



```

EEEEEEEEEE DDDDDDDD IIIIII TTTTTTTTTT
EEEEEEEEEE DDDDDDDD IIIIII TTTTTTTTTT
EE          DD          DD          II          TT
EE          DD          DD          II          TT
EE          DD          DD          II          TT
EE          DD          DD          II          TT
EEEEEEEEEE DD          DD          II          TT
EEEEEEEEEE DD          DD          II          TT
EE          DD          DD          II          TT
EE          DD          DD          II          TT
EE          DD          DD          II          TT
EE          DD          DD          II          TT
EEEEEEEEEE DDDDDDDD IIIIII TTTTTTTTTT
EEEEEEEEEE DDDDDDDD IIIIII TTTTTTTTTT

```

```

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
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```



```

1 0001 0 MODULE MAIL$EDIT( %TITLE 'EDIT OR ENTER A MESSAGE'
2 0002 0 IDENT = 'V04-000' ) =
3 0003 1 BEGIN
4 0004 1
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 * ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 * TRANSFERRED. *
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 * CORPORATION. *
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 **
31 0031 1 FACILITY: VAX/VMS MAIL UTILITY
32 0032 1
33 0033 1 ABSTRACT: Edit or just enter a mail message.
34 0034 1
35 0035 1 ENVIRONMENT: NATIVE/USER MODE
36 0036 1
37 0037 1 AUTHOR: LEN KAWELL, CREATION DATE: 25-NOV-1979
38 0038 1
39 0039 1 MODIFIED BY:
40 0040 1
41 0041 1 V03-016 ROP0023 Robert Posniak 16-JUL-1984
42 0042 1 Change references of XYZ to TPU.
43 0043 1
44 0044 1 V03-015 ROP0007 Robert Posniak 13-JUN-1984
45 0045 1 Check validity of input file spec before
46 0046 1 entering editor. Check for illegal escape
47 0047 1 sequence when entering message.
48 0048 1
49 0049 1 V03-014 BLO280 Benn Schreiber 4-MAR-1984
50 0050 1 Handle terminator not seen and data overrun differently
51 0051 1 since data may be returned.
52 0052 1
53 0053 1 V03-013 BLS0272 Benn Schreiber 25-FEB-1984
54 0054 1 Use LIB$FIND_IMAGE_SYMBOL
55 0055 1
56 0056 1 V03-012 BLS0263 Benn Schreiber 5-FEB-1984
57 0057 1 If not using callable EDT, then copy message to temp file

```

```
58 0058 1 before calling editor.
59 0059 1
60 0060 1 V03-011 BLS0255 Benn Schreiber 28-Dec-1983
61 0061 1 Use global flags. Fix error handling in MAIL$ENTER. Make
62 0062 1 /edit work with sequential mail files.
63 0063 1
64 0064 1 V03-010 BLS0250 Benn Schreiber 8-Dec-1983
65 0065 1 Correct condition handling to not signal error if EDTINI
66 0066 1 not found. Signal error if not callable editor for READ/EDIT
67 0067 1 and EDIT commands.
68 0068 1
69 0069 1 V03-009 BLS0246 Benn Schreiber 28-Nov-1983
70 0070 1 Add EDIT/EXTRACT, support for reading messages.
71 0071 1
72 0072 1 V03-008 BLS0226 Benn Schreiber 27-Jun-1983
73 0073 1 Correct cli$ calls for edit command so that /noxxx qualifiers
74 0074 1 work as advertised
75 0075 1
76 0076 1 V03-007 BLS0222 Benn Schreiber 16-May-1983
77 0077 1 use SMG$READ_STRING rather than LIB$GET_INPUT
78 0078 1
79 0079 1 V03-006 BLS0220 Benn Schreiber 30-Apr-1983
80 0080 1 Make /READ imply /NOJOURNAL
81 0081 1
82 0082 1 V03-005 BLS0211 Benn Schreiber 13-Mar-1983
83 0083 1 Support EDIT command. Load EDTSHR on the fly. In
84 0084 1 MAIL$ENTER, use MAIL's main CTRL/C handler.
85 0085 1
86 0086 1 V03-004 BLS0200 Benn Schreiber 22-Dec-1982
87 0087 1 Remove need for MAIEDTINI by passing SET NOSUMMARY line
88 0088 1 to EDT directly.
89 0089 1
90 0090 1 V03-003 BLS0199 Benn Schreiber 16-Dec-1982
91 0091 1 Use SYSS$SYSTEM:MAIEDTINI.EDT to suppress output file summary.
92 0092 1 Check if file exists and return SENDABORT if not
93 0093 1
94 0094 1 V03-002 BLS0196 Benn Schreiber 22-Nov-1982
95 0095 1 Turn on EDT journalling
96 0096 1
97 0097 1 V03-001 BLS0176 Benn Schreiber 18-Jun-1982
98 0098 1 Add callable EDT interface
99 0099 1
100 0100 1 --
```

```

102 0101 1 |
103 0102 1 | TABLE OF CONTENTS:
104 0103 1 |
105 0104 1 |
106 0105 1 |
107 0106 1 | INCLUDE FILES:
108 0107 1 |
109 0108 1 |     LIBRARY 'SYSS$LIBRARY:STARLET';
110 0109 1 |     REQUIRE 'SRCS:MAILREQ';
111 0255 1 |     LIBRARY 'LIBS:MAILDEF';
112 0256 1 |
113 0257 1 | MACROS:
114 0258 1 |
115 0259 1 |
116 0260 1 |
117 0261 1 | EQUATED SYMBOLS:
118 0262 1 |
119 0263 1 |
120 0264 1 |
121 0265 1 | OWN STORAGE:
122 0266 1 |
123 0267 1 | OWN
124 0268 1 |     COMNOSUM: BYTE,           ! Need to output SET NOSUM
125 0269 1 |     COMNOFILE: BYTE,         ! User's EDTINI not found
126 0270 1 |     COMFILCNT: BYTE,         ! Count fileio calls for init
127 0271 1 |     INPUT_DESC : DYN_DESC_DECL,
128 0272 1 |     COMMAND_DESC : DYN_DESC_DECL,
129 0273 1 |     JOURNAL_DESC : DYN_DESC_DECL,
130 0274 1 |     OUTPUT_DESC : DYN_DESC_DECL;
131 0275 1 |
132 0276 1 | BIND
133 0277 1 |     DYN_DESC = JOURNAL_DESC : $BLOCK,      !Used by MAIL$LOAD_EDT
134 0278 1 |     DYN_DESC_2 = COMMAND_DESC : $BLOCK,    ! ditto
135 0279 1 |     NL_DESC = $DESCRIPTOR('NL:'),
136 0280 1 |     NO_FILE_DESC = $DESCRIPTOR(''),
137 0281 1 |     SYSIN_DESC = $DESCRIPTOR('SYSS$INPUT'),
138 0282 1 |     JOU_FILE_DESC = $DESCRIPTOR('SYSS$SCRATCH:MAIL.JOU');
139 0283 1 |
140 0284 1 | EXTERNAL REFERENCES:
141 0285 1 |
142 0286 1 |
143 0287 1 | EXTERNAL LITERAL
144 0288 1 |     CLIS_NEGATED,           ! Explicitly negated qualifier
145 0289 1 |     CLIS_LOCNEG,           ! Locally negated
146 0290 1 |     MAIL$C_TMP_PRO,        ! Temporary file protection
147 0291 1 |     SMGS_EOF;              ! SMG eof value
148 0292 1 |
149 0293 1 | EXTERNAL
150 0294 1 |     MAIL$MYPID,             ! Process pid
151 0295 1 |     MAIL$M_MSG_RSIZ : WORD, ! Seq. msg. size
152 0296 1 |     MAIL$M_MSG_RADR,        ! Seq. msg address
153 0297 1 |     MAIL$M_MSG_STS,         ! Seq. msg status
154 0298 1 |     MAIL$G_FLAGS : $BLOCK,  ! Control flags
155 0299 1 |     MAIL$SPAWN_CMDS,        ! SPAWN/ATTACH cld commands
156 0300 1 |     MAIL$M_SMG_KEYBOARD,    ! SMG keyboard id
157 0301 1 |     MAIL$Q_INPTRAN: BLOCK[,BYTE], ! SYSS$INPUT translation
158 0302 1 |     MAIL$Q_EDITNAM_DESC: BLOCK[,BYTE], ! translation of MAIL$EDIT

```

```

159 0303 1 MAIL$Q_CALEDTNAM : $BBLOCK,      : Name of editor if callable
160 0304 1 MAIL$Q_USERNAME : $BBLOCK;    : User's name
161 0305 1
162 0306 1 EXTERNAL ROUTINE
163 0307 1 CLISDCL_PARSE,                : Parse command line
164 0308 1 CLISDISPATCH,                : and dispatch on it
165 0309 1 CLISGET_VALUE,                : Get qualifier value
166 0310 1 CLISPRESENT,                 : Test qualifier present
167 0311 1 LIB$GET_INPUT,                 : Read from SYSS$INPUT
168 0312 1 LIB$PUT_OUTPUT,                : Write to SYSS$OUTPUT
169 0313 1 LIB$SPAWN,                     : Spawn a subprocess
170 0314 1 MAIL$DECR_NEWMAIL,             : Decrement new mail count
171 0315 1 MAIL$DISABLE_CTRL_C,          : Disable CTRL/C handling
172 0316 1 MAIL$EXTRACT_MESSAGE,         : Copy message text to file
173 0317 1 MAIL$GET_MSG,                  : Read seq. mail record
174 0318 1 MAIL$GET_I_MSG,                : Read isam mail record
175 0319 1 MAIL$GETMSG,                  : Get error message text
176 0320 1 MAIL$LOAD_LBRSHR,              : Load LBRSHR
177 0321 1 MAIL$NEW_MSG,                  : Test if new message
178 0322 1 MAIL$PUT_OUTPUT,
179 0323 1 MAIL$REWIND_I_MSG,             : Rewind isam mail msg
180 0324 1 SMG$READ_STRING,               : Read string
181 0325 1 STR$CONCAT,                    : String concatenation
182 0326 1 STR$COPY_DX,                   : String copy routine
183 0327 1 STR$FREE_DX,                   : Free dynamic string
184 0328 1 UTIL$REPORT_IO_ERROR,          : Report i/o error
185 0329 1 LIB$FIND_IMAGE_SYMBOL;        : Load image, find symbol val
186 0330 1
187 0331 1 : Define local names for shared messages
188 0332 1
189 P 0333 1 $$HR_MSGDEF(MAIL,126,LOCAL,
190 P 0334 1 (READERR,ERROR),
191 P 0335 1 (OPENIN,ERROR),
192 P 0336 1 (OPENOUT,ERROR),
193 P 0337 1 (CLOSEOUT,ERROR),
194 P 0338 1 (FILNOTDEL,WARNING),
195 0339 1 (WRITEERR,WARNING));
196 0340 1
197 0341 1 : Table of symbol names
198 0342 1
199 0343 1 OWN
200 0344 1 XYZ_HANDLER;
201 0345 1
202 M 0346 1 MACRO EDTSYM(EDTNAME) =
203 M 0347 1 PSECT OWN = $OWNS;
204 M 0348 1
205 M 0349 1 OWN %NAME('EDT_',EDTNAME) : LONG;
206 M 0350 1
207 M 0351 1 PSECT OWN = EDT_ADDRESSES (NOWRITE,EXECUTE);
208 M 0352 1
209 M 0353 1 OWN %NAME(EDTNAME,'_ADDR') : INITIAL(%NAME('EDT_',EDTNAME));
210 M 0354 1
211 M 0355 1 PSECT OWN = EDT_NAMES (NOWRITE,EXECUTE);
212 M 0356 1
213 M 0357 1 OWN %NAME(EDTNAME) : VECTOR[2,LONG]
214 M 0358 1 INITIAL(%CHARCOUNT(%NAME('EDT$',EDTNAME)),
215 M 0359 1 UPLIT(%STRING('EDT$',%NAME(EDTNAME))));

```

```
216 M 0360 1  
217 0361 1 UNDECLARE %NAME(EDTNAME, '_ADDR'), %NAME(EDTNAME)%;  
218 0362 1  
219 0363 1 PSECT OWN = EDT_ADDRESSES(NOWRITE, EXECUTE);  
220 0364 1  
221 0365 1 OWN  
222 0366 1 EDT_ADDR_HEAD : VECTOR[0, LONG];  
223 0367 1  
224 0368 1 PSECT OWN = EDT_NAMES(NOWRITE, EXECUTE);  
225 0369 1  
226 0370 1 OWN  
227 0371 1 EDT_NAMES_HEAD : VECTOR[0, LONG];  
228 0372 1  
229 0373 1 EDTSYM(M_RECOVER);  
230 0374 1 EDTSYM(M_NOJOURNAL);  
231 0375 1 EDTSYM(M_NOOUTPUT);  
232 0376 1 EDTSYM(M_NOCOMMAND);  
233 0377 1 EDTSYM(M_NOCREATE);  
234 0378 1 EDTSYM(K_COMMAND_FILE);  
235 0379 1 EDTSYM(K_INPUT_FILE);  
236 0380 1 EDTSYM(K_OPEN_INPUT);  
237 0381 1 EDTSYM(K_GET);  
238 0382 1 EDTSYM(K_CLOSE);  
239 0383 1 EDTSYM(EDIT);  
240 0384 1 EDTSYM(FILEIO);  
241 0385 1  
242 0386 1 PSECT OWN = EDT_ADDRESSES;  
243 0387 1  
244 0388 1 OWN  
245 0389 1 EDT_ADDR_END : LONG INITIAL(0); !Mark end of table  
246 0390 1  
247 0391 1 PSECT OWN = $OWNS;  
248 0392 1  
249 0393 1  
250 0394 1 ! Define names for qualifier name descriptors  
251 0395 1  
252 P 0396 1 SD (  
253 P 0397 1 'EDTSHR',  
254 P 0398 1 'COMMAND',  
255 P 0399 1 'CREATE',  
256 P 0400 1 'INPUT',  
257 P 0401 1 'JOURNAL',  
258 P 0402 1 'OUTPUT',  
259 P 0403 1 'READ',  
260 0404 1 'RECOVER');
```

