


```

1 0001 0 MODULE OPC$CLUSREPLY (
2 0002 0
3 0003 0     LANGUAGE (BLISS32),
4 0004 0     IDENT = 'V04-000'
5 0005 0 ) =
6 0006 0 *****
7 0007 0 *
8 0008 0 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 0 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 0 *  ALL RIGHTS RESERVED.
11 0011 0 *
12 0012 0 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 0 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 0 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 0 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 0 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 0 *  TRANSFERRED.
18 0018 0 *
19 0019 0 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 0 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 0 *  CORPORATION.
22 0022 0 *
23 0023 0 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 0 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 0 *
26 0026 0 *
27 0027 0 *****
28 0028 0
29 0029 0 **
30 0030 0 FACILITY:
31 0031 0
32 0032 0     OPCOM
33 0033 0
34 0034 0 ABSTRACT:
35 0035 0
36 0036 0     This module contains the specialized logic to service
37 0037 0     a particular type of request sent by a user to OPCOM.
38 0038 0
39 0039 0 Environment:
40 0040 0
41 0041 0     VAX/VMS operating system.
42 0042 0
43 0043 0 Author:
44 0044 0
45 0045 0     CW Hobbs
46 0046 0
47 0047 0 Creation date:
48 0048 0
49 0049 0     16-JUL-1983
50 0050 0
51 0051 0 Revision history:
52 0052 0
53 0053 0     V03-003 CWH3169     CW Hobbs     5-May-1984
54 0054 0     Second pass for cluster-wide OPCOM:
55 0055 0     - Change to use the queued brkthru mechanism. Instead of
56 0056 0     performing a brkthru here, call REPLYBRD_BRKTHRU_QUEUE.
57 0057 0

```

OPCSCLUSREPLY
V04-000

L 7
16-Sep-1984 01:23:30 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:50:40 [OPCOM.SRC]CLUSREPLY.B32;1

Page 2
(1)

: 58 0058 0 !
: 59 0059 0 !
: 60 0060 0 !
: 61 0061 0 !
: 62 0062 0 !--

V03-002 CWH3002 CW Hobbs 16-Sep-1983
Improve \$BRKTHRU timeouts, use VM jacket routines

```

: 64      0063 1 BEGIN                                ! Start of CLUSREPLY
: 65      0064 1
: 66      0065 1 LIBRARY 'SYSS$LIBRARY:LIB.L32';
: 67      0066 1 LIBRARY 'LIBS:OPCOMLIB';
: 68      0067 1
: 69      0068 1 FORWARD ROUTINE
: 70      0069 1     CLUSREPLY_RPYBRD_HANDLER      : NOVALUE,
: 71      0070 1     CLUSREPLY_RPYBRD_LOCAL_HANDLER : NOVALUE,
: 72      0071 1     CLUSREPLY_RPYNOT_HANDLER      : NOVALUE;
: 73      0072 1
: 74      0073 1 EXTERNAL ROUTINE
: 75      0074 1     CLUSCOMM_SEND,
: 76      0075 1     DUMP_LOG_FILE,
: 77      0076 1     REPLYBRD_BRKTHRU_QUEUE        : NOVALUE,
: 78      0077 1     REPLYBRD_FORMAT,
: 79      0078 1     REPLYBRD_IO;
: 80      0079 1
: 81      0080 1 EXTERNAL
: 82      0081 1     LCL_NODENAME      : $bblock,
: 83      0082 1     NOD_HEAD         : VECTOR [2, LONG],
: 84      0083 1     GLOBAL_STATUS    : BITVECTOR;

