


```

1 0001 0 MODULE MAIL$MSGSUBS( %TITLE 'MAIL MESSAGE SUBROUTINES'
2 0002 0 IDENT = 'V04-000' ) =
3 0003 1 BEGIN
4 0004 1
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 * ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 * TRANSFERRED. *
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 * CORPORATION. *
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: VAX/VMS MAIL UTILITY
32 0032 1
33 0033 1 ABSTRACT: Subroutines to read and write message files.
34 0034 1
35 0035 1 ENVIRONMENT: NATIVE/USER MODE
36 0036 1
37 0037 1 AUTHOR: Len Kawell, CREATION DATE: 07-NOV-1979
38 0038 1
39 0039 1 MODIFIED BY:
40 0040 1
41 0041 1 V03-018 BLS0263 Benn Schreiber 5-FEB-1984
42 0042 1 Make routines called by LIB$RENAME global.
43 0043 1
44 0044 1 V03-017 BLS0255 Benn Schreiber 28-Dec-1983
45 0045 1 New global flags.
46 0046 1
47 0047 1 V03-016 BLS0250 Benn Schreiber 9-Dec-1983
48 0048 1 Don't allow multiple versions of MAIL.MAI to have
49 0049 1 mixed file organizations.
50 0050 1
51 0051 1 V03-015 BLS0252 Benn Schreiber 9-Dec-1983
52 0052 1 Look at correct bit for noautopurge.
53 0053 1
54 0054 1 V03-014 BLS0246 Benn Schreiber 28-Nov-1983
55 0055 1 Remove some own storage to shrink image
56 0056 1
57 0057 1 V03-013 BLS0229 Benn Schreiber 16-Jul-1983

```

```

58 0058 1 |
59 0059 1 |
60 0060 1 | V03-012 BLS0227 Benn Schreiber 29-Jun-1983
61 0061 1 | Call MAIL$CLR_DIR_FLAG to clear directory display context
62 0062 1 | when closing mail file.
63 0063 1 |
64 0064 1 | V03-011 BLS0225 Benn Schreiber 17-Jun-1983
65 0065 1 | Enable DFW if MAIL$B_DFW is set. Use UTIL$GETFILENAME
66 0066 1 | for retrieving file specs from fabs.
67 0067 1 |
68 0068 1 | V03-010 BLS0222 Benn Schreiber 15-May-1983
69 0069 1 | Small corrections.
70 0070 1 |
71 0071 1 | V03-009 BLS0221 Benn Schreiber 5-May-1983
72 0072 1 | Correct SYSPRV handling
73 0073 1 |
74 0074 1 | V03-008 BLS0214 Benn Schreiber 27-Mar-1983
75 0075 1 | Modify new message handling. Correct nam block
76 0076 1 | usage error in open_new_file
77 0077 1 |
78 0078 1 | V03-007 BLS0211 Benn Schreiber 13-Mar-1983
79 0079 1 | Extensive rework for ISAM mail files
80 0080 1 |
81 0081 1 | V03-006 BLS0197 Benn Schreiber 9-Dec-1982
82 0082 1 | Implement SET PERSONAL_NAME command.
83 0083 1 |
84 0084 1 | V03-005 BLS0191 Benn Schreiber 27-Oct-1982
85 0085 1 | Use system default MBC. Check errors in MAIL$SEND_MSG
86 0086 1 | better. Do not write <FF><From: name> from the message
87 0087 1 | text. This is a new message marker.
88 0088 1 |
89 0089 1 | V03-004 BLS0189 Benn Schreiber 19-Oct-1982
90 0090 1 | Clean up file data initialization. Interlock the
91 0091 1 | DELETE mail function with the READ function, so that
92 0092 1 | a mail message that has not been read cannot be deleted.
93 0093 1 |
94 0094 1 | V03-003 BLS0188 Benn Schreiber 4-Oct-1982
95 0095 1 | Correct problem in directory listing with hyphenated
96 0096 1 | names
97 0097 1 |
98 0098 1 | V03-002 BLS0184 Benn Schreiber 20-Aug-1982
99 0099 1 | Delete old MAIL.TMP's when starting up. This prevents
100 0100 1 | aborting the sending of a message and having some old
101 0101 1 | garbage being sent. Correct error in info_msg which
102 0102 1 | caused accvio if no date on "from" line.
103 0103 1 |
104 0104 1 | V03-001 BLS0177 Benn Schreiber 21-Jun-1982
105 0105 1 | Add MAIL$Q_MSG_FNM descriptor which describes the
106 0106 1 | filename part of the message file file spec.
107 0107 1 |

```

```

109 0108 1 |
110 0109 1 | TABLE OF CONTENTS:
111 0110 1 |
112 0111 1 |
113 0112 1 | INCLUDE FILES:
114 0113 1 |
115 0114 1 |     LIBRARY 'SYSS$LIBRARY:STARLET';
116 0115 1 |     REQUIRE 'SR$:MAILREQ';
117 0261 1 |     LIBRARY 'LIB$:MAILDEF';
118 0262 1 |
119 0263 1 | LITERAL
120 0264 1 |     LEAVE_SEQ = 0;           !0 >> convert to isam
121 0265 1 |
122 0266 1 | FORWARD ROUTINE
123 0267 1 |     MAIL$CLOSE FILE: NOVALUE,      ! Close current message file
124 0268 1 |     MAIL$REWIND MSG,                ! Rewind current message
125 0269 1 |     OPEN_NEWFILE;                  ! Open new messages file
126 0270 1 |
127 0271 1 | MACROS:
128 0272 1 |
129 0273 1 | MACRO
130 M 0274 1 |     RECSIZ(RECPTR) =                ! Get message record size
131 0275 1 |         (RECPTR)<0,RECSIZ_SIZ * 8>%;
132 0276 1 |
133 M 0277 1 | MACRO     MSGDELIM(RECPTR) =        ! Check for message delimiter
134 M 0278 1 |         ((.RECSIZ(RECPTR) EQL 1)
135 M 0279 1 |         AND (CH$RCHAR(RECPTR + RECSIZ_SIZ) EQL FF)
136 M 0280 1 |         AND (.RECSIZ(RECPTR + RECSIZ_SIZ + 1) GTRU %CHARCOUNT('From'))
137 0281 1 |         AND (.RECPTR + 2*RECSIZ_SIZ + 1) EQL 'From'))%;
138 0282 1 |
139 M 0283 1 | MACRO     NEWMSGFLAG(MSGPTR) =      ! Get new message flag address
140 M 0284 1 |         (LOCAL PTR;
141 M 0285 1 |         PTR = MSGPTR + RECSIZ_SIZ + .RECSIZ(MSGPTR); ! Skip delimiter
142 0286 1 |         .PTR + RECSIZ_SIZ + .RECSIZ(.PTR) - 1)%; ! Addr of last char
143 0287 1 |
144 0288 1 |
145 0289 1 | EQUATED SYMBOLS:
146 0290 1 |
147 0291 1 | LITERAL
148 0292 1 |     FROM_SIZ      = %CHARCOUNT('From: '), ! Size of from delimiter
149 0293 1 |     TO_SIZ        = %CHARCOUNT('To: '),   ! Size of 'to' delimiter
150 0294 1 |     MAXRECSIZ     = 255,                  ! Maximum message record size
151 0295 1 |     RECSIZ_SIZ    = 1,                    ! Message buffer record size
152 0296 1 |     SUBJ_SIZ      = %CHARCOUNT('Subj: '), ! Size of subject delimiter
153 0297 1 |
154 0298 1 |
155 0299 1 | MACRO
156 0300 1 |     DEFFILENAM    = 'MAIL%',              ! Default message file name
157 0301 1 |     DEFFILETYP    = '.MAI%';             ! Default message file type
158 0302 1 |
159 0303 1 | EXTERNAL REFERENCES:
160 0304 1 |
161 0305 1 | EXTERNAL
162 0306 1 |     MAIL$GL_FLAGS : $BBLOCK,             ! Mail control flags
163 0307 1 |     MAIL$SD_NEWMAIL : $BBLOCK,           ! Descriptor for 'NEWMAIL'
164 0308 1 |     MAIL$L_CURMSG : REF $BBLOCK,         ! Descriptor of current ISAM msg
165 0309 1 |     MAIL$L_SYSMAIBUF : REF $BBLOCK,      ! Pointer to VMSMAIL record