```

: 262 0405 1 GLOBAL ROUTINE MAIL$LOAD_EDT =
: 263 0406 1 |++
: 264 0407 1 | FUNCTIONAL DESCRIPTION:
: 265 0408 1 |
: 266 0409 1 |         Load EDT if it has not been loaded
: 267 0410 1 |
: 268 0411 1 | --
: 269 0412 2 BEGIN
: 270 0413 2
: 271 0414 2 LOCAL
: 272 0415 2     EDTINDEX;
: 273 0416 2
: 274 0417 2 IF NOT .MAIL$GL_FLAGS[MAIF_V_CALEDT]
: 275 0418 2     THEN RETURN SIGNAL(MAIL[$_NOTCALEDT]);
: 276 0419 2
: 277 0420 2 IF .EDT_EDIT NEQ 0
: 278 0421 2     THEN RETURN TRUE;
: 279 0422 2
: 280 0423 2 |
: 281 0424 2 | Make sure LBRSHR is loaded first
: 282 0425 2 |
: 283 P 0426 2 IF_ERR(MAIL$LOAD_LBRSHR();,
: 284 0427 2     RETURN .STATUS);
: 285 0428 2
: 286 0429 2 IF .MAIL$GL_FLAGS[MAIF_V_EDTEDT]
: 287 0430 3 THEN BEGIN
: 288 0431 3     EDTINDEX = 0;
: 289 0432 3     WHILE .EDT_ADDR_HEAD[.EDTINDEX] NEQ 0
: 290 0433 4     DO BEGIN
: 291 P 0434 4         IF_ERR(LIB$FIND_IMAGE_SYMBOL(SD_EDTSHR,EDT_NAMES_HEAD[.EDTINDEX*2],
: 292 P 0435 4             .EDT_ADDR_HEAD[.EDTINDEX]);,
: 293 0436 4             RETURN .STATUS);
: 294 0437 4         EDTINDEX = .EDTINDEX + 1;
: 295 0438 3     END;
: 296 0439 3
: 297 0440 3 ELSE BEGIN
: 298 0441 3     STR$CONCAT(DYN_DESC,MAIL$Q_CALEDTNAM,$DESCRIPTOR('SHR'));
: 299 0442 3     STR$CONCAT(DYN_DESC_2,MAIL$Q_CALEDTNAM,$DESCRIPTOR('$EDIT'));
: 300 P 0443 3     IF_ERR(LIB$FIND_IMAGE_SYMBOL(DYN_DESC,DYN_DESC_2,EDT_EDIT));,
: 301 0444 3         RETURN .STATUS);
: 302 0445 3
: 303 P 0446 3     IF .MAIL$GL_FLAGS[MAIF_V_EDTXYZ]
: 304 P 0447 3         THEN IF_ERR(LIB$FIND_IMAGE_SYMBOL(DYN_DESC,$DESCRIPTOR('TPUSHHANDLER'),
: 305 0448 3             XYZ_HANDLER));,
: 306 0449 2         RETURN .STATUS);
: 307 0450 2     END;
: 308 0451 2 RETURN TRUE
: 309 0452 1 END;

```

```

.TITLE MAIL$EDIT EDIT OR ENTER A MESSAGE
.IDENT \V04-000\
.PSECT $CODE$,NOWRT,2
3A 4C 4E 5F 00000 P.AAB: .ASCII \_NL:\
      00000004 00004 P.AAA: .LONG 4

```



MAILSEdit  
V04-000

EDIT OR ENTER A MESSAGE

N 12  
16-Sep-1984 00:51:12  
14-Sep-1984 12:42:17

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[MAIL.SRC]EDIT.B32;1

Page 7  
(3)

																			00000000'	00008		.ADDRESS	P.AAB			
																			00000000'	0000C	P.AAD:	.BLKB	0			
																			00000000'	0000C	P.AAC:	.LONG	0			
																			00000000'	00010		.ADDRESS	P.AAD			
						54	55	50	4E	49	24	53	59	53					00000000'	00014	P.AAF:	.ASCII	\SYSS\$INPUT\			
																			00000009	0001D		.BLKB	3			
																			00000009	00020	P.AAE:	.LONG	9			
																			00000000'	00024		.ADDRESS	P.AAF			
49	41	4D	3A	48	43	54	41	52	43	53	24	53	59	53				00000000'	00028	P.AAH:	.ASCII	\SYSS\$SCRATCH:MAIL.JOU\				
																			00000014	00037						
																			00000000'	0003C	P.AAG:	.LONG	??			
00	00	52	45	56	4F	43	45	52	5F	4D	24	54	44	45				00000000'	00040		.ADDRESS	P.AAH				
																			00000000'	00044	P.AAI:	.ASCII	\EDTSM_RECOVER\<0><0><0>			
4C	41	4E	52	55	4F	4A	4F	4E	5F	4D	24	54	44	45				00000000'	00053							
																			00000000'	00054	P.AAJ:	.ASCII	\EDTSM_NOJOURNAL\<0>			
00	54	55	50	54	55	4F	4F	4E	5F	4D	24	54	44	45				00000000'	00063							
																			00000000'	00064	P.AAK:	.ASCII	\EDTSM_NOOUTPUT\<0><0>			
44	4E	41	4D	4D	4F	43	4F	4E	5F	4D	24	54	44	45				00000000'	00073							
																			00000000'	00074	P.AAL:	.ASCII	\EDTSM_NOCOMMAND\<0>			
00	45	54	41	45	52	43	4F	4E	5F	4D	24	54	44	45				00000000'	00083							
																			00000000'	00084	P.AAM:	.ASCII	\EDTSM_NOCREATE\<0><0>			
46	5F	44	4E	41	4D	4D	4F	43	5F	4B	24	54	44	45				00000000'	00093							
																			00000000'	00094	P.AAN:	.ASCII	\EDTSK_COMMAND_FILE\<0><0>			
4C	49	46	5F	54	55	50	4E	49	5F	4B	24	54	44	45				00000000'	00A3							
																			00000000'	00A8	P.AAO:	.ASCII	\EDTSK_INPUT_FILE\			
55	50	4E	49	5F	4E	45	50	4F	5F	4B	24	54	44	45				00000000'	00B7							
																			00000000'	00B8	P.AAP:	.ASCII	\EDTSK_OPEN_INPUT\			
																			00000000'	00C7						
						00	00	00	54	45	47	5F	4B	24	54	44	45		00000000'	00C8	P.AAQ:	.ASCII	\EDTSK_GET\<0><0><0>			
						00	45	53	4F	4C	43	5F	4B	24	54	44	45		00000000'	00D4	P.AAR:	.ASCII	\EDTSK_CLOSE\<0>			
																			00000000'	00E0	P.AAS:	.ASCII	\EDTSEdit\			
																			00000000'	00E8	P.AAT:	.ASCII	\EDT\$FILEIO\<0><0>			
																			00000000'	00F4	P.AAV:	.ASCII	\EDTSHR\			
																			00000000'	00FA		.BLKB	2			
																			00000006	00FC	P.AAU:	.LONG	6			
																			00000000'	00100		.ADDRESS	P.AAV			
																			00000000'	00104	P.AAX:	.ASCII	\COMMAND\			
																			00000000'	0010B		.BLKB	1			
																			00000007	0010C	P.AAW:	.LONG	7			
																			00000000'	00110		.ADDRESS	P.AAX			
																			00000000'	00114	P.AAZ:	.ASCII	\CREATE\			
																			00000000'	0011A		.BLKB	2			
																			00000006	0011C	P.AAY:	.LONG	6			
																			00000000'	00120		.ADDRESS	P.AAZ			
																			00000000'	00124	P.ABB:	.ASCII	\INPUT\			
																			00000000'	00129		.BLKB	3			
																			00000005	0012C	P.ABA:	.LONG	5			
																			00000000'	00130		.ADDRESS	P.ABB			
																			00000000'	00134	P.ABD:	.ASCII	\JOURNAL\			
																			00000000'	0013B		.BLKB	1			
																			00000007	0013C	P.ABC:	.LONG	7			
																			00000000'	00140		.ADDRESS	P.ABD			
																			00000000'	00144	P.ABF:	.ASCII	\OUTPUT\			
																			00000000'	0014A		.BLKB	2			
																			00000006	0014C	P.ABE:	.LONG	6			
																			00000000'	00150		.ADDRESS	P.ABF			
																			00000000'	00154	P.ABH:	.ASCII	\READ\			

.....

M/V

```

00000004 00158 P.ABG: .LONG 4
00000000' 0015C .ADDRESS P.ABH
52 45 56 4F 43 45 52 00160 P.ABJ: .ASCII \RECOVER\
00167 .BLKB 1
00000007 00168 P.ABI: .LONG 7
00000000' 0016C .ADDRESS P.ABJ
52 48 53 00170 P.ABL: .ASCII \SHR\
00173 .BLKB 1
00000003 00174 P.ABK: .LONG 3
00000000' 00178 .ADDRESS P.ABL
54 49 44 45 24 0017C P.ABN: .ASCII \$EDIT\
00181 .BLKB 3
00000005 00184 P.ABM: .LONG 5
00000000' 00188 .ADDRESS P.ABN
52 45 4C 44 4E 41 48 24 55 50 54 0018C P.ABP: .ASCII \TPUSHANDLER\
00197 .BLKB 1
0000000B 00198 P.ABO: .LONG 11
00000000' 0019C .ADDRESS P.ABP

```

.PSECT EDT\_NAMES,NOWRT,2

```

00000 EDT_NAMES HEAD:
00000000' .BLKB 0
0000000D 00000 M_RECOVER:
00000000' 00004 .LONG 13
0000000F 00008 M_NOJOURNAL: .ADDRESS P.AAI
00000000' 0000C .LONG 15
0000000E 00010 M_NOOUTPUT: .ADDRESS P.AAJ
00000000' 00014 .LONG 14
0000000F 00018 M_NOCOMMAND: .ADDRESS P.AAK
00000000' 0001C .LONG 15
0000000E 00020 M_NOCREATE: .ADDRESS P.AAL
00000000' 00024 .LONG 14
00000012 00028 K_COMMAND_FILE: .ADDRESS P.AAM
00000000' 0002C .LONG 18
00000010 00030 K_INPUT_FILE: .ADDRESS P.AAN
00000000' 00034 .LONG 16
00000010 00038 K_OPEN_INPUT: .ADDRESS P.AAO
00000000' 0003C .LONG 16
00000009 00040 K_GET: .ADDRESS P.AAP
00000000' 00044 .LONG 9
0000000B 00048 K_CLOSE: .ADDRESS P.AAQ
00000000' 0004C .LONG 11
00000008 00050 EDIT: .ADDRESS P.AAR
00000000' 00054 .LONG 8
0000000A 00058 FILEIO: .ADDRESS P.AAS
00000000' 0005C .LONG 10
00000000' .ADDRESS P.AAT

```

.PSECT EDT\_ADDRESSES,NOWRT,2

```
00000 EDT_ADDR_HEAD:
      .BLKB 0
00000000' 00000 M_RECOVER_ADDR:
      .ADDRESS EDT_M_RECOVER ;
00000000' 00004 M_NOJOURNAL_ADDR:
      .ADDRESS EDT_M_NOJOURNAL ;
00000000' 00008 M_NOOUTPUT_ADDR:
      .ADDRESS EDT_M_NOOUTPUT ;
00000000' 0000C M_NOCOMMAND_ADDR:
      .ADDRESS EDT_M_NOCOMMAND ;
00000000' 00010 M_NOCREATE_ADDR:
      .ADDRESS EDT_M_NOCREATE ;
00000000' 00014 K_COMMAND_FILE_ADDR:
      .ADDRESS EDT_K_COMMAND_FILE ;
00000000' 00018 K_INPUT_FILE_ADDR:
      .ADDRESS EDT_K_INPUT_FILE ;
00000000' 0001C K_OPEN_INPUT_ADDR:
      .ADDRESS EDT_K_OPEN_INPUT ;
00000000' 00020 K_GET_ADDR:
      .ADDRESS EDT_K_GET ;
00000000' 00024 K_CLOSE_ADDR:
      .ADDRESS EDT_K_CLOSE ;
00000000' 00028 EDIT_ADDR:
      .ADDRESS EDT_EDIT ;
00000000' 0002C FILEIO_ADDR:
      .ADDRESS EDT_FILEIO ;
00000000 00030 EDT_ADDR_END:
      .LONG 0 ;
```

.PSECT \$OWNS,NOEXE,2

```
00000 COMNOSUM:
      .BLKB 1
00001 COMNOFILE:
      .BLKB 1
00002 COMFILCNT:
      .BLKB 1
00003
      .BLKB 1
00# 00004 INPUT_DESC:
      .BYTE 0[2] ;
02 0E 00006
      .BYTE 14, 2 ;
      .BLKB 4 ;
00# 0000C COMMAND_DESC:
      .BYTE 0[2] ;
02 0E 0000E
      .BYTE 14, 2 ;
      .BLKB 4 ;
00# 00014 JOURNAL_DESC:
      .BYTE 0[2] ;
02 0E 00016
      .BYTE 14, 2 ;
      .BLKB 4 ;
00# 0001C OUTPUT_DESC:
      .BYTE 0[2] ;
02 0E 0001E
      .BYTE 14, 2 ;
      .BLKB 4 ;
00020
      .BLKB 4 ;
00024 XYZ_HANDLER:
      .BLKB 4
```

```

00028 EDT_M_RECOVER:
      .BLKB 4
0002C EDT_M_NOJOURNAL:
      .BLKB 4
00030 EDT_M_NOOUTPUT:
      .BLKB 4
00034 EDT_M_NOCOMMAND:
      .BLKB 4
00038 EDT_M_NOCREATE:
      .BLKB 4
0003C EDT_K_COMMAND_FILE:
      .BLKB 4
00040 EDT_K_INPUT FILE:
      .BLKB 4
00044 EDT_K_OPEN INPUT:
      .BLKB 4
00048 EDT_K_GET:
      .BLKB 4
0004C EDT_K_CLOSE:
      .BLKB 4
00050 EDT_EDIT:
      .BLKB 4
00054 EDT_FILEIO:
      .BLKB 4

```