```

```

86 0084 1 GLOBAL ROUTINE CLUSREPLY_RPYBRD_HANDLER (BUFFER_DESC : $ref_bblock, MSG : $ref_bblock, LEN) : NOVALUE =
87 0085 1
88 0086 1 :++
89 0087 1 : Functional description:
90 0088 1
91 0089 1 : This routine processes a CLM_RPYBRD message. This message is sent to this
92 0090 1 : OPCOM by an OPCOM process on another node in response to a REPLY /ALL, REPLY /USER
93 0091 1 : or REPLY /TERMINAL command. The module formats the message and broadcasts it
94 0092 1 : to the local terminals.
95 0093 1
96 0094 1 : Input:
97 0095 1
98 0096 1 : BUFFER_DESC : The address of a quadword buffer descriptor that
99 0097 1 : describes the buffer containing the message.
100 0098 1
101 0099 1 : Implicit Input:
102 0100 1
103 0101 1 : None.
104 0102 1
105 0103 1 : Output:
106 0104 1
107 0105 1 : None.
108 0106 1
109 0107 1 : Implicit output:
110 0108 1
111 0109 1 : Messages might be sent to terminals.
112 0110 1
113 0111 1 : Side effects:
114 0112 1
115 0113 1 : None.
116 0114 1
117 0115 1 : Routine value:
118 0116 1
119 0117 1 : None.
120 0118 1 :--
121 0119 1
122 0120 2 BEGIN ! Start of CLUSREPLY_RPYBRD_HANDLER
123 0121 2
124 0122 2 LOCAL
125 0123 2 status;
126 0124 2
127 0125 2
128 0126 2 : Check the version number of the message. If the message is from any other version,
129 0127 2 : simply ignore it.
130 0128 2
131 0129 2 IF .msg [clm_b_ds_version] NEQ rpybrd_k_ds_version
132 0130 2 THEN
133 0131 2 RETURN dump_log_file (.buffer_desc, %ASCII 'CLM_RPYBRD mismatch');
134 0132 2
135 0133 2 : Format and broadcast the message
136 0134 2
137 0135 2 IF replybrd_format (.msg, lcl_nodename)
138 0136 2 THEN
139 0137 2 replybrd_io (.msg, lcl_nodename);
140 0138 2
141 0139 2
142 0140 2 : Release any message

```

```

: 143      0141 2  :
: 144      0142 2  IF .msg [rpybrd_a_format_ptr] NEQ 0
: 145      0143 2  THEN
: 146      0144 2  OPC$FREE_VM (msg [rpybrd_l_format_len], msg [rpybrd_a_format_ptr]);
: 147      0145 2
: 148      0146 2 RETURN;
: 149      0147 1 END;

```

! End of CLUSREPLY_RPYBRD_HANDLER

```

73 69 6D 20 44 52 42 59 50 52 5F 5F 4D 4C 43 0000 P.AAB: .TITLE OPC$CLUSREPLY
68 63 74 61 6D 0000F .IDENT \V04-000\
010E0014 00014 P.AAA: .PSECT $SPLITS,NOWRT,NOEXE,2
00000000' 00018 .ASCII \CLM_RPYBRD mismatch\
                                .LONG 17694740
                                .ADDRESS P.AAB
                                .EXTRN CLUSCOMM_SEND, DUMP_LOG_FILE
                                .EXTRN REPLYBRD_BRKTHRU_QUEUE
                                .EXTRN REPLYBRD_FORMAT
                                .EXTRN REPLYBRD_IO, LCL_NODENAME
                                .EXTRN NOD_HEAD, GLOBAL_STATUS
                                .EXTRN OPC$FREE_VM
                                .PSECT $CODE$,NOWRT,2
                                .ENTRY CLUSREPLY_RPYBRD_HANDLER, Save R2
                                MOVL MSG, R2
                                CMPB 2(R2), #6
                                BEQL 1$
                                PUSHAB P.AAA
                                PUSHL BUFFER_DESC
                                CALLS #2, DUMP_LOG_FILE
                                RET
                                PUSHAB LCL_NODENAME
                                PUSHL R2
                                CALLS #2, REPLYBRD_FORMAT
                                BLBC R0, 2$
                                PUSHAB LCL_NODENAME
                                PUSHL R2
                                CALLS #2, REPLYBRD_IO
                                TSTL 44(R2)
                                BEQL 3$
                                PUSHAB 44(R2)
                                PUSHAB 40(R2)
                                CALLS #2, OPC$FREE_VM
                                RET