```

```
166 0310 1 MAIL$L_USRSYSBUF : REF $BBLOCK,      ! User's vmsmail record
167 0311 1 MAIL$B_DFW : BYTE,                    ! True if FOP=DFW
168 0312 1 MAIL$Q_TMPFILE_DESC : $BBLOCK,    ! Descriptor for MAIL.TMP
169 0313 1 MAIL$Q_USERNAME : BLOCK[,BYTE],   ! User's name descriptor
170 0314 1                                     ! descriptor
171 0315 1 MAIL$Q_MAILDEF : $BBLOCK;        ! 'dev:[dir]mail.mai'
172 0316 1
173 0317 1 EXTERNAL ROUTINE
174 0318 1 MAIL$CLR_DIR_FLAG,                ! Clear directory display flags
175 0319 1 MAIL$COPY_MSGS,                  ! Copy msgs into ISAM folder
176 0320 1 MAIL$DESELECT_MSGS,             ! Delete in-memory index
177 0321 1 MAIL$EMPTY_WASTEBASKET,         ! Delete msgs in wastebasket
178 0322 1 MAIL$GET_INPUT,                 ! Read from the terminal
179 0323 1 MAIL$GETMSG,                    ! Get message text
180 0324 1 MAIL$ISAM_INIT,                 ! Init ISAM data
181 0325 1 MAIL$UPDATE_I_MSG,              ! Update ISAM message record
182 0326 1 MAIL$UP_PRIV,                   ! Raise privileges
183 0327 1 MAIL$DOWN_PRIV,                 ! Lower privileges
184 0328 1 LIB$FILE_SCAN,                  ! Perform wild-card file scan
185 0329 1 LIB$FID_TO_NAME,                ! Convert fid to name
186 0330 1 LIB$FREE_VM,                    ! Free virtual memory
187 0331 1 LIB$GET_VM,                     ! Get virtual memory
188 0332 1 LIB$RENAME_FILE,                 ! Rename a file
189 0333 1 STR$COPY_DX,                     ! String copy
190 0334 1 STR$FREE_DX,                     ! Deallocate dynamic string
191 0335 1 OTSS$COPY_R_DX,                  ! String copy
192 0336 1 UTIL$GETFILENAME,                ! Get file spec for fab
193 0337 1 UTIL$REPORT_IO_ERROR;           ! Report I/O error
194 0338 1
195 0339 1 ! OWN STORAGE:
196 0340 1
197 0341 1 OWN
198 0342 1 EXITSTATUS: LONG,                ! Exit status
199 0343 1 EXITBLOCK: VECTOR[4]           ! Exit handler control block
200 0344 1 INITIAL(0,                      ! Forward link
201 0345 1 MAIL$CLOSE_FILE,                ! Routine address
202 0346 1 1,                               ! Argument count
203 0347 1 EXITSTATUS),                    ! Location to receive status
204 0348 1 MODIFIED: LONG;                 ! Buffer modified flag
205 0349 1
206 0350 1 !
207 0351 1 ! Current message file being read RMS blocks
208 0352 1 !
209 0353 1 OWN
210 0354 1 DYN_DESC : DYN_DESC_DECL,        ! Dynamic descriptor
211 0355 1 RD_RSNAM_BUF : REF VECTOR[BYTE], ! Resultant name string buf
212 0356 1 RD_ESNAM_BUF : REF VECTOR[BYTE], ! Expanded name string buffer
213 0357 1 RD_ESNAM_DESC : DYN_DESC_DECL,   ! Expanded name string desc
214 0358 1 RD_NAM:                          ! Name block
215 0359 1 $NAM_DECL,
216 0360 1 RD_FAB:                          ! File access block
217 0361 1 $FAB_DECL,
218 0362 1 RD_RAB:                          ! Record access block
219 0363 1 $RAB_DECL;
220 0364 1
221 0365 1 GLOBAL BIND
222 0366 1 MAIL$T_RD_RAB = RD_RAB,
```

```
.. 223 0367 1 MAIL$Q_RD_DESC = RD_ESNAM_DESC;
.. 224 0368 1
.. 225 0369 1 : Current message buffer parameters
.. 226 0370 1
.. 227 0371 1 OWN
.. 228 0372 1 MSG_BUF_ASIZ: LONG, : Message buffer size allocated
.. 229 0373 1 MSG_BUF_ADR: LONG, : Message buffer address
.. 230 0374 1 MSG_BUF_SIZ: LONG, : Message buffer size
.. 231 0375 1 MSG_ADR: LONG; : Message address
.. 232 0376 1
.. 233 0377 1 GLOBAL BIND
.. 234 0378 1 MAIL$L_MSGBUFSIZ = MSG_BUF_SIZ; : Make available externally
.. 235 0379 1 GLOBAL
.. 236 0380 1 MAIL$L_RECBUF : REF $BBLOCK, : Pointer to isam buffer
.. 237 0381 1 MAIL$Q_MSG_FILE: DYN DESC DECL, : Message file name desc
.. 238 0382 1 MAIL$Q_MSG_FNM: BLOCK[8,BYTE], : Message file filename descr.
.. 239 0383 1 MAIL$L_MSG_STS: LONG : Message status
.. 240 0384 1 INITIAL(MAIL$_NOTREADIN)
.. 241 0385 1 MAIL$L_MSG_NUM: LONG, : Message number
.. 242 0386 1 MAIL$W_MSG_RSIZ:WORD, : Message record size
.. 243 0387 1 MAIL$L_MSG_RADR:LONG; : Message record address
.. 244 0388 1
.. 245 0389 1 : Send/File message RMS blocks
.. 246 0390 1
.. 247 0391 1 GLOBAL BIND
.. 248 0392 1 MAIL$Q_MSGTYPE = DESCRIPTOR(DEFFILETYP);! Default mail message filetype
.. 249 0393 1 MAIL$Q_MSGNAME = DESCRIPTOR(DEFFILENAM);! Default mail message filename
.. 250 0394 1 BLOCK[,BYTE];
.. 251 0395 1
.. 252 0396 1 : Define shared messages for MAIL
.. 253 0397 1
.. 254 P 0398 1 $SHR_MSGDEF(MAIL,126,LOCAL,
.. 255 P 0399 1 (RENAMED,SUCCESS),
.. 256 P 0400 1 (SEARCHFAIL,ERROR),
.. 257 P 0401 1 (OPENIN,ERROR),
.. 258 P 0402 1 (READERR,ERROR),
.. 259 P 0403 1 (WRITEERR,ERROR),
.. 260 0404 1 (CLOSEOUT,ERROR));
```

```
262 0405 1 GLOBAL ROUTINE MAIL$FIND_MSG(MSGNUM,NEW_FILE) =
263 0406 1 |++
264 0407 1 | FUNCTIONAL DESCRIPTION:
265 0408 1 |
266 0409 1 |     This routine finds the specified message and sets the current message
267 0410 1 |     pointers to describe it.
268 0411 1 |
269 0412 1 | INPUTS:
270 0413 1 |
271 0414 1 |     MSGNUM = Relative message number desired.
272 0415 1 |     = Zero if next message is desired.
273 0416 1 |     NEW_FILE = true if new file just opened.  Need not be specified
274 0417 1 |               if false.
275 0418 1 |
276 0419 1 | OUTPUTS:
277 0420 1 |
278 0421 1 |     MAIL$MSG_NUM = Current message number.
279 0422 1 |     MAIL$MSG_ADR = Address of first record in message.
280 0423 1 |     MAIL$MSG_STS = Status of current message.
281 0424 1 |
282 0425 1 |     All failures are signalled as well as returned.
283 0426 1 | --
284 0427 2 BEGIN
285 0428 2 BUILTIN
286 0429 2     ACTUALCOUNT;
287 0430 2
288 0431 2 LOCAL
289 0432 2     SRCH MSGNUM,                ! Message number to search for
290 0433 2     POINTER;                    ! Message buffer pointer
291 0434 2
292 0435 2 MAIL$GL_FLAGS[MAIF V READ_FL] = 0;
293 0436 2 IF ACTUALCOUNT() GEQ 2
294 0437 2     THEN IF .NEW_FILE
295 0438 2         AND .MSGNUM EQL 0
296 0439 3     THEN BEGIN
297 0440 3         MAIL$MSG_STS = STS$K_SUCCESS;
298 0441 3         MAIL$REWIND_MSG();
299 0442 3         RETURN TRUE
300 0443 2     END;
301 0444 2 |
302 0445 2 | If specified message number is zero, then searching for next message,
303 0446 2 | else, searching for specified message number.
304 0447 2 |
305 0448 2 SRCH MSGNUM = (IF .MSGNUM EQL 0 THEN (.MAIL$MSG_NUM + 1) ELSE .MSGNUM);
306 0449 2 IF .MAIL$MSG_NUM EQL 0
307 0450 2     THEN MAIL$MSG_NUM = 1;
308 0451 2 |
309 0452 2 | Check that a message file is open and there are messages in the file.
310 0453 2 |
311 0454 2 IF .MSG_BUF_ADR EQL 0
312 0455 2 THEN
313 0456 2     (SIGNAL (MAIL$_NOTREADIN); RETURN MAIL$_NOTREADIN);
314 0457 2 IF .MSG_BUF_SIZ EQL 0
315 0458 2 THEN
316 0459 2     (SIGNAL (MAIL$_NOMOREMSG); RETURN MAIL$_NOMOREMSG);
317 0460 2 |
318 0461 2 | If the specified message number is less than the current message number
```



```

319 0462 2 | or there is no current message, then set the message file pointers to
320 0463 2 | the beginning of the buffer, so the message search will start over again.
321 0464 2 |
322 0465 3 IF (.SRCH_MSGNUM LSSU .MAIL$ _MSG_NUM) | If number less than current
323 0466 3 OR (.MSG_ADR EQL 0) | or, no current message
324 0467 3 THEN BEGIN
325 0468 3 MAIL$ _MSG_NUM = 1; | Re-init current number
326 0469 3 MSG_ADR = .MSG_BUF_ADR; | Re-init current pointer
327 0470 3 END;
328 0471 2 |
329 0472 2 | Search through the messages until the relative message number equals
330 0473 2 | the specified message number or the end of the buffer is reached.
331 0474 2 |
332 0475 2 POINTER = .MSG_ADR; | Start at current message
333 0476 3 UNTIL (.MAIL$ _MSG_NUM GEQU .SRCH_MSGNUM)
334 0477 4 OR ((POINTER = .POINTER + .RECSIZ(.POINTER) + RECSIZ_SIZ)
335 0478 3 GEQU (.MSG_BUF_ADR + .MSG_BUF_SIZ))
336 0479 3 DO
337 0480 4 IF MSGDELIM(.POINTER) | If message delimiter
338 0481 4 THEN BEGIN | then, found new message
339 0482 4 MSG_ADR = .POINTER;
340 0483 4 MAIL$ _MSG_NUM = .MAIL$ _MSG_NUM + 1;
341 0484 3 END;
342 0485 2 END;
343 0486 2 |
344 0487 2 | Set the current record context to the first text record in the message.
345 0488 2 |
346 0489 2 MAIL$ _MSG_RADR = .MSG_ADR + 2*RECSIZ_SIZ + .RECSIZ(.MSG_ADR); | Skip delimiter
347 0490 2 MAIL$ _MSG_RSIZ = .RECSIZ(.MAIL$ _MSG_RADR - RECSIZ_SIZ);
348 0491 2 |
349 0492 2 | If the next message was requested and no more were found, return 'no more'.
350 0493 2 |
351 0494 3 IF (.MSGNUM EQL 0) AND (.MAIL$ _MSG_NUM LSSU .SRCH_MSGNUM)
352 0495 2 THEN
353 0496 2 MAIL$ _MSG_STS = MAIL$ _NOMOREMSG
354 0497 2 ELSE
355 0498 2 MAIL$ _MSG_STS = STS$K_SUCCESS;
356 0499 2 RETURN .MAIL$ _MSG_STS
357 0500 1 END;