```

DYN_DESC=          JOURNAL_DESC
DYN_DESC_2=        COMMAND_DESC
NL_DESC=           P.AAA
NO_FILE_DESC=      P.AAC
SYSIN_DESC=        P.AAE
JOU_FILE_DESC=     P.AAG
SD_EDTSHR=         P.AAU
SD_COMMAND=        P.AAW
SD_CREATE=         P.AAY
SD_INPUT=          P.ABA
SD_JOURNAL=        P.ABC
SD_OUTPUT=         P.ABE
SD_READ=           P.ABG
SD_RECOVER=        P.ABI
.EXTRN CLIS NEGATED, CLIS LOCNEG
.EXTRN MAIL$C_TMP_PRO, SMGS_EOF
.EXTRN MAIL$M_MYPID, MAIL$W_MSG_RSIZ
.EXTRN MAIL$M_MSG_RADR
.EXTRN MAIL$M_MSG_STS, MAIL$GL_FLAGS
.EXTRN MAIL$SPAWN_CMDS
.EXTRN MAIL$M_SMG_KEYBOARD
.EXTRN MAIL$Q_INPTRAN, MAIL$Q_EDITNAM_DESC
.EXTRN MAIL$Q_CALEDTNAM
.EXTRN MAIL$Q_USERNAME
.EXTRN CLISDCE_PARSE, CLISDISPATCH
.EXTRN CLISGET_VALUE, CLISPRESENT
.EXTRN LIB$GET_INPUT, LIB$PUT_OUTPUT
.EXTRN LIB$SPAWN, MAIL$DECR_NEWMAIL
.EXTRN MAIL$DISABLE_CTRLC
.EXTRN MAIL$EXTRACT_MESSAGE
.EXTRN MAIL$GET_MSG, MAIL$GET_I_MSG
.EXTRN MAIL$GETMSG, MAIL$LOAD_LBRSHR

```

					.EXTRN	MAIL\$NEW MSG, MAIL\$PUT_OJTPUT		
					.EXTRN	MAIL\$REWIND I MSG		
					.EXTRN	SMG\$READ STRING		
					.EXTRN	STR\$CONCAT, STR\$COPY DX		
					.EXTRN	STR\$FREE1 DX, UTIL\$REPORT_IO_ERROR		
					.EXTRN	LIB\$FIND_IMAGE_SYMBOL		
					.PSECT	\$CODE\$,NOWRT,2		
					.ENTRY	MAIL\$LOAD EDT, Save R2,R3,R4,R5,R6,R7,R8		0405
					MOVAB	STR\$CONCAT, R8		
					MOVAB	MAIL\$Q CAL\$DTNAM, R7		
					MOVAB	MAIL\$GL FLAGS, R6		
					MOVAB	LIB\$FIND_IMAGE_SYMBOL, R5		
					MOVAB	SD EDTSHR, R4		
					MOVAB	DYN_DESC, R3		
					TJTB	MAIL\$GL_FLAGS+1		0417
					BLSS	1\$		
					PUSHL	#8290586		0418
					CALLS	#1, LIB\$SIGNAL		
					RET			
					TSTL	EDT_EDIT		0420
					BNEQ	4\$		
					CALLS	#0, MAIL\$LOAD_LBRSHR		0427
					BLBC	STATUS, 5\$		
					BLBC	MAIL\$GL_FLAGS+2, 3\$		0429
					CLRL	EDTINDEX		0431
					MOVL	EDT_ADDR_HEAD[EDTINDEX], R0		0432
					BEQL	4\$		
					PUSHL	R0		0436
					ASHL	#1, EDTINDEX, R0		
					PUSHAL	EDT_NAMES_HEAD[R0]		
					PUSHL	R4		
					CALLS	#3, LIB\$FIND_IMAGE_SYMBOL		
					BLBC	STATUS, 5\$		
					INCL	EDTINDEX		0437
					BRB	2\$		0432
					PUSHAB	P.ABK		0441
					PUSHR	#^M<R3,R7>		
					CALLS	#3, STR\$CONCAT		
					PUSHAB	P.ABM		0442
					PUSHL	R7		
					PUSHAB	DYN_DESC_2		
					CALLS	#3, STR\$CONCAT		
					PUSHAB	EDT_EDIT		0444
					PUSHAB	DYN_DESC_2		
					PUSHL	R3		
					CALLS	#3, LIB\$FIND_IMAGE_SYMBOL		
					BLBC	STATUS, 5\$		
					BBC	#1, MAIL\$GL_FLAGS+2, 4\$		0445
					PUSHAB	XYZ_HANDLER		0448
					PUSHAB	P.ABO		
					PUSHL	R3		
					CALLS	#3, LIB\$FIND_IMAGE_SYMBOL		
					BLBC	STATUS, 5\$		
					MOVL	#1, R0		0451
					RET			0452

MAIL\$EDIT  
V04-000

EDIT OR ENTER A MESSAGE

F 13  
16-Sep-1984 00:51:12  
14-Sep-1984 12:42:17

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[MAIL.SRC]EDIT.B32;1 Page 12  
(3)

; Routine Size: 177 bytes, Routine Base: \$CODE\$ + 01A0

MA  
VO

21

4C

```

311 0453 1 ROUTINE CHECK_MESSAGE_FILE(OUT_DESC) =
312 0454 1 |**
313 0455 1 | FUNCTIONAL DESCRIPTION:
314 0456 1 |
315 0457 1 | This routine determines whether the specified file was successfully
316 0458 1 | created and returns success if it was, and MAIL$_SENDABORT if not
317 0459 1 |
318 0460 1 | INPUTS:
319 0461 1 |
320 0462 1 | OUT_DESC is a descriptor for the file to check
321 0463 1 | --
322 0464 2 BEGIN
323 0465 2
324 0466 2 MAP
325 0467 2 OUT_DESC : REF $BBLOCK;
326 0468 2
327 0469 2 LOCAL
328 0470 2 FAB : $FAB_DECL,
329 0471 2 NAM : $NAM_DECL,
330 0472 2 NAMESS : $BBLOCK[NAM$C_MAXRSS],
331 0473 2 STATUS;
332 0474 2
333 P 0475 2 $FAB_INIT(FAB=FAB,
334 P 0476 2 FNS=.OUT_DESC[DSC$W_LENGTH],
335 P 0477 2 FNA=.OUT_DESC[DSC$A_POINTER],
336 0478 2 NAM=NAM);
337 0479 2
338 P 0480 2 $NAM_INIT(NAM=NAM,
339 P 0481 2 ESS=NAM$C_MAXRSS,
340 P 0482 2 ESA=NAMESS,
341 P 0483 2 RSS=NAM$C_MAXRSS,
342 0484 2 RSA=NAMESS);
343 0485 2 |
344 0486 2 | If the PARSE or SEARCH fails, then return SENDABORT, otherwise return
345 0487 2 | success
346 0488 2 |
347 0489 3 IF NOT $PARSE(FAB=FAB)
348 0490 3 OR NOT $SEARCH(FAB=FAB)
349 0491 2 THEN RETURN MAIL$_SENDABORT
350 0492 2 ELSE RETURN TRUE
351 0493 2
352 0494 1 END;

```

.EXTRN SYS\$PARSE, SYS\$SEARCH

003C 0000 CHECK\_MESSAGE\_FILE:

0050	8F	00	5E	FE50	CE	9E	00002	.WORD	Save R2,R3,R4,R5	: 0453
			6E		00	2C	00007	MOVAB	-432(SP), SP	
				B0	AD		0000E	MOVCS	#0, (SP), #0, #80, \$RMS_PTR	: 0478
			B0	AD	5003	8F	B0	MOVW	#20483, \$RMS_PTR	
			C6	AD		02	90	MOVB	#2, \$RMS_PTR+22	
			CF	AD		02	90	MOVB	#2, \$RMS_PTR+31	
			D8	AD	FF50	CD	9E	MOVAB	NAM, \$RMS_PTR+40	
			50		04	AC	00	MOVL	OUT_DESC, R0	

0060	8F	00	DC E4	AD AD 6E	04	A0 60 00	D0 90 2C	00028 0002D 00031	MOVL MOVB MOVCS	4(R0), \$RMS_PTR+4 (R0), \$RMS_PTR+52 #0, (SP), #0, #96, \$RMS_PTR	:	0484
			FF50 FF52 FF54 FF5A FF5C	CD CD CD CD CD	Ff50 6002	8F 01 6E 01 6E	B0 8E 9E 8E 9E	00038 0003B 00042 00047 0004C 00051	MOVW MNEGB MOVAB MNEGB MOVAB	#24578, \$RMS_PTR #1, \$RMS_PTR+2 NAMESS, \$RMS_PTR+4 #1, \$RMS_PTR+10 NAMESS, \$RMS_PTR+12	:	0489
		00000000G		00 0D	B0	AD 01 50	9F FB E9	00056 00059 00060	PUSHAB CALLS BLBC	FAB #1, SYSSPARSE R0, 1\$	:	0490
		00000000G		00 08 50	B0	AD 01 50	9F FB E8	00063 00066 0006D	PUSHAB CALLS BLBS	FAB #1, SYSSSEARCH R0, 2\$	:	0492
				50 007E805A		8F	D0	00070	1\$: RET	#8290394, R0	:	0494
				50		01	D0	00078	2\$: RET	#1, R0	:	0494
						04	0007B				:	

: Routine Size: 124 bytes, Routine Base: \$CODE\$ + 0251



```

: 354      0495 1 ROUTINE SEND_STRING (DESC) =
: 355      0496 1 |++
: 356      0497 1 | FUNCTIONAL DESCRIPTION:
: 357      0498 1 |
: 358      0499 1 |         Write string to SYS$OUTPUT
: 359      0500 1 |
: 360      0501 1 |--
: 361      0502 2 BEGIN
: 362      0503 2
: 363      0504 2 LIB$PUT_OUTPUT(.DESC);
: 364      0505 2 RETURN 0
: 365      0506 1 END;

```

```

                                0000 00000 SEND_STRING:
                                .WORD   Save nothing
                                PUSHL   DESC
                                CALLS  #1, LIB$PUT_OUTPUT
                                CLRL   R0
                                RET
00000000G 00          04 AC DD 00002
                                01 FB 00005
                                50 D4 0000C
                                04 0000E
: 0495
: 0504
: 0505
: 0506

```

: Routine Size: 15 bytes. Routine Base: \$CODE\$ + 02CD

```

367 0507 1 ROUTINE EDT_HANDLER (SIGARG, MECHARG) =
368 0508 1 |++
369 0509 1 | FUNCTIONAL DESCRIPTION:
370 0510 1 |
371 0511 1 |         Handler to handle EDT signalled errors. Prevents EDT's fatal
372 0512 1 |         errors from exiting MAIL
373 0513 1 | --
374 0514 2 BEGIN
375 0515 2 MAP
376 0516 2     SIGARG : REF $BBLOCK,
377 0517 2     MECHARG : REF $BBLOCK;
378 0518 2
379 0519 2 BUILTIN
380 0520 2     CALLG,AP;
381 0521 2
382 0522 2 BIND
383 0523 2     SIGNAME = SIGARG[CHFSL_SIG_NAME] : $BBLOCK;
384 0524 2
385 0525 2 IF .SIGNAME EQL SSS_UNWIND
386 0526 2     THEN RETURN SSS_CONTINUE;
387 0527 2
388 0528 2 IF .MAIL$GL_FLAGS[MAIF_V_EDTXYZ]
389 0529 2     THEN RETURN CALLG(.AP,.XYZ_HANDLER);
390 0530 2
391 0531 2 IF NOT .SIGNAME
392 0532 2     THEN BEGIN
393 0533 3     SIGARG[CHFSL_SIG_ARGS] = .SIGARG[CHFSL_SIG_ARGS] - 2;
394 0534 3     $PUTMSG(MSGVEC=SIGARG[CHFSL_SIG_ARGS],
395 0535 3         ACTRN = SEND_STRING);
396 0536 3     SIGARG[CHFSL_SIG_ARGS] = .SIGARG[CHFSL_SIG_ARGS] + 2;
397 0537 3     SIGNAME[STSSV_SEVERITY] = STSSK_WARNING;
398 0538 2     END;
399 0539 2
400 0540 2 SETUNWIND();
401 0541 2 RETURN 0
402 0542 1 END;

```

.EXTRN SYSS\$PUTMSG

```

0004 00000 EDT_HANDLER:
52 00000920 04 AC 04 C1 00002 .WORD Save R2
8F 62 D1 00007 ADDL3 #4, SIGARG, R2
04 12 0000E CMPL (R2), #2336
50 01 D0 00010 BNEQ 1$
04 00013 MOVL #1, R0
0B 0000000G 00 01 E1 00014 1$: BBC #1, MAIL$GL_FLAGS+2, 2$
51 00000000' 00 D0 0001C MOVL XYZ_HANDLER, R1
61 6C FA 00023 CALLG (AP), (R1)
04 1A 04 00026 RET
04 BC 62 E8 00027 2$: BLBS (R2), 3$
02 C2 0002A SUBL2 #2, @SIGARG
7E 7C 0002E CLRQ -(SP)
BE AF 9F 00030 PUSHAB SEND_STRING
04 AC DD 00033 PUSHL SIGARG

```



```

: 404 0543 1 ROUTINE EDT_HANDLER_2 (SIGARG, MECHARG) =
: 405 0544 1 |++
: 406 0545 1 | FUNCTIONAL DESCRIPTION:
: 407 0546 1 |
: 408 0547 1 |         Handler to handle EDT signalled errors. Prevents EDT's fatal
: 409 0548 1 |         errors from exiting MAIL. This handler does not print the message.
: 410 0549 1 | --
: 411 0550 2 BEGIN
: 412 0551 2 MAP
: 413 0552 2     SIGARG : REF $BBLOCK,
: 414 0553 2     MECHARG : REF $BBLOCK;
: 415 0554 2
: 416 0555 2 BIND
: 417 0556 2     SIGNAME = SIGARG[CHF$$_SIG_NAME] : $BBLOCK;
: 418 0557 2
: 419 0558 2 IF .SIGNAME EQL S$$_UNWIND
: 420 0559 2     THEN RETURN S$$_CONTINUE;
: 421 0560 2
: 422 0561 2 IF NOT .SIGNAME
: 423 0562 2     THEN SIGNAME[ST$$V_SEVERITY] = ST$$K_WARNING;
: 424 0563 2
: 425 0564 2 SETUNWIND();
: 426 0565 2 RETURN 0
: 427 0566 1 END;

```

```

                                0000 00000 EDT_HANDLER_2:
                                .WORD
50      04 AC          04 C1 00002          ADDL3   Save nothing           : 0543
      00000920 8F      60 D1 00007          ADDL3   #4, SIGARG, R0         : 0556
                                04 12 0000E          CMPL   (R0), #2336       : 0558
                                01 D0 00010          BNEQ   1$
                                04 00013          MOVL   #1, R0           : 0559
                                07 8A 00017          RET
                                60 EB 00014 1$:      BLBS   (R0), 2$         : 0561
                                07 8A 00017          BICB2  #7, (R0)         : 0562
                                7E 7C 0001A 2$:      CLRQ   -(SP)           : 0564
      00000000G 00      02 FB 0001C          CALLS  #2, SY$$UNWIND
                                50 D4 00023          CLRL   R0              : 0565
                                04 00025          RET                    : 0566

```

: Routine Size: 38 bytes, Routine Base: \$CODE\$ + 032C

```

: 429 0567 1 ROUTINE CALL_EDT_FILEIO =
: 430 0568 1 |++
: 431 0569 1 | FUNCTIONAL DESCRIPTION:
: 432 0570 1 |
: 433 0571 1 |     Calls EDT$FILEIO.  If attempting to open the user's EDTINI, enable
: 434 0572 1 |     a condition handler so that the caller of this routine can get control
: 435 0573 1 |     back.
: 436 0574 1 |
: 437 0575 2 BEGIN
: 438 0576 2
: 439 0577 2 BUILTIN
: 440 0578 2     CALLG,
: 441 0579 2     AP,
: 442 0580 2     FP;
: 443 0581 2
: 444 0582 2 |
: 445 0583 2 | We only care about user's EDTINI not being found.  Let SIGNAL of FNF
: 446 0584 2 | on system EDTINI go through to EDT.
: 447 0585 2 |
: 448 0586 2 IF .COMFILCNT EQL 1
: 449 0587 2     THEN .FP = EDT_HANDLER_2;
: 450 0588 2
: 451 0589 2 RETURN CALLG(.AP,.EDT_FILEIO)
: 452 0590 1 END;

```

