	00103	22\$:	MOVL	#361291,	MESSAGE_VECTOR		0175
04	AE	00058353	62	11	0010B	BRB	43\$				
			8F	DO	0010D	24\$:	MOVL	#361299,	MESSAGE_VECTOR		0176
04	AE	0005835B	58	11	00115	BRB	43\$				
			8F	DO	00117	26\$:	MOVL	#361307,	MESSAGE_VECTOR		0177
04	AE	00058363	4E	11	0011F	BRB	43\$				
			8F	DO	00121	28\$:	MOVL	#361315,	MESSAGE_VECTOR		0178
04	AE	0005836B	44	11	00129	BRB	43\$				
			8F	DO	0012B	30\$:	MOVL	#361323,	MESSAGE_VECTOR		0179
04	AE	00058373	3A	11	00133	BRB	43\$				
			8F	DO	00135	32\$:	MOVL	#361331,	MESSAGE_VECTOR		0180
04	AE	0005837B	30	11	0013D	BRB	43\$				
			8F	DO	0013F	34\$:	MOVL	#361339,	MESSAGE_VECTOR		0181
04	AE	00058383	26	11	00147	BRB	43\$				
			8F	DO	00149	36\$:	MOVL	#361347,	MESSAGE_VECTOR		0182
04	AE	0005838B	1C	11	00151	BRB	43\$				
			8F	DO	00153	38\$:	MOVL	#361355,	MESSAGE_VECTOR		0183
04	AE	00058393	12	11	0015B	BRB	43\$				
			8F	DO	0015D	40\$:	MOVL	#361363,	MESSAGE_VECTOR		0184
04	AE	0005839B	08	11	00165	BRB	43\$				
			8F	DO	00167	42\$:	MOVL	#361371,	MESSAGE_VECTOR		0185
		00000000G	5E	DD	0016F	43\$:	PUSHL	SP			0222
0000G	CF		8F	DD	00171		PUSHL	#RQCB_K_TYPE			
	01		02	FB	00177		CALLS	#2, ALLOCATE_DS			
			50	E8	0017C		BLBS	R0, 44\$			
				04	0017F		RET				
	53		6E	DO	00180	44\$:	MOVL	RQCB, R3			0228
5C	A3	80	8F	9A	00183		MOVZBL	#128, 92(R3)			
		08	AE	D4	00188		CLRL	MESSAGE_VECTOR+4			0232
0C	AE	40	A2	9E	0018B		MOVAB	64(R2), MESSAGE_VECTOR+8			0233
10	AE	14	A2	9A	00190		MOVZBL	20(MSG), MESSAGE_VECTOR+12			0237
14	AE	15	A2	9E	00195		MOVAB	21(R2), MESSAGE_VECTOR+16			0238
18	AE	10	A2	3C	0019A		MOVZWL	16(MSG), MESSAGE_VECTOR+20			0239
1C	AE	0C	A2	D0	0019F		MOVL	12(MSG), MESSAGE_VECTOR+24			0240
20	AE	08	A2	D0	001A4		MOVL	8(MSG), MESSAGE_VECTOR+28			0241
24	AE	30	A2	9A	001A9		MOVZBL	48(MSG), MESSAGE_VECTOR+32			0245
28	AE	31	A2	9E	001AE		MOVAB	49(R2), MESSAGE_VECTOR+36			0246
2C	AE	2C	A2	3C	001B3		MOVZWL	44(MSG), MESSAGE_VECTOR+40			0247
30	AE	28	A2	D0	001B8		MOVL	40(MSG), MESSAGE_VECTOR+44			0248
34	AE	24	A2	D0	001BD		MOVL	36(MSG), MESSAGE_VECTOR+48			0249
		04	AE	9F	001C2		PUSHAB	MESSAGE_VECTOR			0253
			53	DD	001C5		PUSHL	R3			
0000G	CF		02	FB	001C7		CALLS	#2, FORMAT MESSAGE			
	0C	05	A2	E9	001CC		BLBC	5(MSG), 45\$			0257

			53	DD	001D0		PUSHL	R3	:	0259
			06	DD	001D2		PUSHL	#6	:	
	7E		01	CE	001D4		MNEGL	#1, -(SP)	:	
	0000G		03	FB	001D7		CALLS	#3, CLUSMSG_RQCB SEND	:	
		00000000G	8F	D0	001DC	45\$:	MOVL	#MAX_SCOPE, -OCD_INDEX	:	0264
	00000000G		52	D1	001E3	46\$:	CMPL	OCD_INDEX, #MIN_SCOPE	:	0265
			35	19	001EA		BLSS	49\$:	
50			01	78	001EC		ASHL	#1, OCD_INDEX, R0	:	0270
		0000GCF	40	D0	001F0		MOVL	OCD_VECTOR-4[R0], OCD_COUNT	:	
50			01	78	001F6		ASHL	#1, OCD_INDEX, R0	:	0271
		0000GCF	40	D0	001FA		MOVL	OCD_VECTOR-8[R0], OCD	:	
			55	D5	00200	47\$:	TSTL	OCD_COUNT	:	0272
			19	15	00202		BLEQ	48\$:	
	24	A3	54	D0	00204		MOVL	OCD, 36(R3)	:	0278
			53	DD	00208		PUSHL	R3	:	0279
	0000G	CF	01	FB	0020A		CALLS	#1, LOG_MESSAGE	:	
			53	DD	0020F		PUSHL	R3	:	0280
	0000G	CF	01	FB	00211		CALLS	#1, NOTIFY_LISTED_OPERATORS	:	
			55	D7	00216		DECL	OCD_COUNT	:	0281
			64	D0	00218		MOVL	(OCD), OCD	:	0282
			E3	11	0021B		BRB	47\$:	0272
			52	D7	0021D	48\$:	DECL	OCD_INDEX	:	0284
			C2	11	0021F		BRB	46\$:	0265
			53	DD	00221	49\$:	PUSHL	R3	:	0289
	0000G	CF	01	FB	00223		CALLS	#1, DEALLOCATE_RQCB	:	
			04	00228			RET		:	0292