```

; Routine Size: 67 bytes, Routine Base: \$CODE\$ + 0000

```

0148 1 GLOBAL ROUTINE CLUSREPLY_RPYBRD_LOCAL_HANDLER (BUFFER_DESC : $ref_bblock, MSG : $ref_bblock, LEN) : NOVALUE
0149 1
0150 1 ++
0151 1 Functional description:
0152 1
0153 1 This routine processes a CLM_RPYBRD_LOCAL message. This message is sent to OPCOM
0154 1 by the REPLY /ALL, REPLY /USER and REPLY /TERMINAL command. The module determines
0155 1 whether the message should be sent to other cluster nodes.
0156 1
0157 1 Input:
0158 1
0159 1 BUFFER_DESC : The address of a quadword buffer descriptor that
0160 1 describes the buffer containing the message.
0161 1
0162 1 Implicit Input:
0163 1
0164 1 None.
0165 1
0166 1 Output:
0167 1
0168 1 None.
0169 1
0170 1 Implicit output:
0171 1
0172 1 Messages might be sent to other nodes in the cluster.
0173 1
0174 1 Side effects:
0175 1
0176 1 None.
0177 1
0178 1 Routine value:
0179 1
0180 1 None.
0181 1 --
0182 1
0183 2 BEGIN ! Start of CLUSREPLY_RPYBRD_LOCAL_HANDLER
0184 2
0185 2 LOCAL
0186 2 status;
0187 2
0188 2
0189 2 Check the version number of the message. If the message is from any other version,
0190 2 simply ignore it.
0191 2
0192 2 IF .msg [clm_b_ds_version] NEQ rpybrd_k_ds_version
0193 2 THEN
0194 2 RETURN dump_log_file (.buffer_desc, %ASCID 'CLM_RPYBRD_LOCAL mismatch');
0195 2
0196 2 Make sure that the sender has OPER privilege
0197 2
0198 2 IF NOT .$bblock [(.buffer_desc [dsc$a_pointer] + 4), prv$v_oper]
0199 2 THEN
0200 2 RETURN dump_log_file (.buffer_desc, %ASCID 'No OPER privilege');
0201 2
0202 2 Well, looks good so far. Change the secondary operation code, and
0203 2 send it to other interested nodes. We don't have to worry about errors.
0204 2

```



```

: 208      0205 2 msg [clm_b_clm_code] = clm_rpybrd;
: 209      0206 2 IF .msg [rpybrd_v_broad_remoteall]
: 210      0207 2 THEN
: 211      0208 2     cluscomm_send (-1, .len, .msg)
: 212      0209 2 ELSE IF .msg [rpybrd_v_broad_remotelst]
: 213      0210 2 THEN
: 214      0211 2     BEGIN
: 215      0212 2     REGISTER
: 216      0213 2     mlen,
: 217      0214 2     mptr : REF VECTOR [, BYTE];
: 218      0215 2     mlen = .msg [rpybrd_w_targ_node_len];
: 219      0216 2     mptr = .msg + .msg [rpybrd_w_targ_node_off]; ! Add offset to start of node items
: 220      0217 2     WHILE .mlen GTR 0
: 221      0218 2     DO
: 222      0219 2     BEGIN
: 223      0220 2     IF .mptr [0] NEQ 5 ! If len <> 5, it isn't a csid and we are in trouble
: 224      0221 2     THEN
: 225      0222 2     BEGIN
: 226      0223 2     dump_log_file (.buffer_desc, %ASCII 'CSID length not equal to 5');
: 227      0224 2     EXIT[OOP];
: 228      0225 2     END;
: 229      0226 2     cluscomm_send (.mptr [1]), .len, .msg);
: 230      0227 2     mlen = .mlen - 5;
: 231      0228 2     mptr = .mptr + 5;
: 232      0229 2     END;
: 233      0230 2     END;
: 234      0231 2     ;
: 235      0232 2     ; Format and broadcast the message if it is destined for the local node
: 236      0233 2     ;
: 237      0234 2 IF .msg [rpybrd_v_broad_local]
: 238      0235 2 THEN
: 239      0236 2 BEGIN
: 240      0237 2 IF replybrd_format (.msg, lcl_nodename)
: 241      0238 2 THEN
: 242      0239 2 BEGIN
: 243      0240 2 msg [rpybrd_v_local_node] = true; ! Let the IO routine know where we are
: 244      0241 2 replybrd_io (.msg, [cl_nodename]);
: 245      0242 2 END;
: 246      0243 2 ;
: 247      0244 2 ; Release any message
: 248      0245 2 ;
: 249      0246 2 ;
: 250      0247 2 IF .msg [rpybrd_a_format_ptr] NEQ 0
: 251      0248 2 THEN
: 252      0249 2 OPCSFREE_VM (msg [rpybrd_l_format_len], msg [rpybrd_a_format_ptr]);
: 253      0250 2 END;
: 254      0251 2 ;
: 255      0252 2 RETURN;
: 256      0253 1 END; ! End of CLUSREPLY_RPYBRD_LOCAL_HANDLER