```

                                0000 00000 CALL_EDT_FILEIO:
                                .WORD   Save nothing
01 00000000' 00 91 00002          CMPB   COMFILCNT, #1
                                04 12 00009          BNEQ   1$
6D          CC  AF 9E 0000B          MOVAB  EDT_HANDLER_2, (FP)
51 00000000' 00 D0 0000F 1$:      MOVL   EDT_FILEIO, R1
61          6C  FA 00016          CALLG  (APT, (R1))
                                04 00019          RET
                                : 0567
                                : 0586
                                : 0587
                                : 0589
                                : 0590

```

: Routine Size: 26 bytes, Routine Base: \$CODE\$ + 0352

```
454 0591 1 ROUTINE MAIL$EDT_FILEIO(FILE_CODE,FILE_STREAM,RECORD_DESC,RHB_DESC) =
455 0592 1 |++
456 0593 1 | FUNCTIONAL DESCRIPTION:
457 0594 1 |
458 0595 1 |     This routine is called by EDT for all fileio. It passes on most
459 0596 1 |     calls, except for two special cases. For the second call to open
460 0597 1 |     the command
461 0598 1 |
462 0599 1 | --
463 0600 2 BEGIN
464 0601 2
465 0602 2 BUILTIN
466 0603 2     AP,
467 0604 2     CALLG;
468 0605 2
469 0606 2 MAP
470 0607 2     FILE_CODE : REF VECTOR[.BYTE],           !Assume codes are all bytes
471 0608 2     FILE_STREAM : REF VECTOR[.BYTE],         ! and stream numbers
472 0609 2     RECORD_DESC : REF $BLOCK,
473 0610 2     RHB_DESC : REF $BLOCK;
474 0611 2
475 0612 2 LOCAL
476 0613 2     STATUS;
477 0614 2
478 0615 2 |
479 0616 2 |     If opening the main input file, set the default file type of .TXT
480 0617 2 |
481 0618 2 IF .FILE_STREAM[0] EQL .EDT_K_INPUT_FILE
482 0619 2     AND .FILE_CODE[0] EQL .EDT_K_OPEN_INPUT
483 0620 3 THEN BEGIN
484 0621 3     STR$COPY_DX(.RHB_DESC,$DESCRIPTOR('.TXT'));
485 0622 3     RETURN CALLG(.AP,.EDT_FILEIO);
486 0623 3     END;
487 0624 2 |
488 0625 2 |     Everything except command file i/o, call EDT$FILEIO
489 0626 2 |
490 0627 2 IF .FILE_STREAM[0] NEQ .EDT_K_COMMAND_FILE
491 0628 2 THEN RETURN CALLG(.AP,.EDT_FILEIO)
492 0629 3 ELSE BEGIN
493 0630 3 |
494 0631 3 |     If opening the command file, set flags and call special routine
495 0632 3 |
496 0633 3 IF .FILE_CODE[0] EQL .EDT_K_OPEN_INPUT
497 0634 4 THEN BEGIN
498 0635 4     COMNOSUM = 0;
499 0636 4     COMNOFILE = 0;
500 0637 4     COMFILCNT = .COMFILCNT + 1;           !Count the edtini
501 0638 4 |
502 0639 4 |     This call will not return if there is a failure opening the
503 0640 4 |     SYSTEM-wide EDTINI, but will return if there is a failure
504 0641 4 |     opening the user's EDTINI
505 0642 4 |
506 0643 4     STATUS = CALLG(.AP,CALL_EDT_FILEIO);
507 0644 4     IF .COMFILCNT EQL 1                   !If opening user's edtini
508 0645 5 THEN BEGIN
509 0646 5     COMNOSUM = 1;                       !Flag need to output SET NOSUMMARY
510 0647 5     IF NOT .STATUS                       !If file not there, pretend it was
```

```

511 0648 5 THEN COMNOFILE = STATUS = 1;!Return ok so EDT will do a GET
512 0649 4 END:
513 0650 4 RETURN .STATUS
514 0651 4 END
515 0652 4 ELSE BEGIN
516 0653 4
517 0654 4 a code of other than OPEN_INPUT...
518 0655 4
519 0656 4 IF .FILE_CODE[0] EQL .EDT_K_GET
520 0657 5 THEN BEGIN
521 0658 5
522 0659 5 If we need to send the SET NOSUMMARY line, do it, and clear flag
523 0660 5
524 0661 5 IF .COMNOSUM
525 0662 6 THEN BEGIN
526 0663 6 STR$COPY_DX(.RECORD_DESC,$DESCRIPTOR('SET NOSUMMARY'));
527 0664 6 COMNOSUM = 0;
528 0665 6 RETURN 1;
529 0666 5 END:
530 0667 5 IF .COMNOFILE !If there was no file, return eof now
531 0668 5 THEN RETURN RMSS EOF
532 0669 5 ELSE RETURN CALLG(.AP,.EDT_FILEIO); !Otherwise, just get record
533 0670 4 END:
534 0671 4 IF .FILE_CODE[0] EQL .EDT_K_CLOSE
535 0672 4 AND .COMNOFILE
536 0673 4 THEN RETURN 1 !If closing non-ex file, pretend ok
537 0674 4 ELSE RETURN CALLG(.AP,.EDT_FILEIO); !EDT will close real file
538 0675 3 END:
539 0676 2 END:
540 0677 1 END:

```

```

54 58 54 2E 0036C P.ABR: .ASCII \.TXT\
00000004 00370 P.ABQ: .LONG 4
0000J000' 00374 .ADDRESS P.ABR
59 52 41 4D 4D 55 53 4F 4E 20 54 45 53 00378 P.ABT: .ASCII \SET NOSUMMARY\
00385 .BLKB 3
0000000D 00388 P.ABS: .LONG 13
00000000' 0038C .ADDRESS P.ABT

```

001C 0000 MAIL\$EDT\_FILEIO:

```

54 00000000G 00 9E 00002 .WORD Save R2,R3,R4 : 0591
53 D4 AF 9E 00009 MOVAB STR$COPY_DX, R4
52 00000000' 00 9E 0000D MOVAB P.ABQ, R3
08 00 ED 00014 MOVAB COMNOSUM, R2
08 13 12 0001B CMPZV #0, #8, @FILE_STREAM, EDT_K_INPUT_FILE : 0618
08 00 ED 0001D BNEQ 1$
0A 12 00024 CMPZV #0, #8, @FILE_CODE, EDT_K_OPEN_INPUT : 0619
10 53 DD 00026 BNEQ 1$
AC DD 00028 PUSHL R3 : 0621
64 02 FB 0002B PUSHL RHB_DESC
66 11 0002E CALLS #2, STR$COPY_DX
3C A2 08 BC 08 00 ED 00030 1$: BRB 6$ : 0622
CMPZV #0, #8, @FILE_STREAM, EDT_K_COMMAND_FILE : 0627

```

44	A2	04	BC	08		5D 12 00037	BNEQ	6\$			
						00 ED 00039	CMPZV	#0, #8, @FILE_CODE, EDT_K_OPEN_INPUT			0633
						1E 12 00040	BNEQ	2\$			
						62 B4 00042	CLRW	COMNOSUM			0635
					02	A2 96 00044	INCB	COMFILCNT			0637
		FF76	CF			6C FA 00047	CALLG	(AP), CALL_EDT_FILEIO			0643
			01		02	A2 91 0004C	CMPB	COMFILCNT, #1			0644
						4B 12 00050	BNEQ	7\$			
			62			01 90 00052	MOVB	#1, COMNOSUM			0646
			45			50 E8 00055	BLBS	STATUS, 7\$			0647
			50			01 D0 00058	MOVL	#1, STATUS			0648
		01	A2			01 90 0005B	MOVB	#1, COMNOFILE			
						04 0005F	RET				0652
48	A2	04	BC	08		00 ED 00060	CMPZV	#0, #8, @FILE_CODE, EDT_K_GET	2\$:		0656
						1C 12 00067	BNEQ	4\$			
				0D		62 E9 00069	BLBC	COMNOSUM, 3\$			0661
					18	A3 9F 0006C	PUSHAB	P.ABS			0663
					0C	AC DD 0006F	PUSHL	RECORD_DESC			
				64		02 FB 00072	CALLS	#2, STR\$COPY_DX			
						62 94 00075	CLRB	COMNOSUM			0664
						19 11 00077	BRB	5\$			0665
			19		01	A2 E9 00079	BLRC	COMNOFILE, 6\$	3\$:		0667
			50		0001827A	8F D0 0007D	MOVL	#98938, R0			0669
						04 00084	RET				
4C	A2	04	BC	08		00 ED 00085	CMPZV	#0, #8, @FILE_CODE, EDT_K_CLOSE	4\$:		0671
						08 12 0008C	BNEQ	6\$			
			04		01	A2 E9 0008E	BLBC	COMNOFILE, 6\$			0672
			50			01 D0 00092	MOVL	#1, R0	5\$:		0674
						04 00095	RET				
			51		54	A2 D0 00096	MOVL	EDT_FILEIO, R1	6\$:		
			61			6C FA 0009A	CALLG	(AP), (R1)			
						04 0009D	RET		7\$:		0677

; Routine Size: 158 bytes, Routine Base: \$CODE\$ + 0390

; F



```

: 542 0678 1 ROUTINE EDT_XLATE (C_DESC) =
: 543 0679 1 |++
: 544 0680 1 | FUNCTIONAL DESCRIPTION:
: 545 0681 1 |
: 546 0682 1 |         Process the XLATE command.  Specifically, SPAWN and ATTACH
: 547 0683 1 |
: 548 0684 1 | |--
: 549 0685 2 BEGIN
: 550 0686 2 MAP
: 551 0687 2     C_DESC : REF $BBLOCK;
: 552 0688 2
: 553 0689 2 ENABLE EDT_HANDLER_2;
: 554 0690 2
: 555 0691 2 IF CLISDCL_PARSE(.C_DESC,MAIL$SPAWN_CMDS,LIB$GET_INPUT)
: 556 0692 2     THEN CLISDISPATCH();
: 557 0693 2
: 558 0694 2 STR$FREE1_DX(.C_DESC);
: 559 0695 2
: 560 0696 2 RETURN TRUE
: 561 0697 1 END;

```

```

                                0000 0000 EDT_XLATE:
                                .WORD   Save nothing
                                MOVAL   2$, (FP)
                                PUSHAB  LIB$GET_INPUT
                                PUSHAB  MAIL$SPAWN_CMDS
                                PUSHL   C_DESC
                                CALLS   #3, CLISDCL_PARSE
                                BLBC    R0, 1$
                                CALLS   #0, CLISDISPATCH
                                PUSHL   C_DESC
                                CALLS   #1, STR$FREE1_DX
                                MOVL    #1, R0
                                RET
                                .WORD   Save nothing
                                CLRL    -(SP)
                                PUSHL   SP
                                MOVQ   4(AP), -(SP)
                                CALLS   #3, EDT_HANDLER_2
                                RET

```

; Routine Size: 69 bytes, Routine Base: \$CODE\$ + 042E

```
563 0698 1 GLOBAL ROUTINE MAIL$EDIT(IN_NAME,OUT_NAME) =
564 0699 1
565 0700 1 !++
566 0701 1 FUNCTIONAL DESCRIPTION:
567 0702 1
568 0703 1 This routine is called to enter and edit a mail message.
569 0704 1
570 0705 1 INPUTS:
571 0706 1
572 0707 1 IN_NAME = Address of input file name descriptor.
573 0708 1 OUT_NAME = Address of output file name descriptor.
574 0709 1
575 0710 1 OUTPUTS:
576 0711 1
577 0712 1 Message copied to file OUT_NAME.
578 0713 1
579 0714 1 --
580 0715 2 BEGIN
581 0716 2 MAP
582 0717 2 IN_NAME: REF BLOCK[.BYTE], ! Input file desc
583 0718 2 OUT_NAME: REF BLOCK[.BYTE]; ! Output file desc
584 0719 2
585 0720 2 BUILTIN
586 0721 2 AP,
587 0722 2 CALLG;
588 0723 2
589 0724 2 LOCAL
590 0725 2 STATUS, ! Routine status
591 0726 2 FILEIO_BPV : VECTOR[2, LONG],
592 0727 2 XLATE_BPV : VECTOR[2, LONG],
593 0728 2 EDITCMD_BUF: BLOCK[255, BYTE], ! Edit command buffer
594 0729 2 EDITCMD_DESC: BLOCK[8, BYTE]; ! Edit command desc
595 0730 2
596 0731 2 MAIL$DISABLE_CTRL(C);
597 0732 2
598 0733 2 If callable EDT requested then call EDT, otherwise create the
599 0734 2 editor subprocess.
600 0735 2
601 0736 2 IF .MAIL$GL_FLAGS[MAIF_V_CALEDT]
602 0737 3 THEN BEGIN
603 P 0738 3 IF _ERR(MAIL$LOAD_EDT());
604 0739 3 RETURN .STATUS;
605 0740 3 IF .MAIL$GL_FLAGS[MAIF_V_EDTEDT]
606 0741 4 THEN BEGIN
607 0742 4 FILEIO_BPV[0] = MAIL$EDT_FILEIO;
608 0743 4 FILEIO_BPV[1] = 0;
609 0744 4 XLATE_BPV[0] = EDT_XLATE;
610 0745 4 XLATE_BPV[1] = 0;
611 0746 4 COMF1[CNT = -1;
612 0747 6 IF NOT (STATUS=(.EDT EDIT)((IF .IN_NAME[DSC$W_LENGTH] EQL 0
613 0748 6 THEN NL_DESC
614 0749 5 ELSE .IN_NAME),
615 0750 5 .OUT_NAME, NO FILE_DESC, JOU_FILE_DESC,
616 0751 5 %REFTO), FILEIO_BPV, 0, XLATE_BPV))
617 0752 4 THEN RETURN .STATUS;
618 0753 4 END
619 0754 4 ELSE IF NOT (STATUS = CALLG(.AP, .EDT_EDIT))
```