: Routine Size: 553 bytes, Routine Base: \$CODE\$ + 0000

: 295 0293 1 END
: 296 0294 0 ELUDOM

. End of CLUMBX

PSECT SUMMARY

Name	Bytes	Attributes
\$SPLITS	72	NOVEC,NOWR', RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$CODES	553	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

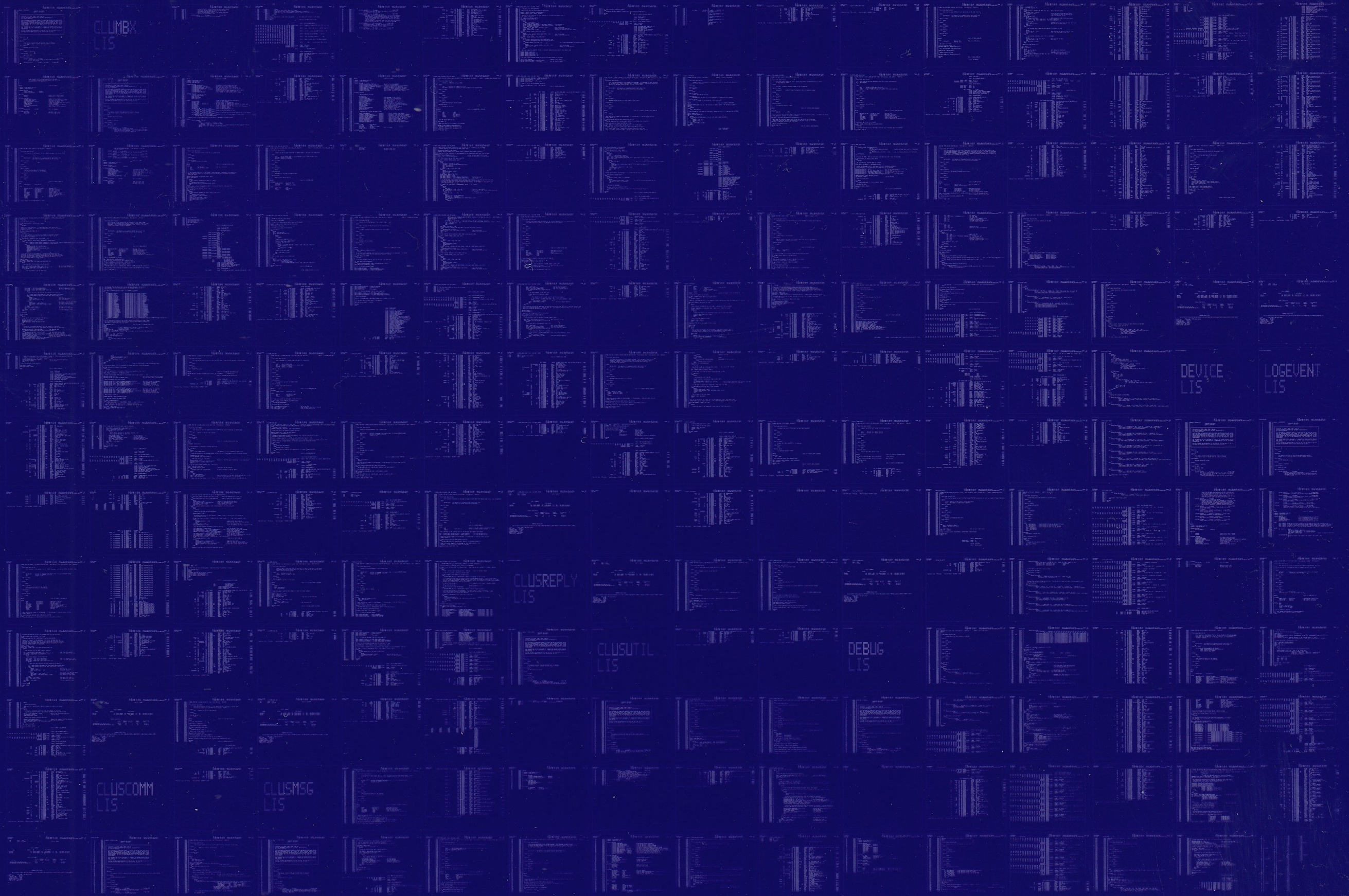
Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	29	0	1000	00:01.9
_\$255\$DUA28:[OPCOM.OBJ]OPCOMLIB.L32;1	633	44	6	43	00:00.8

COMMAND QUALIFIERS

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

: Size: 553 code + 72 data bytes
: Run Time: 00:11.6
: Elapsed Time: 00:42.3
: Lines/CPU Min: 1520
: Lexemes/CPU-Min: 14591
: Memory Used: 139 pages
: Compilation Complete



DEVICE LIS
LOGEVENT LIS

CLUSREPLY LIS

CLUSUTIL LIS

DEBUG LIS

CLUSCOMM LIS

CLUSMSG LIS