```

.PSECT \$SPLITS,NOWRT,NOEXE,2

```

43 4F 4C 5F 44 52 42 59 50 52 5F 5F 4D 4C 43 0001C P.AAD: .ASCII \CLM__RPYBRD_LOCAL mismatch\<<0><0>
00 00 68 63 74 61 6D 75 69 6D 20 4C 41 0002B
010E001A 0003B P.AAC: .LONG 17694746

```


0000G	CF		54	DD	0008A		PUSHL	R4		
	OF		02	FB	0008C		CALLS	#2, REPLYBRD_FORMAT		
OD	A4		50	E9	00091		BLBC	R0, 8\$		
			10	88	00094		BISB2	#16, 13(R4)		0240
		0000G	CF	9F	00098		PUSHAB	LCL_NODENAME		0241
			54	DD	0009C		PUSHL	R4		
0000G	CF		02	FB	0009E		CALLS	#2, REPLYBRD_IO		
		2C	A4	D5	000A3	8\$:	TSTL	44(R4)		0247
			0B	13	000A6		BEQL	9\$		
		2C	A4	9F	000A8		PUSHAB	44(R4)		0249
		28	A4	9F	000AB		PUSHAB	40(R4)		
0000G	CF		02	FB	000AE		CALLS	#2, OPC\$FREE_VM		
			04	000B3	9\$:		RET			0253

; Routine Size: 180 bytes. Routine Base: \$CODE\$ + 0043

```

258 0254 1 GLOBAL ROUTINE CLUSREPLY_RPYNOT_HANDLER (BUFFER_DESC : $ref_bblock, MSG : $ref_bblock, LEN) : NOVALUE =
259 0255 1
260 0256 1 !++
261 0257 1 | Functional description:
262 0258 1 |
263 0259 1 |     This routine processes a CLM_RPYNOT message. This message is sent to this
264 0260 1 |     OPCOM by an OPCOM process on another node in response to a REPLY /NOTIFY
265 0261 1 |     /TERMINAL command. The module formats the message and broadcasts it
266 0262 1 |     to the specified terminal.
267 0263 1 |
268 0264 1 | Input:
269 0265 1 |
270 0266 1 |     BUFFER_DESC : The address of a quadword buffer descriptor that
271 0267 1 |     describes the buffer containing the message.
272 0268 1 |
273 0269 1 | Implicit Input:
274 0270 1 |
275 0271 1 |     None.
276 0272 1 |
277 0273 1 | Output:
278 0274 1 |
279 0275 1 |     None.
280 0276 1 |
281 0277 1 | Implicit output:
282 0278 1 |
283 0279 1 |     Messages should be sent to a terminal.
284 0280 1 |
285 0281 1 | Side effects:
286 0282 1 |
287 0283 1 |     None.
288 0284 1 |
289 0285 1 | Routine value:
290 0286 1 |
291 0287 1 |     None.
292 0288 1 | --
293 0289 1 |
294 0290 2 BEGIN                                     ! Start of CLUSREPLY_RPYNOT_HANDLER
295 0291 2
296 0292 2 LOCAL
297 0293 2     iosh : VECTOR [4, WORD],
298 0294 2     msgdsc : VECTOR [2, LONG],
299 0295 2     trmdsc : VECTOR [2, LONG],
300 0296 2     ptr,
301 0297 2     status;
302 0298 2 |
303 0299 2 |
304 0300 2 | Check the version number of the message. If the message is from any other version,
305 0301 2 | simply ignore it.
306 0302 2 |
307 0303 2 IF .msg [clm_b_ds_version] NEQ rpynot_k_ds_version
308 0304 2 THEN
309 0305 2     RETURN dump_log_file (.buffer_desc, %ASCII 'CLM_RPYNOT mismatch');
310 0306 2 |
311 0307 2 | Make descriptors for the two message components
312 0308 2 |
313 0309 2 ptr = msg [rpynot_t_text];
314 0310 2 trmdsc [1] = .ptr;