```

: 620      0755      3 THEN RETURN .STATUS;
: 621      0756      3
: 622      0757      3 END
: 623      0758      3 ELSE BEGIN
: 624      0759      3     . Create and execute the command procedure execution command.
: 625      0760      3
: 626      0761      3 EDITCMD_DESC[0,0,32,0] = 255;
: 627      0762      3 EDITCMD_DESC[DSC$A_POINTER] = EDITCMD_BUF;
: 628      P 0763      3 $FAO(  -DESCRIPTOR('$ @!AS '!AS' '!AS'),
: 629      P 0764      3     EDITCMD_DESC[DSC$W_LENGTH],
: 630      P 0765      3     EDITCMD_DESC,
: 631      P 0766      3     (IF .MAIL$GL_FLAGS[MAIF V USEREDIT]
: 632      P 0767      3         THEN MAIL$Q EDITNAM_DESC
: 633      P 0768      3         ELSE DESCRIPTOR('SYS$SYSTEM:MAILEDIT.COM')),
: 634      P 0769      3     .IN_NAME,
: 635      0770      3     .OUT_NAME);
: 636      P 0771      3 IF_ERR(LIB$SPAWN(EDITCMD_DESC),
: 637      0772      3     SIGNAL(MAIL$_EDITPROC,0,.STATUS));
: 638      0773      3 END;
: 639      0774      2 RETURN CHECK_MESSAGE_FILE(.OUT_NAME)
: 640      0775      1 END;

```

21	22	20	22	53	41	21	22	20	53	41	21	40	20	24	00473	P.ABV:	.ASCII	\\$ @!AS '!AS' '!AS'	:		
												22	53	41	00482					:	
															00000012	00485	P.ABU:	.BLKB	3	:	
															00000000'	00488		.LONG	18	:	
															00000000'	0048C		.ADDRESS	P.ABV	:	
4C	49	41	4D	3A	4D	45	54	53	59	53	24	53	59	53	00490	P.ABX:	.ASCII	\SYS\$SYSTEM:MAILEDIT.COM	:		
							4D	4F	43	2E	54	49	44	45	0049F					:	
															004A7			.BLKB	1	:	
															00000017	004A8	P.ABW:	.LONG	23	:	
															00000000'	004AC		.ADDRESS	P.ABX	:	
																		.EXTRN	SYS\$FAO	:	
															001C	00000		.ENTRY	MAIL\$EDIT, Save R2,R3,R4	:	0698
							54	00000000G	00	9E	00002						MOVAB	MAIL\$GL_FLAGS, R4	:		
							53	FCE3	CF	9E	00009						MOVAB	MAIL\$LOAD_EDT, R3	:		
							5E	FEE4	CE	9E	0000E						MOVAB	-284(SP), -SP	:		
							00000000G	00	00	FB	00013						CALLS	#0, MAIL\$DISABLE_CTRL	:	0731	
									01	A4	95	0001A					TSTB	MAIL\$GL_FLAGS+1	:	0736	
										6D	18	0001D					BGEQ	6\$	:		
									63	00	FB	0001F					CALLS	#0, MAIL\$LOAD_EDT	:	0739	
									01	50	E8	00022					BLBS	STATUS, 1\$	:		
											04	00025					RET		:		
									52	00000000'	00	D0	00026	1\$:		MOVL	EDT EDIT, R2	:	0747		
									4E	02	A4	E9	0002D			BLBC	MAIL\$GL_FLAGS+2, 4\$	:	0740		
							F8	AD	01F0	C3	9E	00031				MOVAB	MAIL\$EDT_FILEIO, FILEIO_BP	:	0742		
										FC	AD	D4	00037			CLRL	FILEIO_BP+4	:	0743		
							F0	AD	028E	C3	9E	0003A				MOVAB	EDT_XLATE, XLATE_BP	:	0744		
										F4	AD	D4	00040			CLRL	XLATE_BP+4	:	0745		
							00000000'	00	01	8E	00043					MNEGB	#1, COMFILCNT	:	0746		
										F0	AD	9F	0004A			PUSHAB	XLATE_BP	:	0747		
										7E	D4	0004D				CLRL	-(SP)	:			
										F8	AD	9F	0004F			PUSHAB	FILEIO_BP	:			

			OC	AE	D4	00052		CLRL	12(SP)		: 0751
			OC	AE	9F	00055		PUSHAB	12(SP)		:
			FB30	CF	9F	00058		PUSHAB	JOU FILE DESC		: 0747
			FAFC	CF	9F	0005C		PUSHAB	NO FILE DESC		:
			08	AC	DD	00060		PUSHL	OUT_NAME		: 0750
			04	BC	B5	00063		TSTW	@IN_NAME		: 0747
				09	12	00066		BNEQ	2\$		:
		50	FAE8	CF	9E	00068		MOVAB	NL_DESC, R0		:
				50	DD	0006D		PUSHL	R0		:
				03	11	0006F		BRB	3\$		:
				04	AC	DD	00071	2\$:	PUSHL	IN_NAME	: 0749
		62		08	FB	00074	3\$:	CALLS	#8, (R2)		: 0747
		51		50	DD	00077		MOVL	R0, STATUS		:
		61		51	E8	0007A		BLBS	STATUS, 9\$		:
				09	11	0007D		BRB	5\$		: 0752
		62		6C	FA	0007F	4\$:	CALLG	(AP), (R2)		: 0754
		51		50	DD	00082		MOVL	R0, STATUS		:
		56		50	E8	00085		BLBS	R0, 9\$		:
		50		51	DD	00088	5\$:	MOVL	STATUS, R0		: 0755
					04	0008B		RET			:
	04		AE	FF	8F	9A	0008C	6\$:	MOVZBL	#255, EDITCMD_DESC	: 0761
	08		AE	OC	AE	9E	00091		MOVAB	EDITCMD_BUF, EDITCMD_DESC+4	: 0762
			7E	04	AC	7D	00096		MOVQ	IN_NAME, -(SP)	: 0770
09	01		A4		05	E1	0009A		BBC	#5, MAIL\$GL_FLAGS+1, 7\$	:
			50	00000000G	00	9E	0009F		MOVAB	MAIL\$Q_EDITNAM_DESC, R0	:
					05	11	000A6		BRB	8\$	:
			50	FF4C	CF	9E	000A8	7\$:	MOVAB	P.ABW, R0	: 0772
					50	DD	000AD	8\$:	PUSHL	R0	:
				10	AE	9F	000AF		PUSHAB	EDITCMD_DESC	:
				14	AE	9F	000B2		PUSHAB	EDITCMD_DESC	:
				FF1F	CF	9F	000B5		PUSHAB	P.ABU	:
		00000000G	00		06	FB	000B9		CALLS	#6, SYSS\$FAO	:
				04	AE	9F	000C0		PUSHAB	EDITCMD_DESC	: 0772
		00000000G	00		01	FB	000C3		CALLS	#1, LIB\$SPAWN	:
			11		50	E8	000CA		BLBS	STATUS, 9\$	:
					50	DD	000CD		PUSHL	STATUS	:
					7E	D4	000CF		CLRL	-(SP)	:
				007E8062	8F	DD	000D1		PUSHL	#8290402	:
		00000000G	00		03	FB	000D7		CALLS	#3, LIB\$SIGNAL	:
				08	AC	DD	000DE	9\$:	PUSHL	OUT_NAME	: 0774
	00B1	C3			01	FB	000E1		CALLS	#1, CHECK_MESSAGE_FILE	:
					04	000E6		RET			: 0775

; Routine Size: 231 bytes, Routine Base: \$CODE\$ + 0480

```
0776 1 GLOBAL ROUTINE MAIL$ENTER(OUT_RAB) =
0777 1 ++
0778 1 FUNCTIONAL DESCRIPTION:
0779 1
0780 1 This routine is called to enter a mail message.
0781 1
0782 1 INPUTS:
0783 1
0784 1 OUT_RAB = Address of a RAB connected to an open output file.
0785 1
0786 1 OUTPUTS:
0787 1
0788 1 SYSS$INPUT copied to file OUT_RAB.
0789 1 --
0790 2 BEGIN
0791 2 MAP
0792 2 LOCAL OUT_RAB: REF BLOCK[,BYTE]; ! Output file RAB
0793 2 LOCAL DESC: BLOCK[8,BYTE], ! String desc
0794 2 STATUS; ! Routine status
0795 2
0796 2
0797 2 IF .MAIL$GL_FLAGS[MAIF V_ITEM]
0798 2 THEN MAIL$PUT_OUTPUT(MAIL$GETMSG(MAIL$_EDIT10));
0799 2
0800 2 ! Read from input and copy to output file
0801 2
0802 2 DESC[DSC$W_LENGTH] = 0;
0803 2 DESC[DSC$B_CLASS] = DSC$K_CLASS_D;
0804 2 DESC[DSC$B_DTYPE] = 0;
0805 2 DESC[DSC$A_POINTER] = 0;
0806 2 STATUS = TRUE;
0807 2
0808 2 OUT_RAB[RAB$_CTX] = MAIL$_WRITEERR;
0809 2 WHILE 1
0810 2 DO
0811 3 BEGIN
0812 3 DO
0813 4 BEGIN
0814 4 STATUS = SMG$READ_STRING(MAIL$_SMG_KEYBOARD,DESC);
0815 4 IF (.STATUS EQL SSS$_BADESCAPE) THEN
0816 4 SIGNAL(RMS$_BES,0,MAIL$_REENTER);
0817 4 END
0818 3 UNTIL (.STATUS NEQ SSS$_BADESCAPE);
0819 4 IF (.STATUS EQL RMS$_TNS)
0820 4 OR (.STATUS EQL SSS$_DATAOVERUN)
0821 3 THEN STATUS = SSS$_NORMAL;
0822 3 IF NOT .STATUS
0823 4 THEN BEGIN
0824 4 IF .STATUS EQL SMG$_EOF
0825 4 THEN EXITLOOP;
0826 4 SIGNAL(MAIL$_READERR,1,SYSIN_DESC,.STATUS);
0827 4 EXITLOOP;
0828 3 END;
0829 3 IF .DESC[DSC$W_LENGTH] GTRU 255
0830 4 THEN BEGIN
0831 4 SIGNAL(MAIL$_READERR,1,SYSIN_DESC,RMS$_RTB,.DESC[DSC$W_LENGTH]);
0832 4 STATUS = RMS$_RTB;
```

```

699 0833 4          EXITLOOP;
700 0834 3          END;
701 0835 3          OUT_RAB[RAB$W_RSZ] = .DESC[DSC$W_LENGTH];
702 0836 3          OUT_RAB[RAB$L_RBF] = .DESC[DSC$A_POINTER];
703 0837 4          IF NOT (STATUS = $PUT(RAB = .OUT_RAB, ERR=UTIL$REPORT, IO_ERROR))
704 0838 3          OR .MAIL$GL_FLAGS[MAIF_V_CTR[CFL]] THEN EXITLOOP (STATUS = 0);
705 0839 2          END;
706 0840 2
707 0841 2 STR$FREE1_DX(DESC);
708 0842 2
709 0843 3 RETURN (IF .STATUS
710 0844 4         OR (.STATUS EQL SMGS_EOF)
711 0845 3         THEN TRUE
712 0846 3         ELSE MAIL$_SENDABORT)
713 0847 1 END;

```

```

                                .EXTRN  SYSS$PUT
                                .ENTRY  MAIL$ENTER, Save R2,R3,R4,R5,R6
007C 00000
56      FAB3      CF 9E 00002
55 00000000G    8F DD 00007
54 00000000G    00 9E 0000E
5E      00      08 C2 00015
16 000C0000G    00 02 E1 00018
      007EDE98    8F DD 00020
      00000000G    00 01 FB 00026
      00000000G    00 50 DD 0002D
      00000000G    00 01 FB 0002F
6E 02000000    8F DD 00036 1$:
      04      AE D4 0003D
53      00      01 D0 00040
52      04      AC D0 00043
18 A2 007E10D0    8F DD 00047
      00000000G    5E DD 0004F 2$:
      00000000G    00 9F 00051
00000000G    00 02 FB 00057
53      50 DD 0005E
3C      53 D1 00061
      11 12 00064
      007EBEBB    8F DD 00066
      000181C0    7E D4 0006C
64      8F DD 0006E
3C      03 FB 00074
      53 D1 00077 3$:
000181B8    8F 53 D1 0007A
      09 13 0007C
00000838    8F 53 D1 00085
      03 12 0008C
53      01 D0 0008E 4$:
16      53 E8 00091 5$:
55      53 D1 00094
      61 13 00097
      53 DD 00099
      56 DD 0009B
                                .ENTRY  MAIL$ENTER, Save R2,R3,R4,R5,R6
MOVAB  SYSIN_DESC, R6
MOVL   #SMGS_EOF, R5
MOVAB  LIB$SIGNAL, R4
SUBL2  #8, SP
BBC    #2, MAIL$GL_FLAGS, 1$
PUSHL  #8314520
CALLS  #1, MAIL$GETMSG
PUSHL  R0
CALLS  #1, MAIL$PUT_OUTPUT
MOVL   #33554432, DESC
CLRL   DESC+4
MOVL   #1, STATUS
MOVL   GJT_RAB, R2
MOVL   #8281840, 24(R2)
PUSHL  SP
PUSHAB MAIL$_SMG_KEYBOARD
CALLS  #2, SMGS_READ_STRING
MOVL   R0, STATUS
CPL    STATUS, #60
BNEQ   3$
PUSHL  #8306363
CLRL   -(SP)
PUSHL  #98752
CALLS  #3, LIB$SIGNAL
CPL    STATUS, #60
BEQL   2$
CPL    STATUS, #98744
BEQL   4$
CPL    STATUS, #2104
BNEQ   5$
MOVL   #1, STATUS
BLBS   STATUS, 6$
CPL    STATUS, R5
BEQL   9$
PUSHL  STATUS
PUSHL  R6

```

```

0776
0797
0798
0802
0805
0806
0808
0814
0815
0816
0818
0819
0820
0821
0822
0824
0826

```

		01	DD	0009D	PUSHL	#1	
	007E10B2	8F	DD	0009F	PUSHL	#8261810	
64		04	FB	000A5	CALLS	#4, LIB\$SIGNAL	
		50	11	000A8	BRB	9\$	0823
00FF		8F	B1	000AA 6\$:	CMPW	DESC, #255	0829
		1F	1B	000AF	BLEQU	7\$	
7E		6E	3C	000B1	MOVZWL	DESC, -(SP)	0831
	000181A8	8F	DD	000B4	PUSHL	#98728	
		56	DD	000BA	PUSHL	R6	
		01	DD	000BC	PUSHL	#1	
	007E10B2	8F	DD	000BE	PUSHL	#8261810	
64		05	FB	000C4	CALLS	#5, LIB\$SIGNAL	
53	000181A8	8F	D0	000C7	MOVL	#98728, STATUS	0832
		2A	11	000CE	BRB	9\$	0830
22	A2	6E	B0	000D0 7\$:	MOVW	DESC, 34(R2)	0835
28	A2	AE	D0	000D4	MOVL	DESC+4, 40(R2)	0836
	00000000G	00	9F	000D9	PUSHAB	UTIL\$REPORT_IO_ERROR	0837
		52	DD	000DF	PUSHL	R2	
00000000G	00	02	FB	000E1	CALLS	#2, SYSSPUT	
	53	50	D0	000E8	MOVL	R0, STATUS	
	0A	53	E9	000EB	3LBC	STATUS, 8\$	
	03	00	E8	000EE	BLBS	MAIL\$GL_FLAGS+1, 8\$	0838
		FF	31	000F5	BRW	2\$	
		53	D4	000F8 8\$:	CLRL	STATUS	
		5E	DD	000FA 9\$:	PUSHL	SP	0841
00000000G	00	01	FB	000FC	CALLS	#1, STR\$FREE1_DX	
	05	53	E8	00103	BLBS	STATUS, 10\$	0843
	55	53	D1	00106	CMPL	STATUS, R5	0844
		04	12	00109	BNEQ	11\$	
	50	01	D0	0010B 10\$:	MOVL	#1, R0	0843
		04	00	10E	RET		
	50	8F	D0	0010F 11\$:	MOVL	#8290394, R0	
		04	00	116	RET		0847

; Routine Size: 279 bytes, Routine Base: \$CODE\$ + 0597