```

.TITLE MAIL\$MSGSUBS MAIL MESSAGE SUBROUTINES
.IDENT \V04-000\

.PSECT \$CODE\$,NOWRT,2

```

49 41 4D 2E 00000 P.AAB: .ASCII \.MAIL\
      00000004 C0004 P.AAA: .LONG 4
      00000000' 00008 .ADDRESS P.AAB
4C 49 41 4D 0000C P.AAD: .ASCII \MAIL\
      00000004 00010 P.AAC: .LONG 4
      00000000' 00014 .ADDRESS P.AAD

```

.PSECT \$OWNS\$,NOEXE,2

00000 EXITSTATUS:
.BLKB 4

```

00000000 00004 EXITBLOCK:
                                .LONG 0
00000000V 00008 .ADDRESS MAIL$CLOSE_FILE
00000001 0000C .LONG 1
00000000' 00010 .ADDRESS EXITSTATUS
                                00014 MODIFIED:
                                .BLKB 4
                                00# 00018 DYN_DESC:
                                .BYTE 0[2]
02 OE 0001A .BYTE 14, 2
                                0001C .BLKB 4
                                00020 RD_RSNAM_BUF:
                                .BLKB 4
                                00024 RD_ESNAM_BUF:
                                .BLKB 4
                                00# 00028 RD_ESNAM_DESC:
                                .BYTE 0[2]
02 OE 0002A .BYTE 14, 2
                                0002C .BLKB 4
                                00030 RD_NAM: .BLKB 96
                                00090 RD_FAB: .BLKB 80
                                000E0 RD_RAB: .BLKB 68
                                00124 MSG_BUF_ASIZ:
                                .BLKB 4
                                00128 MSG_BUF_ADR:
                                .BLKB 4
                                0012C MSG_BUF_SIZ:
                                .BLKB 4
                                00130 MSG_ADR: .BLKB 4

                                .PSECT $GLOBAL$,NOEXE,2

00000 MAIL$_RECBUF::
                                .BLKB 4
                                00# 00004 MAIL$_MSG_FILE:
                                .BYTE 0[2]
02 OE 00006 .BYTE 14, 2
                                00008 .BLKB 4
                                0000C MAIL$_MSG_FNM::
                                .BLKB 8
007E8022 00014 MAIL$_MSG_STS::
                                .LONG 8290338
                                00018 MAIL$_MSG_NUM::
                                .BLKB 4
                                0001C MAIL$_MSG_RSIZ::
                                .BLKB 2
                                0001E .BLKB 2
                                00020 MAIL$_MSG_RADR::
                                .BLKB 4

MAIL$_RD_RAB== RD_RAB
MAIL$_RD_DESC== RD_ESNAM_DESC
MAIL$_MSGBUFSIZ== MSG_BUF_SIZ
MAIL$_MSGTYPE== P.AAA
MAIL$_MSGNAME== P.AAC
                                .EXTRN MAIL$_GL_FLAGS, MAIL$_SD_NEWMAIL
                                .EXTRN MAIL$_CURMSG, MAIL$_SYSMAILBUF

```

```
.EXTRN MAIL$L_USRSYSBUF
.EXTRN MAIL$B_DFW, MAIL$Q_TMPFILE_DESC
.EXTRN MAIL$Q_USERNAME
.EXTRN MAIL$Q_MAILDEF, MAIL$CLR_DIR_FLAG
.EXTRN MAIL$COPY_MSGS, MAIL$DESELECT_MSGS
.EXTRN MAIL$EMPTY_WASTEBASKET
.EXTRN MAIL$GET_INPUT, MAIL$GETMSG
.EXTRN MAIL$ISAM_INIT, MAIL$UPDATE_I_MSG
.EXTRN MAIL$UP_PRIV, MAIL$DOWN_PRIV
.EXTRN LIB$FILE_SCAN, LIB$FID_TO_NAME
.EXTRN LIB$FREE_VM, LIB$GET_VM
.EXTRN LIB$RENAME_FILE
.EXTRN STR$COPY_DX, STR$FREE1_DX
.EXTRN OTS$SCOPY_R_DX, UTIL$GETFILENAME
.EXTRN UTIL$REPORT_IO_ERROR
```

.PSECT \$CODE\$,NOWRT,2

			007C	00000		.ENTRY	MAIL\$FIND MSG, Save R2,R3,R4,R5,R6		0405
	56	00000000G	00	9E	00002	MOVAB	LIB\$SIGNAC, R6		
	55	00000000'	00	9E	00009	MOVAB	MSG_ADR, R5		
	54	00000000'	00	9E	00010	MOVAB	MAIL\$L_MSG_NUM, R4		
	00000000G		00	8F	8A	BICB2	#64, MAIL\$GL_FLAGS		0435
			02	6C	91	CMPB	(AP), #2		0436
				18	1F	BLSSU	1\$		
	14		08	AC	E9	BLBC	NEW FILE, 1\$		0437
				04	AC	TSTL	MSGNUM		0438
				0F	12	BNEQ	1\$		
	FC	A4		01	D0	MOVL	#1, MAIL\$L_MSG_STS		0440
	00000000V		00	00	FB	CALLS	#0, MAIL\$REWIND_MSG		0441
			50	01	D0	MOVL	#1, R0		0442
					04	RET			
				53	D4	CLRL	R3		0448
			04	AC	D5	TSTL	MSGNUM		
				08	12	BNEQ	2\$		
52				53	D6	INCL	R3		
	64		01	C1	00045	ADDL3	#1, MAIL\$L_MSG_NUM, SRCH_MSGNUM		
			04	11	00049	BRB	3\$		
	52		04	AC	D0	MOVL	MSGNUM, SRCH_MSGNUM		
				64	D5	TSTL	MAIL\$L_MSG_NUM		0449
				03	12	BNEQ	4\$		
	64		01	D0	00053	MOVL	#1, MAIL\$L_MSG_NUM		0450
				F8	A5	TSTL	MSG_BUF_ADR		0454
				11	12	BNEQ	5\$		
		007E8022		8F	DD	PUSHL	#8290338		0456
	66		01	FB	00061	CALLS	#1, LIB\$SIGNAL		
	50	007E8022		8F	D0	MOVL	#8290338, R0		
					04	RET			
				FC	A5	TSTL	MSG_BUF_SIZ		0457
				11	12	BNEQ	6\$		
		007E8042		8F	DD	PUSHL	#8290370		0459
	66		01	FB	00077	CALLS	#1, LIB\$SIGNAL		
	50	007E8042		8F	D0	MOVL	#8290370, R0		
					04	RET			
	64		52	D1	00082	CMPB	SRCH_MSGNUM, MAIL\$L_MSG_NUM		0465
				04	1F	BLSSU	7\$		
				65	D5	TSTL	MSG_ADR		0466