```

```

: 315      0311 2 ptr = .ptr + (trmdsc [0] = .msg [rpynot_w_term_len]);
: 316      0312 2 msgdsc [1] = .ptr;
: 317      0313 2 msgdsc [0] = .msg [rpynot_w_message_len];
: 318      0314 2
: 319      0315 2     Queue up to broadcast the message
: 320      0316 2
: 321      0317 2 replybrd_brkthru_queue (msgdsc,           ! text to send
: 322      0318 2     trmdsc,                             ! target name
: 323      0319 2     brk$c_device,                       ! type of target
: 324      0320 2     32,                                 ! carriage control
: 325      0321 2     0,                                 ! brkthru flags
: 326      0322 2     brk$c_general,                     ! request id
: 327      0323 2     0,0,0,0);                          ! no completion routine or parameters
: 328      0324 2
: 329      0325 2 RETURN;
: 330      0326 1 END;

```

! End of CLUSREPLY_RPYNOT_HANDLER

```

.PSECT $SPLITS$,NOWRT,NOEXE,2
73 69 6D 20 54 4F 4E 59 50 52 5F 5F 4D 4C 43 00080 P.AAJ: .ASCII \CLM__RPYNOT mismatch\
68 63 74 61 6D 0008F
010E0014 00094 P.AAI: .LONG 17694740
00000000' 00098 .ADDRESS P.AAJ

```

```

.PSECT $CODE$,NOWRT,2
0004 00000 .ENTRY CLUSREPLY_RPYNOT_HANDLER, Save R2
5E 18 C2 00002 0004 00000
52 08 AC D0 00005 0004 00000
02 02 A2 91 00009 0004 00000
0D 13 0000D 0004 00000
0000' CF 9F 0000F 0004 00000
04 AC DD 00013 0004 00000
0000G CF 02 FB 00016 0004 00000
50 10 A2 9E 0001C 1$: 0004 00000
04 AE 50 D0 00020 0004 00000
51 0C A2 3C 00024 0004 00000
6E 51 D0 00028 0004 00000
50 51 C0 0002B 0004 00000
0C AE 50 D0 0002E 0004 00000
08 AE 0E A2 3C 00032 0004 00000
7E 7C 00037 0004 00000
7E 7C 00039 0004 00000
7E 7C 0003B 0004 00000
7E 20 7D 0003D 0004 00000
01 DD 00040 0004 00000
24 AE 9F 00042 0004 00000
30 AE 9F 00045 0004 00000
0000G CF 0B FB 00048 0004 00000
04 0004D 0004 00000

```

; Routine Size: 78 bytes. Routine Base: \$CODE\$ + 00F7

OPCSCLUSREPLY
V04-000

I 8
16-Sep-1984 01:23:30
14-Sep-1984 12:50:40

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

Page 12
(5)

0
V

: 332 0327 1 END
: 333 0328 0 ELUDOM

! End of module

PSECT SUMMARY

Name	Bytes	Attributes
\$PLITS	156	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$CODE\$	325	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