```
0848 1 ROUTINE EDT_EXTRACT_IO(FILE_CODE,FILE_STREAM,RECORD_DESC,RMB_DESC) =
0849 1 |++
0850 1 | FUNCTIONAL DESCRIPTION:
0851 1 |
0852 1 |     This routine is called by EDT when an EDT/EXTRACT is done.
0853 1 |     This routine only handles input file i/o, and passes the rest
0854 1 |     onto EDT's input routine
0855 1 |
0856 1 | --
0857 2 BEGIN
0858 2
0859 2 LOCAL
0860 2     STATUS,
0861 2     MORE_FLAG,
0862 2     DESC : VECTOR[2, LONG];
0863 2
0864 2 BUILTIN
0865 2     AP,
0866 2     CALLG;
0867 2
0868 2 MAP
0869 2     FILE_CODE : REF VECTOR[BYTE],
0870 2     FILE_STREAM : REF VECTOR[BYTE],
0871 2     RECORD_DESC : REF $BLOCK,
0872 2     RMB_DESC : REF $BLOCK;
0873 2
0874 2 | If not input file i/o then pass it on to EDT
0875 2 |
0876 2 IF .FILE_STREAM[0] NEQ .EDT_K_INPUT_FILE
0877 2 THEN RETURN CALLG(.AP, .EDT_FICEIO)
0878 3 ELSE BEGIN
0879 3     IF .FILE_CODE[0] EQL .EDT_K_OPEN_INPUT
0880 3     OR .FILE_CODE[0] EQL .EDT_K_CLOSE
0881 3     THEN RETURN TRUE;
0882 3     IF .FILE_CODE[0] EQL .EDT_K_GET
0883 4     THEN BEGIN
0884 4         IF .MAIL$GL_FLAGS[MAIF_V_ISAM]
0885 5         THEN BEGIN
0886 5             IF NOT MAIL$GET_I MSG(DESC, MORE_FLAG)
0887 5             THEN RETURN RM$ EOF;
0888 5             STR$COPY DX(.RECORD_DESC, DESC);
0889 5             RETURN TRUE;
0890 5             END
0891 5         ELSE BEGIN
0892 5             IF NOT .MAIL$L MSG STS
0893 5             THEN RETURN RM$ EOF;
0894 5             DESC[0] = .MAIL$W MSG_RSIZ;
0895 5             DESC[1] = .MAIL$L MSG_RADR;
0896 5             STR$COPY DX(.RECORD_DESC, DESC);
0897 5             MAIL$GET MSG();
0898 5             RETURN TRUE;
0899 5             END
0900 4         END
0901 3     ELSE SIGNAL(0);
0902 3     RETURN TRUE
0903 2     END;
0904 1 END;
```



000C 00000 EDT\_EXTRACT IO:

				53	00000000G	00	9E	00002	.WORD	Save R2,R3	0848
				52	00000000'	00	9E	00009	MOVAB	STR\$COPY_DX, R3	
				5E		0C	C2	00010	MOVAB	EDT_K_INPUT_FILE, R2	
62		08	BC	08		00	ED	00013	SUBL2	#12, SP	
				08		08	13	00019	CMPZV	#0, #8, @FILE_STREAM, EDT_K_INPUT_FILE	0876
				51		14	A2	0001B	BEQL	1\$	
				61			6C	0001F	MOVL	EDT_FILEIO, R1	0877
								00022	CALLG	(APT, (R1))	
04	A2	04	BC	08			00	ED	RET		0878
								00023	CMPZV	#0, #8, @FILE_CODE, EDT_K_OPEN_INPUT	0879
								0002A	BEQL	6\$	
0C	A2	04	BC	08			00	ED	CMPZV	#0, #8, @FILE_CODE, EDT_K_CLOSE	0880
								0002C	BEQL	6\$	
								00033	CMPZV	#0, #8, @FILE_CODE, EDT_K_GET	0882
08	A2	04	BC	08			00	ED	BEQL	6\$	
								00035	CMPZV	#0, #8, @FILE_CODE, EDT_K_GET	0882
								0003C	BNEQ	5\$	
				1A	00000000G	00	E9	0003E	BLBC	MAIL\$GL_FLAGS, 2\$	0884
								00045	PUSHL	SP	0886
						08	AE	00047	PUSHAB	DESC	
				00000000G	00		02	FB	CALLS	#2, MAIL\$GET_I_MSG	
								0004A	BLBC	R0, 3\$	
								00051	PUSHAB	DESC	0888
						04	AE	00054	PUSHL	RECORD_DESC	
						0C	AC	00057	CALLS	#2, STR\$COPY_DX	
				63			02	FB	BRB	6\$	0891
								0005D	BLBS	MAIL\$L_MSG_STS, 4\$	0892
				08	00000000G	00	E8	0005F	MOVZWL	MAIL\$W_MSG_RSIZ, DESC	0894
				50	0001C27A	8F	D0	00066	MOVL	MAIL\$L_MSG_RADR, DESC+4	0895
								0006D	PUSHAB	DESC	0896
								0006E	PUSHL	RECORD_DESC	
04	AE	00000000G		00			3C	0006E	CALLS	#2, STR\$COPY_DX	
08	AE	00000000G		00			D0	00076	CALLS	#0, MAIL\$GET_MSG	0897
								0007E	BRB	6\$	0898
								00081	CLRL	-(SP)	0901
								00084	CALLS	#1, LIB\$SIGNAL	
				00000G00G	00		00	FB	MOVL	#1, R0	0902
								00087	RET		0904
								0008E			
								00090			
				00000000G	00		01	FB			
								00092			
								00099			
								0009C			

; Routine Size: 157 bytes, Routine Base: \$CODE\$ + 06AE

```

773 0905 1 ROUTINE COPY_MSG(FAB,RAB,NAM) =
774 0906 1 |++
775 0907 1 | Copy the current message to a temporary file
776 0908 1 |
777 0909 1 |
778 0910 1 |--
779 0911 2 BEGIN
780 0912 2 MAP
781 0913 2     FAB : REF $BBLOCK,
782 0914 2     RAB : REF $BBLOCK,
783 0915 2     NAM : REF $BBLOCK;
784 0916 2
785 0917 2 LOCAL
786 0918 2     SND_XABPRO : $XABPRO_DECL,
787 0919 2     STATUS,
788 0920 2     DESC : VECTOR[2, LONG];
789 0921 2
790 0922 2     DESC[0] = NAM$C MAXRSS;
791 0923 2     DESC[1] = .NAM[NAM$R RSA];
792 0924 2     $FAO((IF .MAIL$L MYPID<16,16,0> NEQ 0
793 0925 2         THEN $DESCRIPTOR('SYSS$SCRATCH:MAIL_!XL_EDIT.TMP')
794 0926 2         ELSE $DESCRIPTOR('SYSS$SCRATCH:MAIL_!XW_EDIT.TMP')),
795 0927 2         DESC, DESC, .MAIL$L MYPID);
796 0928 2     $FAB_INIT(FAB=.FAB,
797 0929 2         FNS=.DESC[0],
798 0930 2         FNA=.DESC[1],
799 0931 2         FAC=(PUT, TRN),
800 0932 2         CTX=MAIL$_OPENOUT,
801 0933 2         NAM=.NAM,
802 0934 2         RAT=CR,
803 0935 2         RFM=VAR);
804 0936 2
805 0937 2     $XABPRO_INIT(XAB=SND_XABPRO,
806 0938 2         UIC=(0,0));
807 0939 2
808 0940 2     SND_XABPRO[XAB$W_PRO] = MAIL$_TMP_PRO;
809 0941 2
810 0942 2     IF_ERR($CREATE(FAB=.FAB, ERR=UTIL$REPORT_IO_ERROR);,
811 0943 2         RETURN .STATUS);
812 0944 2
813 0945 2     FAB[FAB$L_XAB] = 0;
814 0946 2     FAB[FAB$B_FNS] = .NAM[NAM$B_RSL];
815 0947 2     DESC[0] = .NAM[NAM$B_RSL];
816 0948 2     $RAB_INIT(RAB=.RAB,
817 0949 2         CTX=MAIL$_OPENOUT,
818 0950 2         FAB=.FAB);
819 0951 2
820 0952 2     IF_ERR($CONNECT(RAB=.RAB, ERR=UTIL$REPORT_IO_ERROR);,
821 0953 2         .FAB[FAB$V_DLT] = 1;
822 0954 2         $CLOSE(FAB=.FAB);
823 0955 2         RETURN .STATUS);
824 0956 2
825 0957 2     RAB[RAB$L_CTX] = MAIL$_WRITEERR;
826 0958 2     STATUS = MAIL$EXTRACT_MESSAGE(1, .RAB);
827 0959 2     FAB[FAB$L_CTX] = MAIL$_CLOSEOUT;
828 0960 2     IF_ERR($CLOSE(FAB=.FAB, ERR=UTIL$REPORT_IO_ERROR);,
829 0961 2         RETURN .STATUS);

```

```

: 830      0962 2
: 831      0963 2 STR$COPY DX(INPUT_DESC,DESC);
: 832      0964 2 RETURN TRUE
: 833      0965 1 END;

```

```

49 41 4D 3A 48 43 54 41 52 43 53 24 53 59 53 0074B P.ABZ: .ASCII \SYS$SCRATCH:MAIL_!XL_EDIT.TMP\
50 4D 54 2E 54 49 44 45 5F 4C 58 21 5F 4C 0075A
                                0000001D 00768 P.ABY: .LONG 29
                                00000000' 0076C .ADDRESS P.ABZ
49 41 4D 3A 48 43 54 41 52 43 53 24 53 59 53 00770 P.ACB: .ASCII \SYS$SCRATCH:MAIL_!XW_EDIT.TMP\
50 4D 54 2E 54 49 44 45 5F 57 58 21 5F 4C 0077F
                                0078D
                                0000001D 00790 P.ACA: .BLKB 3
                                00000000' 00794 .LONG 29
                                                .ADDRESS P.ACB

                                .EXTRN SYS$CREATE, SYS$CONNECT
                                .EXTRN SYS$CLOSE

```

03FC 0000 COPY\_MSG:

```

59 00000000G 00 9E 00002 .WORD Save R2,R3,R4,R5,R6,R7,R8,R9
58 00000000G 00 9E 00009 MOVAB SYS$CLOSE, R9
5E A4 AE 9E 00010 MOVAB UTIL$REPORT_IO_ERROR, R8
7E FF 8F 9A 00014 MOVAB -92(SP), SP
56 0C AC D0 00018 MOVZBL #255, DESC
04 AE 04 A6 D0 0001C MOVL NAM, R6
                                00000000G 00 DD 00021 MOVL 4(R6), DESC+4
                                04 AE 9F 00027 PUSHL MAIL$MYPID
                                08 AE 9F 0002A PUSHAB DESC
                                00000000G 00 B5 0002D PUSHAB DESC
                                06 13 00033 TSTW MAIL$MYPID+2
50 98 AF 9E 00035 BEQL 1$
                                04 11 00039 MOVAB P.ABY, R0
50 BA AF 9E 0003B BRB 2$
                                50 DD 0003F MOVAB P.ACA, R0
00000000G 00 04 FB 00041 1$:
                                57 04 AC D0 00048 2$:
                                6E 00 2C 0004C CALLS #4, SYS$FAO
0050 8F 00 67 8F B0 00054 MOVW #20483, (R7)
                                16 A7 11 90 00059 MOVW #17, 22(R7)
                                18 A7 007E10A2 8F D0 0005D MOVL #8261794, 24(R7)
                                1E A7 0202 8F B0 00065 MOVW #514, 30(R7)
                                28 A7 56 D0 0006B MOVL R6, 40(R7)
                                2C A7 04 AE D0 0006F MOVL DESC+4, 44(R7)
0058 8F 00 34 A7 6E 90 00074 MOVW DESC, 52(R7)
                                6E 00 2C 00078 MOVW #0, (SP), #0, #80, (R7)
                                08 AE 0007F
                                08 AE 5813 8F B0 00081 MOVW #22547, $RMS_PTR
                                10 AE 0000G 8F B0 00087 MOVW #MAIL$C_TMP_PRO, SND_XABPRO+8
                                0180 8F BB 0008D PUSHR #^M<R7,R8>
                                00000000G 00 02 FB 00091 CALLS #2, SYS$CREATE
                                6E 50 E9 00098 BLBC STATUS, 4$
                                34 A7 24 D4 0009B CLRL 36(R7)
                                03 A6 90 0009E MOVW 3(R6), 52(R7)

```

```

: 0905
:
:
: 0922
: 0923
:
: 0927
:
:
:
: 0935
:
:
:
: 0938
:
: 0940
: 0943
:
: 0945
: 0946

```

0044	8F	00	6E	03	A6	9A	000A3	MOVZBL	3(R6), DESC	0947
			56	08	AC	D0	000A7	MOVL	RAB, R6	0950
			6E		00	2C	000AB	MOVCS	#0, (SP), #0, #68, (R6)	
					66		000B2			
			66	4401	8F	B0	000B3	MOVW	#17409, (R6)	
		18	A6	007E10A2	8F	D0	000B8	MOVL	#8261794, 24(R6)	
		3C	A6		57	D0	000C0	MOVL	R7, 60(R6)	
				0140	8F	BB	000C4	PUSHR	#*M<R6,R8>	0955
		00000000G	00		02	FB	000C8	CALLS	#2, SYS\$CONNECT	
			52		50	D0	000CF	MOVL	R0, STATUS	
			12		52	EB	000D2	BLBS	STATUS, 3\$	
	50	05	A7		01	EF	000D5	EXTZV	#7, #1, 5(R7), R0	
					01	D0	000DB	MOVL	#1, (R0)	
					57	DD	000DE	PUSHL	R7	
			69		01	FB	000E0	CALLS	#1, SYS\$CLOSE	
			50		52	D0	000E3	MOVL	STATUS, R0	
					04		000E6	RET		
		18	A6	007E10D0	8F	D0	000E7	3\$: MOVL	#8261840, 24(R6)	0957
					56	DD	000EF	PUSHL	R6	0958
					01	DD	000F1	PUSHL	#1	
		00000000G	00		02	FB	000F3	CALLS	#2, MAIL\$EXTRACT_MESSAGE	
		18	A7	007E105A	8F	D0	000FA	MOVL	#8261722, 24(R7)	0959
				0180	8F	BB	00102	PUSHR	#*M<R7,R8>	0961
			69		02	FB	00106	CALLS	#2, SYS\$CLOSE	
			12		50	E9	00109	4\$: BLBC	STATUS, 5\$	
					5E	DD	0010C	PUSHL	SP	0963
				00000000G	00	9F	0010E	PUSHAB	INPUT_DESC	
					02	FB	00114	CALLS	#2, STR\$COPY_DX	
					01	D0	0011B	MOVL	#1, R0	0964
					04		0011E	5\$: RET		0965

; Routine Size: 287 bytes, Routine Base: \$CODE\$ + 0798