			07	12	00087		BNEQ	8\$					
	64		01	D0	00088	7\$:	MOVL	#1, MAIL\$ MSG_NUM		0468			
	65	F8	A5	D0	0008E		MOVL	MSG_BUF_ADR, MSG_ADR		0469			
	50		65	D0	00092	8\$:	MOVL	MSG_ADR, POINTER		0475			
	52		64	D1	00095	9\$:	CMPL	MAIL\$ MSG_NUM, SRCH_MSGNUM		0476			
			35	1E	00098		BGEQU	10\$					
	51		60	9A	0009A		MOVZBL	(POINTER), R1		0477			
	50		01	A140	9E	0009D	MOVAB	1(R1)[POINTER], POINTER					
51	F8		FC	A5	C1	000A2	ADDL3	MSG_BUF_SIZ, MSG_BUF_ADR, R1		0478			
				50	D1	000A8	CMPL	POINTER, R1					
				22	1E	000AB	BGEQU	10\$					
				01	60	91	000AD	CMPB	(POINTER), #1		0480		
					E3	12	000B0	BNEQ	9\$				
				0C	01	A0	91	000B2	CMPB	1(POINTER), #12			
					DD	12	000B6	BNEQ	9\$				
				04	02	A0	91	000B8	CMPB	2(POINTER), #4			
					D7	1B	000BC	BLEQU	9\$				
	6D6F7246			8F	03	A0	D1	000BE	CMPL	3(POINTER), #1836020294			
					CD	12	000C6	BNEQ	9\$				
				65	50	D0	000C8	MOVL	POINTER, MSG_ADR		0482		
					64	D6	000CB	INCL	MAIL\$ MSG_NUM		0483		
					C6	11	000CD	BRB	9\$		0476		
				50	65	D0	000CF	10\$:	MOVL	MSG_ADR, R0	0489		
				51	60	9A	000D2	MOVZBL	(R0), R1				
	08			02	A140	9E	000D5	MOVAB	2(R1)[R0], MAIL\$ MSG_RADR				
				08	A4	D0	000DB	MOVL	MAIL\$ MSG_RADR, R0		0490		
	04			FF	A0	9B	000DF	MOVZBW	-1(R0), MAIL\$ MSG_RSIZ				
					53	E9	000E4	BLBC	R3, 11\$		0494		
				52	64	D1	000E7	CMPL	MAIL\$ MSG_NUM, SRCH_MSGNUM				
					0A	1E	000EA	BGEQU	11\$				
				FC	A4	007E8042	6F	D0	000EC	MOVL	#8290370, MAIL\$ MSG_STS	0496	
					04	11	000F4	BRB	12\$				
				FC	A4	01	D0	000F6	11\$:	MOVL	#1, MAIL\$ MSG_STS	0498	
					50	FC	A4	D0	000FA	12\$:	MOVL	MAIL\$ MSG_STS, R0	0499
					04	000FE		RET		0500			

: Routine Size: 255 bytes. Routine Base: \$CODE\$ + 0018

: 358 0501 1

```

360 0502 1 GLOBAL ROUTINE MAIL$GET_MSG =
361 0503 1  ++
362 0504 1  FUNCTIONAL DESCRIPTION:
363 0505 1
364 0506 1      This routine finds the next record in the current message and sets
365 0507 1      the size and address parameters to describe it.
366 0508 1
367 0509 1  INPUTS:
368 0510 1
369 0511 1      MAIL$W_MSG_RSIZ = Current message record size.
370 0512 1      MAIL$L_MSG_RADR = Current message record address.
371 0513 1
372 0514 1  OUTPUTS:
373 0515 1
374 0516 1      MAIL$W_MSG_RSIZ = Next message record size.
375 0517 1      MAIL$L_MSG_RADR = Next message record address.
376 0518 1      MAIL$L_MSG_STS = ST$SK_SUCCESS if another record exists.
377 0519 1      = Zero if no more records in message.
378 0520 1  --
379 0521 2 BEGIN
380 0522 2
381 0523 2      Set pointer and size to next record, unless next record is past
382 0524 2      the end of the buffer or is another message delimiter.
383 0525 2
384 0526 2 MAIL$L_MSG_RADR = .MAIL$L_MSG_RADR + .MAIL$W_MSG_RSIZ + RECSIZ_SIZ;
385 0527 3 IF .MAIL$L_MSG_RADR GTRU T.MSG_BUF_ADR + .MSG_BUF_SIZ)
386 0528 3 OR MSGDELIM(.MAIL$L_MSG_RADR - RECSIZ_SIZ)
387 0529 2 THEN
388 0530 2     MAIL$L_MSG_STS = 0
389 0531 3 ELSE BEGIN
390 0532 3     MAIL$W_MSG_RSIZ = .RECSIZ(.MAIL$L_MSG_RADR - RECSIZ_SIZ);
391 0533 3     MAIL$L_MSG_STS = ST$SK_SUCCESS;
392 0534 2     END;
393 0535 2 RETURN .MAIL$L_MSG_STS
394 0536 1 END;

```

			0004 0000	.ENTRY MAIL\$GET MSG, Save R2	: 0502
	52 00000000'	00	9E 00002	MOVAB MAIL\$L_MSG_RADR, R2	: 0526
	50 FC	A2	3C 00009	MOVZWL MAIL\$W_MSG_RSIZ, R0	
	50	62	C0 0000D	ADDL2 MAIL\$L_MSG_RADR, R0	
	62 01	A0	9E 00010	MOVAB 1(R0), .MAIL\$L_MSG_RADR	
	51	62	D0 00014	MOVL MAIL\$L_MSG_RADR, R1	: 0527
	50 00000000'	00	C1 00017	ADDL3 MSG_BUF_SIZ, MSG_BUF_ADR, R0	
	50	51	D1 00023	CMP R1, R0	
		1F	1A 00026	BGTRU 1\$	
	50 FF	A1	9E 00028	MOVAB -1(R1), R0	: 0528
	01	60	91 0002C	CMPB (R0), #1	
		1B	12 0002F	BNEQ 2\$	
	0C 01	A0	91 00031	CMPB 1(R0), #12	
		15	12 00035	BNEQ 2\$	
	04 02	A0	91 00037	CMPB 2(R0), #4	
		0F	1B 0003B	BLEQU 2\$	
	6D6F7246 8F	03	A0 D1 0003D	CMP 3(R0), #1836020294	


```

397 0538 1 GLOBAL ROUTINE MAIL$REWIND_MSG =
398 0539 1 ++
399 0540 1 FUNCTIONAL DESCRIPTION:
400 0541 1
401 0542 1 This routine "rewinds" the current message by resetting the current
402 0543 1 record pointers to the first text record in the message.
403 0544 1
404 0545 1 INPUTS:
405 0546 1
406 0547 1 MSG_ADR = Address of first record in current message.
407 0548 1
408 0549 1 OUTPUTS:
409 0550 1
410 0551 1 MAIL$W_MSG_RSIZ = Size of first text record in current message.
411 0552 1 MAIL$L_MSG_RADR = Address of first text record in current message.
412 0553 1 --
413 0554 2 BEGIN
414 0555 3 IF (NOT .MAIL$L_MSG_STS AND .MAIL$L_MSG_STS NEQ 0)
415 0556 2 OR .MSG_ADR EQL 0
416 0557 2 THEN
417 0558 2 RETURN MAIL$ NOTREADIN;
418 0559 2 MAIL$L_MSG_RADR = .MSG_ADR + 2*RECSIZ_SIZ + .RECSIZ(.MSG_ADR); ! Skip delimiter
419 0560 2 MAIL$W_MSG_RSIZ = .RECSIZ(.MAIL$L_MSG_RADR - RECSIZ_SIZ);
420 0561 2 MAIL$L_MSG_STS = STS$K_SUCCESS;
421 0562 2 RETURN TRUE;
422 0563 1 END;

```

			000C 00000	.ENTRY	MAIL\$REWIND_MSG, Save R2,R3	: 0538
	53	00000000'	00 9E 00002	MOVAB	MSG_ADR, R3	:
	52	00000000'	00 9E 00009	MOVAB	MAIL\$L_MSG_STS, R2	:
	50		62 D0 00010	MOVL	MAIL\$L_MSG_STS, R0	: 0555
	02		50 E8 00013	BLBS	R0, 1\$:
			04 12 00016	BNEQ	2\$:
			63 D5 00018	TSTL	MSG_ADR	: 0556
			08 12 0001A	BNEQ	3\$:
	50	007E8022	8F D0 0001C	MOVL	#8290338, R0	: 0558
			04 00023	RET		:
	51		63 D0 00024	MOVL	MSG_ADR, R1	: 0559
	50		61 9A 00027	MOVZBL	(R1), R0	:
0C	A2	02 A041	9E 0002A	MOVAB	2(R0)[R1], MAIL\$L_MSG_RADR	:
	50	0C A2	D0 00030	MOVL	MAIL\$L_MSG_RADR, R0	: 0560
08	A2	FF A0	9B 00034	MOVZBW	-1(R0), MAIL\$W_MSG_RSIZ	:
	62		01 D0 00039	MOVL	#1, MAIL\$L_MSG_STS	: 0561
	50		01 D0 0003C	MOVL	#1, R0	: 0562
			04 0003F	RET		: 0563