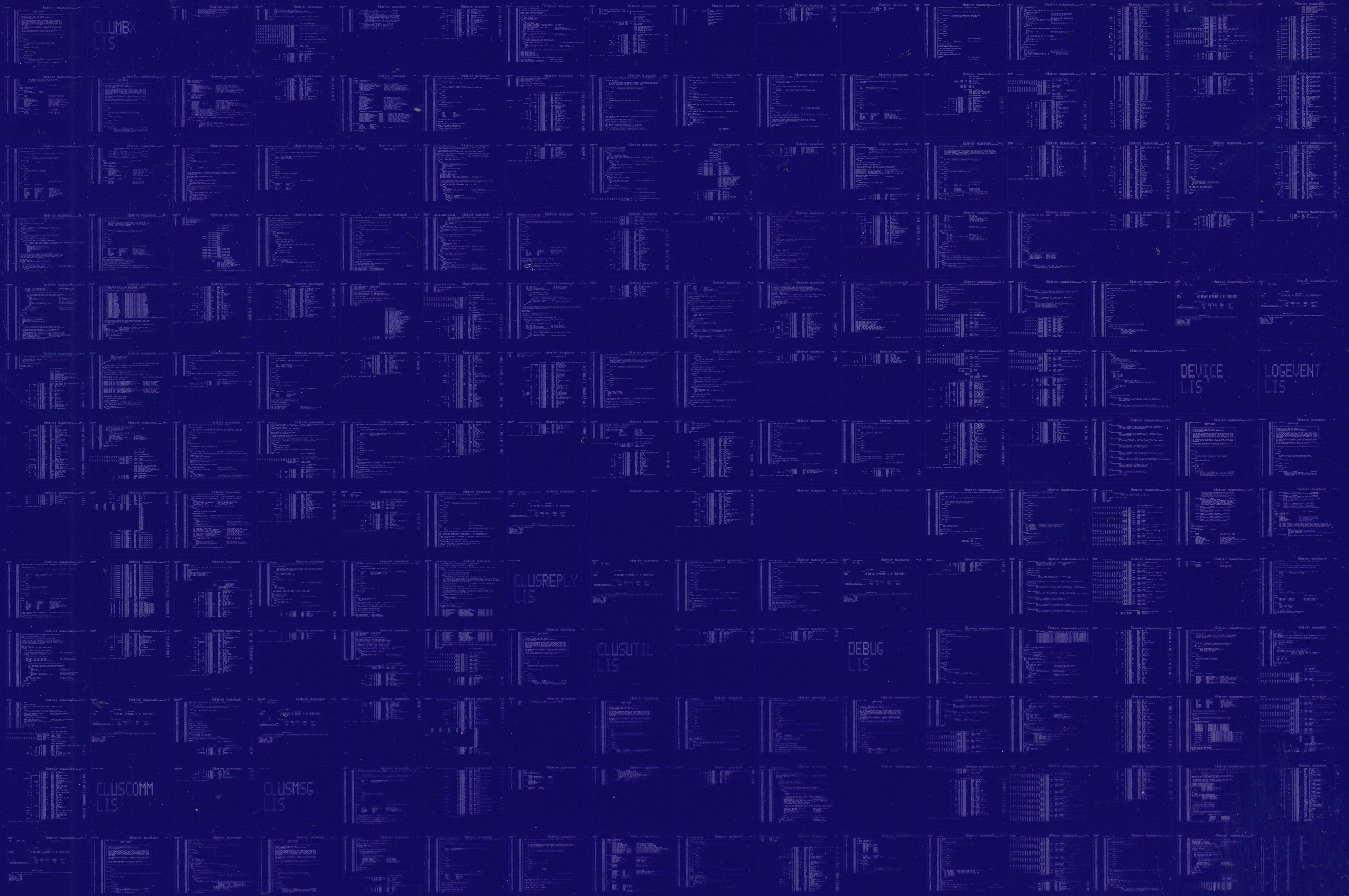
Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	5	0	1000	00:01.8
_\$255\$DUA28:[OPCOM.OBJ]OPCOMLIB.L32;1	633	19	3	43	00:00.8

COMMAND QUALIFIERS

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

: Size: 325 code + 156 data bytes
: Run Time: 00:09.5
: Elapsed Time: 00:38.2
: Lines/CPU Min: 2071
: Lexemes/CPU-Min: 10560
: Memory Used: 97 pages
: Compilation Complete



DEVICE LIS
LOGEVENT LIS

CLUSREPLY LIS

CLUSUTIL LIS

DEBUG LIS

CLUSCOMM LIS

CLUSMSG LIS