```

: 835      0966 1 ROUTINE DELETE_TMP(FAB) =
: 836      0967 1 |++
: 837      0968 1 | Delete the file that FAB describes
: 838      0969 1 |
: 839      0970 1 |--
: 840      0971 2 BEGIN
: 841      0972 2 MAP
: 842      0973 2     FAB : REF $BBLOCK;
: 843      0974 2
: 844      0975 2 FAB[FAB$V_DLT] = TRUE;
: 845      0976 2 FAB[FAB$SL_CTX] = MAIL$ OPENIN;
: 846      P 0977 2 IF_ERR($OPEN(FAB=.FAB,ERR=UTIL$REPORT_IO_ERROR);,
: 847      0978 2     RETURN .STATUS);
: 848      0979 2 FAB[FAB$SL_CTX] = MAIL$ FILNOTDEL;
: 849      P 0980 2 IF_ERR($CLOSE(FAB=.FAB,ERR=UTIL$REPORT_IO_ERROR);,
: 850      0981 2     RETURN .STATUS);
: 851      0982 2
: 852      0983 2 RETURN TRUE
: 853      0984 1 END;

```

.EXTRN SYSS\$OPEN

```

                                000C 00000 DELETE_TMP:
                                .WORD   Save R2,R3
                                MOVAB   UTIL$REPORT_IO_ERRJR, R3
                                53 0000000G 00 9E 00002
                                MOVL   FAB, R2
                                52 00000000 AC D0 00009
                                05 A2 00000000 8F 88 0000D
                                18 A2 007E109A 8F D0 00012
                                MOVL   #8261786, 24(R2)
                                00000000G 00 0C BB 0001A
                                PUSHR  #^M<R2,R3>
                                17 00000000 02 FB 0001C
                                CALLS  #2, SYSS$OPEN
                                18 A2 007E11F0 8F D0 00026
                                MOVL   #8262128, 24(R2)
                                00000000G 00 0C BB 0002E
                                PUSHR  #^M<R2,R3>
                                03 00000000 02 FB 00030
                                CALLS  #2, SYSS$CLOSE
                                50 E9 00037
                                BLBC   STATUS, 1$
                                50 00000000 01 D0 0003A
                                MOVL   #1, R0
                                04 0003D 1$
                                RET

```

; Routine Size: 62 bytes, Routine Base: \$CODE\$ + 08B7

```

855 0985 1 ROUTINE CALL_EDT_EDIT (FLAGS,FILEIO_RTN) =
856 0986 1 |++
857 0987 1 | FUNCTIONAL DESCRIPTION:
858 0988 1 |
859 0989 1 |         Call EDT to edit the file. Done this way so that if EDT signals
860 0990 1 |         an error, the handler does a default unwind to the caller of this
861 0991 1 |         routine
862 0992 1 | --
863 0993 2 BEGIN
864 0994 2
865 0995 2 LOCAL
866 0996 2     FAB : $FAB_DECL,
867 0997 2     NAM : $NAM_DECL,
868 0998 2     RAB : $RAB_DECL,
869 0999 2     STATUS,
870 1000 2     TMPFAB : $FAB_DECL,
871 1001 2     TMPNAM : $NAM_DECL,
872 1002 2     TMPRESS : $BLOCK[NAM$C-MAXRSS],
873 1003 2     TMPRSS : $BLOCK[NAM$C-MAXRSS],
874 1004 2     NAMBUF : $BLOCK[NAM$C-MAXRSS],
875 1005 2     XLATE_BPV : VECTOR[2, LONG],
876 1006 2     FILEIO_BPV : VECTOR[2, LONG];
877 1007 2
878 1008 2 ENABLE EDT_HANDLER;
879 1009 2
880 1010 2 |
881 1011 2 | Initialize temporary file FAB
882 1012 2 |
883 P 1013 2 $FAB_INIT(fab=tmpfab,
884 P 1014 2     fns=.input_desc[dsc$w_length],
885 P 1015 2     fna=.input_desc[dsc$a_pointer],
886 1016 2     nam=tmpnam);
887 1017 2
888 P 1018 2 $NAM_INIT(NAM=TMPNAM,
889 P 1019 2     ESS=NAM$C-MAXRSS,
890 P 1020 2     ESA=TMPRESS,
891 P 1021 2     RSS=NAM$C-MAXRSS,
892 1022 2     RSA=TMPRSS);
893 1023 2
894 1024 2 IF NOT (status=$PARSE(fab=tmpfab)) THEN
895 1025 2     SIGNAL(MAIL$_OPENIN,1,input_desc,.status);
896 1026 2
897 1027 2 IF .MAIL$GL_FLAGS[MAIF_V_EDTEDT]
898 1028 2 THEN BEGIN
899 1029 3     XLATE_BPV[0] = EDT_XLATE;
900 1030 3     XLATE_BPV[1] = 0;
901 1031 3     IF .FILEIO_RTN EQL 0
902 1032 3     THEN (.EDT_EDIT)(INPUT_DESC,OUTPUT_DESC,
903 1033 3         COMMAND_DESC,JOURNAL_DESC,FLAGS,0,0,XLATE_BPV)
904 1034 4     ELSE BEGIN
905 1035 4         FILEIO_BPV[0] = EDT_EXTRACT_IO;
906 1036 4         FILEIO_BPV[1] = 0;
907 1037 4         (.EDT_EDIT)(INPUT_DESC,OUTPUT_DESC,COMMAND_DESC,JOURNAL_DESC,
908 1038 4             FLAGS,FILEIO_BPV,0,XLATE_BPV);
909 1039 3     END;
910 1040 3 RETURN TRUE
911 1041 2 END;

```

```

: 912      1042  2  :IF .MAIL$GL_FLAGS[MAIF_V_EDTXYZ]
: 913      1043  2  :THEN BEGIN
: 914      1044  2  :   RETURN TRUE
: 915      1045  2  :   END;
: 916      1046  2  :
: 917      1047  2  :   Not an editor we know about.  If reading a mail message, then copy
: 918      1048  2  :   it to a temporary file.
: 919      1049  2  :
: 920      1050  2  :IF .FILEIO_RTN NEQ 0
: 921      1051  2  :THEN BEGIN
: 922      1052  3  :   $NAM INIT(NAM=NAM,
: 923      1053  3  :   RSS=NAM$C MAXRSS,
: 924      1054  3  :   RSA=NAM$B);
: 925      1055  3  :   IF_ERR(COPY_MSG(FAB,RAB,NAM));
: 926      1056  3  :   RETURN TRUE);
: 927      1057  2  :   END;
: 928      1058  2  :   (.EDT EDIT)(INPUT_DESC,OUTPUT_DESC);
: 929      1059  2  :IF .FILEIO_RTN NEQ 0
: 930      1060  2  :THEN DELETE_TMP(FAB);
: 931      1061  2  :RETURN TRUE
: 932      1062  1  :END;

```

01FC 00000 CALL\_EDT\_EDIT:

									WORD	Save R2,R3,R4,R5,R6,R7,R8	: 0985
		56	FB33	CF	9E	00002			MOVAB	EDT_XLATE, R8	
		57	00000000	00	9E	00007			MOVAB	INPUT_DESC, R7	
		5E	FB4C	CE	9E	0000E			MOVAB	-1204(SP), SP	
		6D	010C	CF	DE	00013			MOVAL	7\$, (FP)	: 0993
0050	8F	00		00	2C	00018			MOVCS	#0, (SP), #0, #80, \$RMS_PTR	: 1016
			FEBC	CD		0001F					
			FEBC	CD	5003	8F	B0	00022	MOVW	#20483, \$RMS_PTR	
			FED2	CD		02	90	00029	MOVB	#2, \$RMS_PTR+22	
			FEDB	CD		02	90	0002E	MOVB	#2, \$RMS_PTR+31	
			FEE4	CD	FE5C	CD	9E	00033	MOVAB	TMPNAM, \$RMS_PTR+40	
			FEE8	CD	04	A7	D0	0003A	MOVL	INPUT_DESC+4, \$RMS_PTR+44	
			FEF0	CD		67	90	00040	MOVB	INPUT_DESC, \$RMS_PTR+52	
0060	8F	00		00	2C	00045			MOVCS	#0, (SP), #0, #96, \$RMS_PTR	: 1022
				00	2C	0004C					
			FE5C	CD	6002	8F	B0	0004F	MOVW	#24578, \$RMS_PTR	
			FE5E	CD		01	8E	00056	MNEGB	#1, \$RMS_PTR+2	
			FE60	CD	0110	CE	9E	0005B	MOVAB	TMPRSS, \$RMS_PTR+4	
			FE66	CD		01	8E	00062	MNEGB	#1, \$RMS_PTR+10	
			FE68	CD	0210	CE	9E	00067	MOVAB	TMPRESS, \$RMS_PTR+12	
				CD	FEBC	CD	9F	0006E	PUSHAB	TMPFAB	: 1024
		00000000G		00		01	FB	00072	CALLS	#1, SYSSPARSE	
				13		50	EB	00079	BLBS	STATUS, 1\$	
						50	DD	0007C	PUSHL	STATUS	: 1025
						57	DD	0007E	PUSHL	R7	
						01	DD	00080	PUSHL	#1	
					007E109A	8F	DD	00082	PUSHL	#8261786	
		00000000G		00		04	FB	00088	CALLS	#4, LIB\$SIGNAL	
			3A	00000000G		00	E9	0008F	BLBC	MAIL\$GL_FLAGS+2, 4\$	: 1027
		08	AE			68	9E	00096	MOVAB	EDT_XLATE, XLATE_BPV	: 1029

			0C	AE	D4	0009A	CLRL	XLATE_BPV+4	1030		
		52	4C	A7	D0	0009D	MOVL	EDT_EDIT, R2	1032		
			08	AC	D5	000A1	TSTL	FILEIO_RTN	1031		
			07	12	000A4	BNEQ	2\$				
			08	AE	9F	000A6	PUSHAB	XLATE_BPV	1032		
			7E	7C	000A9	CLRQ	-(SP)				
			10	11	000AB	BRB	3\$				
		6E	0280	C8	9E	000AD	2\$: MOVAB	EDT_EXTRACT_IO, FILEIO_BPV	1035		
			04	AE	D4	000B2	CLRL	FILEIO_BPV+4	1036		
			08	AE	9F	000B5	PUSHAB	XLATE_BPV	1037		
			7E	D4	000B8	CLRL	-(SP)				
			08	AE	9F	000BA	PUSHAB	FILEIO_BPV			
			04	AC	9F	000BD	3\$: PUSHAB	FLAGS			
			10	A7	9F	000C0	PUSHAB	JOURNAL_DESC			
			08	A7	9F	000C3	PUSHAB	COMMAND_DESC			
			18	A7	9F	000C6	PUSHAB	OUTPUT_DESC			
			57	DD	000C9	PUSHL	R7				
		62		08	FB	000CB	CALLS	#8, (R2)			
			4F	11	000CE	BRB	6\$		1040		
			56	D4	000D0	4\$: CLRL	R6		1050		
			08	AC	D5	000D2	TSTL	FILEIO_RTN			
			31	13	000D5	BEQL	5\$				
			56	D6	000D7	INCL	R6				
0060	8F		00	6E	00	2C	000D9	MOVCS	#0, (SP), #0, #96, \$RMS_PTR	1054	
			FF50	CD			000E0				
	FF50	CD	6002	8F	B0	000E3	MOVW	#24578, \$RMS_PTR			
	FF52	CD		01	8E	000EA	MNEGB	#1, \$RMS_PTR+2			
	FF54	CD		10	AE	9E	000EF	MOVAB	NAMBUF, \$RMS_PTR+4		
			FF50	CD	9F	000F5	PUSHAB	NAM		1056	
			FF0C	CD	9F	000F9	PUSHAB	RAB			
			B0	AD	9F	000FD	PUSHAB	FAB			
	036A	C8		03	FB	00100	CALLS	#3, COPY_MSG			
		17		50	E9	00105	BLBC	STATUS, 8\$			
		50	4C	A7	D0	00108	5\$: MOVL	EDT_EDIT, R0	1058		
			18	A7	9F	0010C	PUSHAB	OUTPUT_DESC			
			57	DD	0010F	PUSHL	R7				
		60		02	FB	00111	CALLS	#2, (R0)			
		08		56	E9	00114	BLBC	R6, 6\$	1059		
			B0	AD	9F	00117	PUSHAB	FAB	1060		
	0489	C8		01	FB	0011A	CALLS	#1, DELETE_TMP			
		50		01	D0	0011F	6\$: MOVL	#1, R0	1061		
				04	00122	RET			1062		
				0000	00123	7\$: .WORD	Save nothing		0993		
				7E	D4	00125	CLRL	-(SP)			
				5E	DD	00127	PUSHL	SP			
		7E	04	AC	7D	00129	MOVQ	4(AP), -(SP)			
F8B5	CF			03	FB	0012D	CALLS	#3, EDT_HANDLER			
				04	00132	RET					

; Routine Size: 307 bytes, Routine Base: \$CODE\$ + 08F5