: Routine Size: 64 bytes, Routine Base: \$CODE\$ + 0171

: 423 0564 1

```

: 425 0565 1 GLOBAL ROUTINE MAIL$INFO_MSG(SENDER,SUBJ,DATE,TO_DESC, MESSAGE_SIZE) =
: 426 0566 1 **
: 427 0567 1 FUNCTIONAL DESCRIPTION:
: 428 0568 1
: 429 0569 1 This routine is called to get the sender, date/time, subject,
: 430 0570 1 and size of the current message.
: 431 0571 1
: 432 0572 1 INPUTS:
: 433 0573 1
: 434 0574 1 SENDER = Address of descriptor of buffer to receive message sender.
: 435 0575 1 SUBJ = Address of descriptor of buffer to receive message subject.
: 436 0576 1 DATE = Optional address of descriptor of buffer to receive date/time
: 437 0577 1 of message.
: 438 0578 1 TO_DESC = Optional address of descriptor for 'to' list
: 439 0579 1 MESSAGE_SIZE = Optional address of longword to return message size
: 440 0580 1 (in bytes)
: 441 0581 1
: 442 0582 1 OUTPUTS:
: 443 0583 1
: 444 0584 1 Message info copied to buffers.
: 445 0585 1 --
: 446 0586 2 BEGIN
: 447 0587 2 BUILTIN
: 448 0588 2 NULLPARAMETER,
: 449 0589 2 ACTUALCOUNT: ! Argument count
: 450 0590 2 LOCAL
: 451 0591 2 MSGADDR,
: 452 0592 2 POINTER_1, ! String pointer
: 453 0593 2 POINTER_2,
: 454 0594 2 POINTER_3;
: 455 0595 2
: 456 P 0596 2 IF_ERR( MAIL$REWIND MSG();, ! Rewind the message
: 457 0597 2 RETURN .STATUS);
: 458 0598 2
: 459 0599 2 Get the sender of the current message
: 460 0600 2
: 461 0601 2 POINTER_1 = 0;
: 462 0602 2 IF .MAIL$W_MSG_RSIZ GTRU FROM_SIZ
: 463 0603 2 THEN
: 464 0604 2 POINTER_1 = CH$FIND_CH(.MAIL$W_MSG_RSIZ - FROM_SIZ, ! Find end of name
: 465 0605 2 ;MAIL$L_MSG_RADR + FROM_SIZ,
: 466 0606 2 ');
: 467 0607 2 IF NOT CH$FAIL(.POINTER_1)
: 468 0608 2 THEN
: 469 0609 2 OTSS$COPY_R_DX( .POINTER_1 - (.MAIL$L_MSG_RADR + FROM_SIZ),
: 470 0610 2 .MAIL$L_MSG_RADR + FROM_SIZ,
: 471 0611 2 .SENDER)
: 472 0612 2 ELSE
: 473 0613 2 OTSS$COPY_R_DX( 0,0,.SENDER);
: 474 0614 2
: 475 0615 2 Get date of current message
: 476 0616 2
: 477 0617 2 IF NOT NULLPARAMETER(3)
: 478 0618 2 THEN BEGIN
: 479 0619 2 IF .POINTER_1 EQL 0
: 480 0620 2 THEN POINTER_1 = .MAIL$L_MSG_RADR + .MAIL$W_MSG_RSIZ;
: 481 0621 2 POINTER_2 = CH$FIND_CH( !Look for double quotes
```



```
482 0622 3 .MAIL$W MSG_RSIZ-(.POINTER_1-.MAIL$M_MSG_RADR),
483 0623 3 .POINTER_1+T, ''););
484 0624 3 IF NOT CH$FAIL(.POINTER_2)
485 0625 4 THEN BEGIN
486 0626 4     POINTER_3 = CH$FIND_CH(
487 0627 4         .MAIL$W MSG_RSIZ-(.POINTER_2-.MAIL$M_MSG_RADR),
488 0628 4         .POINTER_2+T, ''););
489 0629 4     IF NOT CH$FAIL(.POINTER_3)
490 0630 4         THEN POINTER_1 = .POINTER_3;
491 0631 3     END;
492 0632 3     POINTER_2 = CH$FIND_CH(
493 0633 3         .MAIL$W MSG_RSIZ-(.POINTER_1-.MAIL$M_MSG_RADR), ! Find date
494 0634 3         .POINTER_1 + 1,
495 0635 3         '-'););
496 0636 3     IF NOT CH$FAIL(.POINTER_2)
497 0637 3     THEN
498 0638 3         OTSS$COPY_R_DX( .MAIL$W MSG_RSIZ-
499 0639 3             (.POINTER_2 - .MAIL$M_MSG_RADR - 2),
500 0640 3             .POINTER_2 - 2,
501 0641 3             .DATE)
502 0642 3     ELSE
503 0643 3         OTSS$COPY_R_DX( 0,0,.DATE);
504 0644 2     END;
505 0645 2
506 0646 2 : Get the 'to' list
507 0647 2
508 0648 2 IF MAIL$GET_MSG()
509 0649 2     THEN IF NOT NULLPARAMETER(4)
510 0650 2         THEN OTSS$COPY_R_DX(.MAIL$W MSG_RSIZ - TO_SIZ,
511 0651 2             .MAIL$M_MSG_RADR + TO_SIZ,
512 0652 2             .TO_DEST);
513 0653 2
514 0654 2 : Get subject, if there is one, of the current message
515 0655 2
516 0656 2 IF MAIL$GET_MSG()
517 0657 2     THEN
518 0658 2         IF .MAIL$W MSG_RSIZ GTRU SUBJ_SIZ
519 0659 2         AND (.MAIL$M_MSG_RADR) EQL 'Subj'
520 0660 2         THEN
521 0661 2             OTSS$COPY_R_DX( .MAIL$W MSG_RSIZ - FROM_SIZ,
522 0662 2                 .MAIL$M_MSG_RADR + FROM_SIZ,
523 0663 2                 .SUBJ)
524 0664 2         ELSE
525 0665 2             OTSS$COPY_R_DX( 0,0,.SUBJ);
526 0666 2
527 0667 2 : If requested, count bytes in message
528 0668 2
529 0669 2 IF NOT NULLPARAMETER(5)
530 0670 3     THEN BEGIN
531 0671 3         MSGADDR = .MSG_ADR;
532 0672 3         MSGADDR = .MSGADDR + .RECSIZ(.MSGADDR) + RECSIZ_SIZ; !Skip From line
533 0673 4         WHILE (.MSGADDR LEQU (.MSG_BUF_ADR + .MSG_BUF_SIZ))
534 0674 4             AND NOT MSG$DELIM(.MSGADDR)
535 0675 3             DO MSGADDR = .MSGADDR + .RECSIZ(.MSGADDR) + RECSIZ_SIZ;
536 0676 3         .MESSAGE_SIZE = .MSGADDR - .MSG_ADR;
537 0677 2     END;
538 0678 2 RETURN TRUE
```

: 539

0679 1 END;

			03FC 00000	.ENTRY	MAIL\$INFO MSG, Save R2,R3,R4,R5,R6,R7,R8,R9	: 0565
		59	FF60 CF 9E 00002	MOVAB	MAIL\$GET_MSG, R9	
		58	00000000' 00 9E 00007	MOVAB	MSG_ADR, R8	
		57	000000000G 00 9E 0000E	MOVAB	OTSS\$COPY_R_DX, R7	
		56	00000000' 00 9E 00015	MOVAB	MAIL\$ MSG_RADR, R6	
5A		A9	00 FB 0001C	CALLS	#0, MAIL\$REWIND_MSG	: 0597
		01	50 EB 00020	BLBS	STATUS, 1\$	
			04 00023	RET		
			53 D4 00024 1\$:	CLRL	POINTER_1	: 0601
		50	FC A6 3C 00026	MOVZWL	MAIL\$W_MSG_RSIZ, R0	: 0602
		06	50 B1 0002A	CMPW	R0, #6	
			12 1B 0002D	BLEQU	3\$	
		50	06 C2 0002F	SUBL2	#6, R0	: 0604
06	A1	51	66 D0 00032	MOVL	MAIL\$ MSG_RADR, R1	: 0605
		50	20 3A 00035	LOCC	#32, R0, 6(R1)	: 0604
			02 12 0003A	BNEQ	2\$	
			51 D4 0003C	CLRL	R1	
		53	51 D0 0003E 2\$:	MOVL	R1, POINTER_1	
			52 D4 00041 3\$:	CLRL	R2	: 0607
			53 D5 00043	TSTL	POINTER_1	
			04 12 00045	BNEQ	4\$	
			52 D6 00047	INCL	R2	
			0F 11 00049	BRB	5\$	
			04 AC DD 0004B 4\$:	PUSHL	SENDER	: 0611
	50	66	06 C1 0004E	ADDL3	#6, MAIL\$ MSG_RADR, R0	: 0610
			50 DD 00052	PUSHL	R0	
	7E	53	50 C3 00054	SUBL3	R0, POINTER_1, -(SP)	: 0609
			05 11 00058	BRB	6\$	
			04 AC DD 0005A 5\$:	PUSHL	SENDER	: 0613
			7E 7C 0005D	CLRQ	-(SP)	
		67	03 FB 0005F 6\$:	CALLS	#3, OTSS\$COPY_R_DX	
		03	6C 91 00062	CMPB	(AP), #3	: 0617
			72 1F 00065	BLSSU	14\$	
			0C AC D5 00067	TSTL	12(AP)	
			6D 13 0006A	BEQL	14\$	
		07	52 E9 0006C	BLBC	R2, 7\$: 0619
		53	FC A6 3C 0006F	MOVZWL	MAIL\$W_MSG_RSIZ, POINTER_1	: 0620
		53	66 C0 00073	ADDL2	MAIL\$ MSG_RADR, POINTER_1	
		55	FC A6 3C 00076 7\$:	MOVZWL	MAIL\$W_MSG_RSIZ, R5	: 0622
		54	66 D0 0007A	MOVL	MAIL\$ MSG_RADR, R4	
	50	54	53 C3 0007D	SUBL3	POINTER_1, R4, R0	
		50	55 C0 00081	ADDL2	R5, R0	
01	A3	50	22 3A 00084	LOCC	#34, R0, 1(POINTER_1)	: 0621
			02 12 00089	BNEQ	8\$	
			51 D4 0008B	CLRL	R1	
		52	51 D0 0008D 8\$:	MOVL	R1, POINTER_2	
			17 13 00090	BEQL	10\$: 0624
	50	54	52 C3 00092	SUBL3	POINTER_2, R4, R0	: 0627
		50	55 C0 00096	ADDL2	R5, R0	
01	A2	50	22 3A 00099	LOCC	#34, R0, 1(POINTER_2)	: 0626
			02 12 0009E	BNEQ	9\$	

			51	D4	000A0		CLRL	R1		
			51	D5	000A2	9\$:	TSTL	POINTER_3		0629
			03	13	000A4		BEQL	10\$		
		53	51	D0	000A6		MOVL	POINTER_3, POINTER_1		0630
	50	54	53	C3	000A9	10\$:	SUBL3	POINTER_1, R4, R0		0633
		50	55	C0	000AD		ADDL2	R5, R0		
01	A3	50	2D	3A	000B0		LOCC	#4\$, R0, 1(POINTER_1)		0632
			02	12	000B5		BNEQ	11\$		
			51	D4	000B7		CLRL	R1		
		52	51	D0	000B9	11\$:	MOVL	R1, POINTER_2		
			13	13	000BC		BEQL	12\$		0636
			OC	AC	DD	000BE	PUSHL	DATE		0641
			FE	A2	9F	000C1	PUSHAB	-2(POINTER_2)		0640
	51	52	54	C3	000C4		SUBL3	R4, POINTER_2, R1		0639
	50	55	51	C3	000C8		SUBL3	R1, R5, R0		
			02	A0	9F	000CC	PUSHAB	2(R0)		0638
			05	11	000CF		BRB	13\$		
			OC	AC	DD	000D1	12\$:	PUSHL	DATE	0643
			7E	7C	000D4		CLRQ	-(SP)		
		67	03	FB	000D6	13\$:	CALLS	#3, OTSS\$COPY_R_DX		
		69	00	FB	000D9	14\$:	CALLS	#0, MAIL\$GET_MSG		0648
		1B	50	E9	000DC		BLBC	R0, 15\$		
		04	6C	91	000DF		CMPB	(AP), #4		0649
			16	1F	000E2		BLSSU	15\$		
			10	AC	D5	000E4	TSTL	16(AP)		
			11	13	000E7		BEQL	15\$		
			10	AC	DD	000E9	PUSHL	TO_DESC		0652
	7E	66	04	C1	000EC		ADDL3	#4, MAIL\$L_MSG_RADR, -(SP)		0651
		7E	FC	A6	3C	000F0	MOVZWL	MAIL\$W_MSG_RSIZ, -(SP)		0650
		6E	04	C2	000F4		SUBL2	#4, (SP)		
		67	03	FB	000F7		CALLS	#3, OTSS\$COPY_R_DX		
		69	00	FB	000FA	15\$:	CALLS	#0, MAIL\$GET_MSG		0656
		29	50	E9	000FD		BLBC	R0, 18\$		
		50	FC	A6	3C	00100	MOVZWL	MAIL\$W_MSG_RSIZ, R0		0658
		06	50	C1	00104		CMPW	R0, #6		
			18	1B	00107		BLEQU	16\$		
		51	66	D0	00109		MOVL	MAIL\$L_MSG_RADR, R1		0659
	6A627553	8F	61	D1	0010C		CMPB	(R1), #1784837459		
			0C	12	00113		BNEQ	16\$		
			08	AC	DD	00115	PUSHL	SUBJ		0663
			06	C1	00118		ADDL3	#6, MAIL\$L_MSG_RADR, -(SP)		0662
	7E	66	FA	A0	9F	0011C	PUSHAB	-6(R0)		0661
			05	11	0011F		BRB	17\$		
			08	AC	DD	00121	16\$:	PUSHL	SUBJ	0665
			7E	7C	00124		CLRQ	-(SP)		
		67	03	FB	00126	17\$:	CALLS	#3, OTSS\$COPY_R_DX		
		05	6C	91	00129	18\$:	CMPB	(AP), #5		0669
			48	1F	0012C		BLSSU	22\$		
			14	AC	D5	0012E	TSTL	20(AP)		
			43	13	00131		BEQL	22\$		
		53	68	D0	00133		MOVL	MSG_ADR, R3		0671
		50	53	D0	00136		MOVL	R3, MSGADDR		
		51	60	9A	00139		MOVZBL	(MSGADDR), R1		0672
		50	01	A140	9E	0013C	MOVAB	1(R1)[MSGADDR], MSGADDR		
	52	F8	FC	A8	C1	00141	ADDL3	MSG_BUF_SIZ, MSG_BUF_ADR, R2		0673
			50	D1	00147	19\$:	CMPB	MSGADDR, R2		
			25	1A	0014A		BGTRU	21\$		

MAIL\$MSGSUBS
V04-000

MAIL MESSAGE SUBROUTINES

E 9
16-Sep-1984 01:08:07
14-Sep-1984 12:42:28

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

Page 18
(6)

01		60	91	0014C	CMPB	(MSGADDR), #1	:	0674
		16	12	0014F	BNEQ	20\$:	
0C	01	A0	91	00151	CMPB	1(MSGADDR), #12	:	
		10	12	00155	BNEQ	20\$:	
04	02	A0	91	00157	CMPB	2(MSGADDR), #4	:	
		0A	1B	0015B	BLEQU	20\$:	
6D6F7246	8F	03	A0	D1 0015D	CMPL	3(MSGADDR), #1836020294	:	
		0A	13	00165	BEQL	21\$:	
51		60	9A	00167	MOVZBL	(MSGADDR), R1	:	0675
50	01	A140	9E	0016A	MOVAB	1(R1)[MSGADDR], MSGADDR	:	
		D6	11	0016F	BRB	19\$:	
14 BC		50	53	C3 00171	SUBL3	R3, MSGADDR, @MESSAGE_SIZE	:	0676
		50	01	D0 00176	MOVL	#1, R0	:	0678
			04	00179	RET		:	0679

; Routine Size: 378 bytes, Routine Base: \$CODE\$ + 01B1

; 540 0680 1

MAI
V04

:

```

542 0681 1 GLOBAL ROUTINE MAIL$TEST_SUBJ =
543 0682 1  +-
544 0683 1  FUNCTIONAL DESCRIPTION:
545 0684 1
546 0685 1      This routine tests if the SUBJ: line of the message is present
547 0686 1
548 0687 1  ROUTINE VALUE:
549 0688 1
550 0689 1      True if SUBJ: line present, false if not.
551 0690 1
552 0691 1  --
553 0692 2 BEGIN
554 0693 2
555 0694 2 LOCAL
556 0695 2      STATUS,
557 0696 2      DESC_1 : $BBLOCK[8],
558 0697 2      DESC_2 : $BBLOCK[8];
559 0698 2
560 0699 2 INIT_DDESC(DESC_1);
561 0700 2 INIT_DDESC(DESC_2);
562 0701 2
563 0702 2 MAIL$INFO_MSG(DESC_1,DESC_2);
564 0703 2
565 0704 2 STATUS = .DESC_2[DESC$W_LENGTH] NEQ 0;
566 0705 2 STR$FREE1_DX(DESC_1);
567 0706 2 STR$FREE1_DX(DESC_2);
568 0707 2
569 0708 2 MAIL$REWIND_MSG();
570 0709 2
571 0710 2 RETURN .STATUS
572 0711 1 END;

```

```

                                000C 00000      .ENTRY MAIL$TEST_SUBJ, Save R2,R3      : 0681
                                00 9E 00002      MOVAB STR$FREE1_DX, R3      :
                                0C C2 00009      SUBL2 #12, SP      :
                                04 AE 020E000 8F D0 0000C      MOVL #34471936, DESC_1      : 0699
                                08 AE D4 00014      CLRL DESC_1+4      :
                                020E0000 8F DD 00017      PUSHL #34471936      : 0700
                                04 AE D4 0001D      CLRL DESC_2+4      :
                                SE DD 0C020      PUSHL SP      : 0702
                                0C AE 9F 00022      PUSHAB DESC_1      :
                                FESC CF 02 FB 00025      CALLS #2, MAIL$INFO_MSG      :
                                50 D4 0002A      CLRL R0      : 0704
                                6E B5 0002C      TSTW DESC_2
                                02 13 0002E      BEQL 1$
                                50 D6 00030      INCL R0
                                52 50 D0 00032 1$:      MOVL R0, STATUS      :
                                08 AE 9F 00035      PUSHAB DESC_1      : 0705
                                63 01 FB 00038      CALLS #1, STR$FREE1_DX      :
                                SE DD 0003B      PUSHL SP      : 0706
                                FE01 63 01 FB 0003D      CALLS #1, STR$FREE1_DX      :
                                CF 00 FB 00040      CALLS #0, MAIL$REWIND_MSG      : 0708
                                50 52 D0 00045      MOVL STATUS, R0      : 0710