```

: 934 1063 1 GLOBAL ROUTINE MAIL$EDIT_READ_MSG =
: 935 1064 1 ++
: 936 1065 1 FUNCTIONAL DESCRIPTION:
: 937 1066 1
: 938 1067 1 Invoke the editor to read the curent message
: 939 1068 1
: 940 1069 1 --
: 941 1070 2 BEGIN
: 942 1071 2 LOCAL
: 943 1072 2 EDT_FLAGS;
: 944 1073 2
: 945 1074 2 STR$FREE1_DX(INPUT_DESC);
: 946 1075 2 STR$FREE1_DX(OUTPUT_DESC);
: 947 1076 2 STR$FREE1_DX(JOURNAL_DESC);
: 948 1077 2 STR$FREE1_DX(COMMAND_DESC);
: 949 1078 2
P 950 1079 2 IF_ERR(MAIL$LOAD_EDT());
: 951 1080 2 RETURN .STATUS);
: 952 1081 2
: 953 1082 2 IF MAIL$NEW_MSG()
: 954 1083 2 THEN MAIL$DECR_NEWMAIL(MAIL$Q_USERNAME);
: 955 1084 2
: 956 1085 2 EDT_FLAGS = .EDT_M_NOOUTPUT OR .EDT_M_NOCREATE OR .EDT_M_NOJOURNAL;
: 957 1086 2 CALL_EDT_EDIT(.EDT_FLAGS,EDT_EXTRACT_IO);
: 958 1087 2
: 959 1088 2 RETURN TRUE
: 960 1089 1 END;

```

			001C 00000	.ENTRY MAIL\$EDIT_READ_MSG, Save R2,R3,R4	: 1063
	54	F772	CF 9E 00002	MOVAB MAIL\$LOAD_EDT, R4	
	53	00000000G	00 9E 00007	MOVAB STR\$FREE1_DX, R3	
	52	00000000'	00 9E 0000E	MOVAB INPUT_DESC, R2	
			52 DD 00015	PUSHL R2	: 1074
	63		01 FB 00017	CALLS #1, STR\$FREE1_DX	
		18	A2 9F 0001A	PUSHAB OUTPUT_DESC	: 1075
	63		01 FB 0001D	CALLS #1, STR\$FREE1_DX	
		10	A2 9F 00020	PUSHAB JOURNAL_DESC	: 1076
	63		01 FB 00023	CALLS #1, STR\$FREE1_DX	
		08	A2 9F 00026	PUSHAB COMMAND_DESC	: 1077
	63		01 FB 00029	CALLS #1, STR\$FREE1_DX	
	64		00 FB 0002C	CALLS #0, MAIL\$LOAD_EDT	: 1080
	2F		50 E9 0002F	BLBC STATUS, 2\$	
	00000000G	00	00 FB 00032	CALLS #0, MAIL\$NEW_MSG	: 1082
	0D		50 E9 00039	BLBC R0, 1\$	
		00000000G	00 9F 0003C	PUSHAB MAIL\$Q_USERNAME	: 1083
	00000000G	00	01 FB 00042	CALLS #1, MAIL\$DECR_NEWMAIL	
50	2C	A2 34	A2 C9 00049 1\$:	BISL3 EDT_M_NOCREATE, EDT_M_NOOUTPUT, R0	: 1085
		50 28	A2 C8 0004F	BISL2 EDT_M_NOJOURNAL, EDT_FLAGS	: 1086
		050E	C4 9F 00053	PUSHAB EDT_EXTRACT_IO	
			50 DD 00057	PUSHL EDT_FLAGS	: 1086
	0755	C4	02 FB 00059	CALLS #2, CALL_EDT_EDIT	
		50	01 D0 0005E	MOVL #1, R0	: 1088
			04 00061 2\$:	RET	: 1089

MAIL\$EDIT  
V04-000

EDIT OR ENTER A MESSAGE

H 15  
16-Sep-1984 00:51:12  
14-Sep-1984 12:42:17

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[MAIL.SRC]EDIT.B32;1 Page 40  
(17)

; Routine Size: 98 bytes, Routine Base: \$CODES + 0A28

```

: 962      1090  1 GLOBAL ROUTINE MAIL$CMD_EDIT =
: 963      1091  1 !++
: 964      1092  1 ! FUNCTIONAL DESCRIPTION:
: 965      1093  1 !
: 966      1094  1 !         Process the EDIT command by invoking callable EDT to edit the file
: 967      1095  1 !
: 968      1096  1 ! --
: 969      1097  2 BEGIN
: 970      1098  2 LOCAL
: 971      1099  2     STATUS,
: 972      1100  2     EDT_FLAGS;
: 973      1101  2
: 974      1102  2 STR$FREE1_DX(INPUT_DESC);
: 975      1103  2 STR$FREE1_DX(OUTPUT_DESC);
: 976      1104  2 STR$FREE1_DX(JOURNAL_DESC);
: 977      1105  2 STR$FREE1_DX(COMMAND_DESC);
: 978      1106  2
: 979      1107  2 CLISGET_VALUE(SD_INPUT,INPUT_DESC);
: 980      1108  2
: 981      1109  2 IF_ERR(MAIL$LOAD_EDT());
: 982      1110  2     RETURN .STATUS);
: 983      1111  2
: 984      1112  2 EDT_FLAGS = 0;
: 985      1113  2 STATUS = CLISPRESENT(SD_COMMAND);
: 986      1114  2 IF .STATUS
: 987      1115  2     THEN STATUS = CLISGET_VALUE(SD_COMMAND,COMMAND_DESC)
: 988      1116  2     ELSE IF .STATUS EQL CLIS$NEGATED
: 989      1117  2         OR .STATUS EQL CLIS$COCNEG
: 990      1118  2         THEN EDT_FLAGS = .EDT_FLAGS OR .EDT_M_NOCOMMAND;
: 991      1119  2
: 992      1120  2 STATUS = CLISPRESENT(SD_JOURNAL);
: 993      1121  2 IF .STATUS
: 994      1122  2     THEN STATUS = CLISGET_VALUE(SD_JOURNAL,JOURNAL_DESC)
: 995      1123  2     ELSE IF .STATUS EQL CLIS$NEGATED
: 996      1124  2         OR .STATUS EQL CLIS$COCNEG
: 997      1125  2         THEN EDT_FLAGS = .EDT_FLAGS OR .EDT_M_NOJOURNAL;
: 998      1126  2
: 999      1127  2 STATUS = CLISPRESENT(SD_OUTPUT);
: 1000     1128  2 IF .STATUS
: 1001     1129  2     THEN STATUS = CLISGET_VALUE(SD_OUTPUT,OUTPUT_DESC)
: 1002     1130  2     ELSE IF .STATUS EQL CLIS$NEGATED
: 1003     1131  2         OR .STATUS EQL CLIS$COCNEG
: 1004     1132  2         THEN EDT_FLAGS = .EDT_FLAGS OR .EDT_M_NOOUTPUT;
: 1005     1133  2
: 1006     1134  2 STATUS = CLISPRESENT(SD_CREATE);
: 1007     1135  2 IF .STATUS EQL CLIS$NEGATED
: 1008     1136  2     OR .STATUS EQL CLIS$LOCNEG
: 1009     1137  2     THEN EDT_FLAGS = .EDT_FLAGS OR .EDT_M_NOCREATE;
: 1010     1138  2
: 1011     1139  2 IF CLISPRESENT(SD_READ)
: 1012     1140  2     THEN EDT_FLAGS = .EDT_FLAGS OR .EDT_M_NOOUTPUT OR .EDT_M_NOCREATE
: 1013     1141  2         OR .EDT_M_NOJOURNAL;
: 1014     1142  2
: 1015     1143  2 CALL_EDT_EDIT(.EDT_FLAGS,0);
: 1016     1144  2
: 1017     1145  2 RETURN TRUE
: 1018     1146  1 END;
```

			03FC	00000	.ENTRY	MAIL\$CMD_EDIT, Save R2,R3,R4,R5,R6,R7,R8,R9	:	1090
59	00000000G	8F	DO	00002	MOVL	#CLIS_LOCNeg, R9		
58	00000000G	8F	DO	00009	MOVL	#CLIS_NEGATED, R8		
57	00000000G	00	9E	00010	MOVAB	CLISGET VALUE, R7		
56	00000000G	00	9E	00017	MOVAB	STR\$FREE1_DX, R6		
55	00000000G	00	9E	0001E	MOVAB	CLISPRESENT, R5		
54	F659	CF	9E	00025	MOVAB	SD_COMMAND, R4		
53	00000000'	00	9E	0002A	MOVAB	INPUT_DESC, R3		
			53	DD	PUSHL	R3		1102
66		01	FB	00033	CALLS	#1, STR\$FREE1_DX		
	18	A3	9F	00036	PUSHAB	OUTPUT_DESC		1103
66		01	FB	00039	CALLS	#1, STR\$FREE1_DX		
	10	A3	9F	0003C	PUSHAB	JOURNAL_DESC		1104
66		01	FB	0003F	CALLS	#1, STR\$FREE1_DX		
	08	A3	9F	00042	PUSHAB	COMMAND_DESC		1105
66		01	FB	00045	CALLS	#1, STR\$FREE1_DX		
			53	DD	PUSHL	R3		1107
	20	A4	9F	0004A	PUSHAB	SD_INPUT		
67		02	FB	0004D	CALLS	#2, CLISGET_VALUE		
F6C1	CF	00	FB	00050	CALLS	#0, MAIL\$LOAD_EDT		1110
	01	50	E8	00055	BLBS	STATUS, 1\$		
			04	00058	RET			
			52	D4	CLRL	EDT_FLAGS		1112
			54	DD	PUSHL	R4		1113
65		01	FB	0005D	CALLS	#1, CLISPRESENT		
0A		50	E9	00060	BLBC	STATUS, 2\$		1114
	08	A3	9F	00063	PUSHAB	COMMAND_DESC		1115
			54	DD	PUSHL	R4		
67		02	FB	00068	CALLS	#2, CLISGET_VALUE		
		0E	11	0006B	BRB	4\$		
58		50	D1	0006D	Cmpl	STATUS, R8		1116
		05	13	00070	BEQL	3\$		
59		50	D1	00072	Cmpl	STATUS, R9		1117
		04	12	00075	BNEQ	4\$		
52	30	A3	C8	00077	BISL2	EDT_M_NOCOMMAND, EDT_FLAGS		1118
	30	A4	9F	0007B	PUSHAB	SD_JOURNAL		1120
65		01	FB	0007E	CALLS	#1, CLISPRESENT		
0B		50	E9	00081	BLBC	STATUS, 5\$		1121
	10	A3	9F	00084	PUSHAB	JOURNAL_DESC		1122
	30	A4	9F	00087	PUSHAB	SD_JOURNAL		
67		02	FB	0008A	CALLS	#2, CLISGET_VALUE		
		0E	11	0008D	BRB	7\$		
58		50	D1	0008F	Cmpl	STATUS, R8		1123
		05	13	00092	BEQL	6\$		
59		50	D1	00094	Cmpl	STATUS, R9		1124
		04	12	00097	BNEQ	7\$		
52	28	A3	C8	00099	BISL2	EDT_M_NOJOURNAL, EDT_FLAGS		1125
	40	A4	9F	0009D	PUSHAB	SD_OUTPUT		1127
65		01	FB	000A0	CALLS	#1, CLISPRESENT		
0B		50	E9	000A3	BLBC	STATUS, 8\$		1128
	18	A3	9F	000A6	PUSHAB	OUTPUT_DESC		1129
	40	A4	9F	000A9	PUSHAB	SD_OUTPUT		

	67		02	FB	000AC		CALLS	#2, CLISGET_VALUE	:
			0E	11	000Af		BRB	10\$	:
	58		50	D1	000B1	8\$:	CMPL	STATUS, R8	1130
			05	13	000B4		BEQL	9\$	:
	59		50	D1	000B6		CMPL	STATUS, R9	1131
			04	12	000B9		BNEQ	10\$	:
	52	2C	A3	C8	000BB	9\$:	BISL2	EDT_M_NOOUTPUT, EDT_FLAGS	1132
		10	A4	9F	000BF	10\$:	PUSHAB	SD_CREATE	1134
	65		01	FB	000C2		CALLS	#1, CLISPRESENT	:
	58		50	D1	000C5		CMPL	STATUS, R8	1135
			05	13	000C8		BEQL	11\$	:
	59		50	D1	000CA		CMPL	STATUS, R9	1136
			04	12	000CD		BNEQ	12\$	:
	52	34	A3	C8	000CF	11\$:	BISL2	EDT_M_NOCREATE, EDT_FLAGS	1137
		4C	A4	9F	000D3	12\$:	PUSHAB	SD_READ	1139
	65		01	FB	000D6		CALLS	#1, CLISPRESENT	:
	0E		50	E9	000D9		BLBC	RO, 13\$	:
50	52	2C	A3	C9	000DC		BISL3	EDT_M_NOOUTPUT, EDT_FLAGS, RO	1140
	50	34	A3	C8	000E1		BISL2	EDT_M_NOCREATE, RO	:
52	50	28	A3	C9	000E5		BISL3	EDT_M_NOJOURNAL, RO, EDT_FLAGS	1141
			7E	D4	000EA	13\$:	CLRL	-(SP)	1143
			52	DD	000EC		PUSHL	EDT_FLAGS	:
	FD78	CF	02	FB	000EE		CALLS	#2, CALL_EDT_EDIT	:
		50	01	D0	000F3		MOVL	#1, RO	1145
			04	000F6			RET		1146

; Routine Size: 247 bytes, Routine Base: \$CODE\$ + 0A8A

MAIL\$EDIT  
V04-000

EDIT OR ENTER A MESSAGE

L 15  
16-Sep-1984 00:51:12  
14-Sep-1984 12:42:17

VAX-11 Bliss-32 V4.0-742  
DISK\$VM\$MASTER:[MAIL.SRC]EDIT.B32;1

Page 44  
(19)

: 1020 1147 1 END  
: 1021 1148 0 ELUDOM

.EXTRN LIB\$SIGNAL, SYSS\$UNWIND

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	88	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	2945	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
EDT_ADDRESSES	52	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
EDT_NAMES	96	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]STARLET.L32;1	9776	143	1	581	00:00.7
_\$255\$DUA28:[MAIL.OBJ]MAILDEF.L32;1	457	12	2	26	00:00.2

COMMAND QUALIFIERS

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

: Size: 2355 code + 826 data bytes  
: Run Time: 00:30.5  
: Elapsed Time: 01:53.3  
: Lines/CPU Min: 2256  
: Lexemes/CPU-Min: 44018  
: Memory Used: 229 pages  
: Compilation Complete

0228 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

The image displays a dense grid of 100 small terminal window screenshots, arranged in 10 rows and 10 columns. Each window shows a different screen from a VAX/VMS system, likely related to the 'LIS' (List) utilities mentioned in the labels. The screens contain various data formats, including lists, tables, and command-line prompts. Several larger labels are overlaid on the grid, identifying specific screens:

- CMS LIS** (top row, 5th window)
- DIRECTORY LIS** (top row, 8th window)
- EDIT LIS** (middle row, 8th window)
- GETUSER LIS** (middle row, 10th window)
- ADDRLIST LIS** (bottom row, 2nd window)
- GETNEW LIS** (bottom row, 9th window)

Other visible labels include **MAILREQ REQ**, **MAILDEF SDL**, and **SPAWN CMD CLD**. The overall appearance is that of a multi-processor system running various system utilities simultaneously.