```

MAIL\$MSGSUBS MAIL MESSAGE SUBROUTINES
V04-000

G 9
16-Sep-1984 01:08:07
14-Sep-1984 12:42:28

VAX-11 Bliss-32 V4.0-742 Page 20
DISK\$VM\$MASTER:[MAIL.SRC]MSGSUBS.B32;1 (7)

04 00048 RET

; 0711

; Routine Size: 73 bytes, Routine Base: \$CODE\$ + 032B

MAI
V04

: R

```

: 574 0712 1 GLOBAL ROUTINE MAIL$TEST_SEQ_NEW =
: 575 0713 1 |++
: 576 0714 1 | FUNCTIONAL DESCRIPTION:
: 577 0715 1 |
: 578 0716 1 |         Test if the current sequential message is a new message
: 579 0717 1 | --
: 580 0718 2 BEGIN
: 581 0719 2
: 582 0720 2 IF CH$RCHAR(NEWMSGFLAG(.MSG_ADR)) EQL '+'
: 583 0721 2     THEN RETURN TRUE
: 584 0722 2     ELSE RETURN FALSE
: 585 0723 1 END;

```

```

                                0000 00000
50 00000000' 00 D0 00002      .ENTRY MAIL$TEST_SEQ_NEW, Save nothing
51           60 9A 00009      MOVL MSG_ADR, R0
50           01 A140 9E 0000C  MOVZBL (R0), R1
51           60 9A 00011      MOVAB 1(R1)[R0], PTR
2A           6140 91 00014      MOVZBL (PTR), R1
                                04 12 00018      CMPB (R1)[PTR], #42
50           01 D0 0001A      BNEQ 1$
                                04 0001D      MOVL #1, R0
                                50 D4 0001E 1$: RET
                                04 00020      CLRL R0
                                RET
: 0712
: 0720
: 0722
: 0723

```

: Routine Size: 33 bytes, Routine Base: \$CODE\$ + 0374

```
587 0724 1 GLOBAL ROUTINE MAIL$NEW_MSG =
588 0725 1 ++
589 0726 1 FUNCTIONAL DESCRIPTION:
590 0727 1
591 0728 1 This routine is called to test if the current message is a 'new'
592 0729 1 (previously unread) message. If it is, the new flag in the message
593 0730 1 is cleared.
594 0731 1
595 0732 1 INPUTS:
596 0733 1
597 0734 1 MSG_ADR = Address of first record in current message.
598 0735 1
599 0736 1 OUTPUTS:
600 0737 1
601 0738 1 STS$K_SUCCESS if message is new.
602 0739 1 STS$K_ERROR if message is not new.
603 0740 1 --
604 0741 2 BEGIN
605 0742 2 LOCAL
606 0743 2 RECBUFFER : $BBLOCK[$BYTEOFFSET(MFR_T_DATA)+8],
607 0744 2 RECDISC : VECTOR[2, LONG],
608 0745 2 PTR; ! Buffer pointer
609 0746 2
610 0747 2 IF .MAIL$GL_FLAGS[MAIF_V_ISAM]
611 0748 3 THEN BEGIN
612 0749 3 IF .MAIL$CURMSG[MMB_V_NEWSMSG]
613 0750 4 THEN BEGIN
614 0751 4
615 0752 4 ! Update record in file to show it's been read
616 0753 4
617 0754 4 BIND
618 0755 4 RECDATIM = RECBUFFER[MFR_Q_DATIM] : VECTOR[2, LONG],
619 0756 4 MFR = MAIL$CURMSG[MMB_T_ADREC] : $BBLOCK,
620 0757 4 MMH = MFR + $BYTEOFFSET(MFR_T_DATA) : $BBLOCK;
621 0758 4
622 0759 4 MMH[MMH_V_NEWSMSG] = FALSE;
623 0760 4 MAIL$CURMSG[MMB_V_NEWSMSG] = FALSE;
624 0761 4
625 0762 4 ! Only set key to MAIL if it was NEWSMAIL
626 0763 4
627 0764 4 IF CH$EQL(.MFR[MFR_B_FILEKEYLEN], MFR[MFR_T_FILEKEY],
628 0765 4 .MAIL$SD_NEWSMAIL[DSC$W_LENGTH],
629 0766 4 .MAIL$SD_NEWSMAIL[DSC$A_POINTER])
630 0767 5 THEN BEGIN
631 0768 5 MFR[MFR_B_FILEKEYLEN] = %CHARCOUNT('MAIL');
632 0769 5 CH$COPY(%CHARCOUNT('MAIL'), UPLIT(%ASCII 'MAIL'), 0,
633 0770 5 MFR_S_FILEKEY, MFR[MFR_T_FILEKEY]);
634 0771 4 END;
635 0772 4 MAIL$UPDATE_I_MSG();
636 0773 4 RETURN TRUE
637 0774 4 END
638 0775 3 ELSE RETURN FALSE
639 0776 3 END
640 0777 3 ELSE BEGIN
641 0778 3 IF CH$RCHAR(PTR = NEWSMSGFLAG(.MSG_ADR)) EQL '*' ! If 'new' flag
642 0779 4 THEN BEGIN
643 0780 4 CH$WCHAR(' ', PTR); ! Clear text flag
```



```

: 644      0781 4      MODIFIED = 1;      ! Set modified flag
: 645      0782 4      RETURN TRUE      ! Return record is new
: 646      0783 4      END
: 647      0784 3      ELSE
: 648      0785 3      RETURN FALSE      ! Return not 'new'
: 649      0786 2      END:
: 650      0787 1      END:

```

```

4C 49 41 4D 00395 .BLKB 3
00398 P.AAE: .ASCII \MAIL\

```

```

00000000G 00      00      09      A4      51 2D 0002F      .ENTRY MAIL$NEW_MSG, Save R2,R3,R4,R5      : 0724
: 00000000G 00      00      09      A4      51 2D 0002F      MOVAB -64(SP), SP      : 0747
: 00000000G 00      00      09      A4      51 2D 0002F      BLBC MAIL$GL_FLAGS, 2$      : 0749
: 00000000G 00      00      09      A4      51 2D 0002F      MOVL MAIL$_CURMSG, R0      : 0756
: 00000000G 00      00      09      A4      51 2D 0002F      BLBC 10(R0), 4$      : 0759
: 00000000G 00      00      09      A4      51 2D 0002F      MOVAB 71(R0), R4      : 0760
: 00000000G 00      00      09      A4      51 2D 0002F      BICB2 #1, 48(R4)      : 0764
: 00000000G 00      00      09      A4      51 2D 0002F      BICB2 #1, 10(R0)      : 0766
: 00000000G 00      00      09      A4      51 2D 0002F      MOVZBL 8(R4), R1      : 0764
: 00000000G 00      00      09      A4      51 2D 0002F      MOVL MAIL$SD_NEWMAIL+4, R0      : 0766
: 00000000G 00      00      09      A4      51 2D 0002F      CMPC5 R1, 9(R4), #0, MAIL$SD_NEWMAIL, (R0)      : 0764
: 00000000G 00      00      09      A4      51 2D 0002F      BNEQ 1$      : 0768
: 00000000G 00      00      09      A4      51 2D 0002F      MOVB #4, 8(R4)      : 0770
: 00000000G 00      00      09      A4      51 2D 0002F      MOVCS #4, P.AAE, #0, #39, 9(R4)      : 0772
: 00000000G 00      00      09      A4      51 2D 0002F      CALLS #0, MAIL$UPDATE_I_MSG      : 0773
: 00000000G 00      00      09      A4      51 2D 0002F      BRB 3$      : 0778
: 00000000G 00      00      09      A4      51 2D 0002F      MOVL MSG_ADR, R0      : 0780
: 00000000G 00      00      09      A4      51 2D 0002F      MOVZBL (R0), R1      : 0781
: 00000000G 00      00      09      A4      51 2D 0002F      MOVAB 1(R1)[R0], PTR      : 0782
: 00000000G 00      00      09      A4      51 2D 0002F      MOVZBL (PTR), R1      : 0787
: 00000000G 00      00      09      A4      51 2D 0002F      ADDL2 R1, PTR      :
: 00000000G 00      00      09      A4      51 2D 0002F      CMPB (PTR), #42      :
: 00000000G 00      00      09      A4      51 2D 0002F      BNEQ 4$      :
: 00000000G 00      00      09      A4      51 2D 0002F      MOVB #32, (PTR)      :
: 00000000G 00      00      09      A4      51 2D 0002F      MOVL #1, MODIFIED      :
: 00000000G 00      00      09      A4      51 2D 0002F      MOVL #1, R0      :
: 00000000G 00      00      09      A4      51 2D 0002F      RET      :
: 00000000G 00      00      09      A4      51 2D 0002F      CLRL R0      :
: 00000000G 00      00      09      A4      51 2D 0002F      RET      :

```

: Routine Size: 124 bytes, Routine Base: \$CODE\$ + 039C

: 651 0788 1

```
653 0789 1 GLOBAL ROUTINE MAIL$DELETE_MSG =
654 0790 1 |++
655 0791 1 | FUNCTIONAL DESCRIPTION:
656 0792 1 |
657 0793 1 |     This routine is called to delete the current message from the message
658 0794 1 |     buffer and, ultimately, from the message file.
659 0795 1 |
660 0796 1 | INPUTS:
661 0797 1 |
662 0798 1 |     MSG_ADR = Address of first record in current message.
663 0799 1 |
664 0800 1 | OUTPUTS:
665 0801 1 |
666 0802 1 |     Current message removed from the message buffer.
667 0803 1 | --
668 0804 2 BEGIN
669 0805 2 LOCAL
670 0806 2     TEXT_LEFT,           ! Size of text left
671 0807 2     FROM_PTR,           ! Move from text pointer
672 0808 2     TO_PTR,             ! Move to text pointer
673 0809 2     MOVE_SIZ;          ! Size of text moving
674 0810 2
675 0811 2 IF_ERR(MAIL$REWIND MSG()); RETURN .STATUS); ! Rewind the message
676 0812 2 IF NOT .MAIL$GL_FLAGS[MAIF_V_READ_FL]
677 0813 2 THEN RETURN MAIL$NOTREADIN;
678 0814 2 MODIFIED = 1;          ! Set modified flag
679 0815 2 WHILE MAIL$GET_MSG() DO 1; ! Find end of message
680 0816 2
681 0817 2 TEXT_LEFT = (.MSG_BUF_ADR + .MSG_BUF_SIZ) - ! Compute text left after msg
682 0818 2     (.MAIL$MSG_RADR - RECSIZ_SIZ);
683 0819 2 FROM_PTR = .MAIL$MSG_RADR - RECSIZ_SIZ; ! Get address to move from
684 0820 2 TO_PTR = .MSG_ADR; ! C * address to move to
685 0821 2 DO
686 0822 2     BEGIN
687 0823 2     MOVE_SIZ = MIN(.TEXT_LEFT, 65535); ! Compute max that can be moved
688 0824 2     CH$MOVE(.MOVE_SIZ, .FROM_PTR, .TO_PTR); ! Move text
689 0825 2     TO_PTR = .TO_PTR + .MOVE_SIZ; ! Compute new to pointer
690 0826 2     FROM_PTR = .FROM_PTR + .MOVE_SIZ; ! Compute new from pointer
691 0827 2     TEXT_LEFT = .TEXT_LEFT - .MOVE_SIZ; ! Compute amount left now
692 0828 2     END
693 0829 2 UNTIL .TEXT_LEFT EQL 0; ! Continue until all moved
694 0830 2 MSG_BUF_SIZ = .MSG_BUF_SIZ - ! Compute new size of text
695 0831 2     ((.MAIL$MSG_RADR - RECSIZ_SIZ) - .MSG_ADR);
696 0832 2
697 0833 2 MAIL$GL_FLAGS[MAIF_V_READ_FL] = 0;
698 0834 2 IF .MSG_ADR LSSU (.MSG_BUF_ADR + .MSG_BUF_SIZ) ! If not past end of buffer
699 0835 2 THEN
700 0836 2     MAIL$REWIND_MSG() ! Rewind new current message
701 0837 2 ELSE
702 0838 2     MSG_ADR = MAIL$MSG_STS = 0; ! Else, end of file
703 0839 2 RETURN ST$R_SUCCESS
704 0840 1 END;
```



```

: 706 0841 1 ROUTINE TELL_CVTFILE (DESC) =
: 707 0842 1 :++
: 708 0843 1 : FUNCTIONAL DESCRIPTION:
: 709 0844 1 :
: 710 0845 1 : Signal that a message file is being converted to an ISAM file
: 711 0846 1 :
: 712 0847 1 : --
: 713 0848 1 SIGNAL(MAIL$_CVTFILE,2,MAIL$_RD_DESC,.DESC);

```

```

0000 0000 TELL_CVTFILE:
          04 AC DD 00002 .WORD Save nothing
00000000' 00 9F 00005 PUSHL DESC
          02 DD 0000B PUSHAB MAIL$_RD_DESC
          8F DD 0000D PUSHL #2
00000000G 00 04 FB 00013 CALLS #8306323
          04 0001A RET #4, LIB$SIGNAL

```

```

: 0841
: 0848
:
:
:

```

; Routine Size: 27 bytes, Routine Base: \$CODE\$ + 04D4

```
0849 1 GLOBAL ROUTINE MAIL$CLOSE_FILE : NOVALUE =
0850 1 **
0851 1 FUNCTIONAL DESCRIPTION:
0852 1
0853 1 This routine closes the current message file. If any messages were
0854 1 deleted, all the un-deleted messages are first copied, contiguously,
0855 1 to the message file, thus truncating the file. The message buffer
0856 1 is then deallocated.
0857 1
0858 1 INPUTS:
0859 1
0860 1 MSG_BUF_ASIZ = Size of allocated message buffer.
0861 1 MSG_BUF_ADR = Starting address of message buffer.
0862 1 MSG_BUF_SIZ = Size of messages in message buffer.
0863 1
0864 1 OUTPUTS:
0865 1
0866 1 Messages copied to message file, message file truncated and closed,
0867 1 message buffer deallocated, message file context initialized.
0868 1
0869 1 All failures are signalled.
0870 1 --
0871 2 BEGIN
0872 2 LOCAL
0873 2 PTR, ! Message buffer pointer
0874 2 SIZ, ! Message buffer size
0875 2 STATUS; ! Routine status
0876 2
0877 2 ! If no file open, just return
0878 2
0879 2 IF .RD_FAB[FAB$W_IFI] EQL 0 THEN RETURN;
0880 2
0881 2 IF .MAIL$GL_FLAGS[MAIF_V_ISAM]
0882 3 THEN BEGIN
0883 3 IF NOT .MAIL$L_USRSYSBUF[VMD V_NOAUTOPURGE]
0884 3 THEN MAIL$EMPTY_WASTEBASKET();
0885 3 MAIL$DESELECT_MSGS();
0886 3 END
0887 3 ELSE BEGIN
0888 3
0889 3 ! If there are messages left, rewind the message file and copy the
0890 3 ! messages to it. If none left, mark the file for deletion.
0891 3
0892 3 IF .MODIFIED THEN
0893 3 IF .MSG_BUF_SIZ NEQ 0 ! If text left
0894 4 THEN BEGIN
0895 4 $DCLEXH(DESBLK = EXITBLOCK); ! Re-declare if interrupted
0896 4 $WAIT(RAB = RD_RAB); ! Wait for all quiet
0897 4 STATUS = $REWIND(RAB = RD_RAB);
0898 4 PTR = .MSG_BUF_ADR;
0899 4 SIZ = .MSG_BUF_SIZ;
0900 4 WHILE .SIZ > 0 AND .STATUS
0901 5 DO BEGIN
0902 5 RD_RAB[RAB$L_RBF] = .PTR + RECSIZ_SIZ;
0903 5 RD_RAB[RAB$W_RSZ] = .RECSIZ(.PTR);
0904 5 SIZ = .SIZ - .RECSIZ(.PTR) - RECSIZ_SIZ;
0905 5 PTR = .PTR + .RECSIZ(.PTR) + RECSIZ_SIZ;
```

```

772 0906 5 STATUS = $PUT(RAB = RD_RAB,ERR=UTIL$REPORT_IO_ERROR);
773 0907 4 END;
774 0908 4 END
775 0909 3 ELSE
776 0910 3 RD FAB[FAB$V_DLT] = 1; ! Set to delete
777 0911 3 LIB$FREE_VM(MSG_BUF_A$IZ,MSG_BUF_ADR); ! Deallocate buffer
778 0912 2 END;
779 0913 2
780 0914 2 ! Close the message file
781 0915 2
782 0916 2 MAIL$UP_PRIV(1*$BITPOSITION(PRV$V_SYSPRV),0); ! Enable system priv
783 P 0917 2 IF_ERR($CLOSE(FAB = RD_FAB);, ! Close the file
784 0918 2 SIGNAL (.STATUS)); ! possibly deleting it
785 0919 2 MAIL$DOWN_PRIV();
786 0920 2 $SCANEXH(DESBLK = EXITBLOCK); ! Cancel exit handler
787 0921 2
788 0922 2 ! Re-initialize all the message file context
789 0923 2
790 0924 2 MAIL$CLR_DIR_FLAG(); !Clear directory display flags
791 0925 2 MODIFIED = 0;
792 0926 2 MSG_BUF_A$IZ = 0;
793 0927 2 MSG_BUF_SIZ = 0;
794 0928 2 MSG_BUF_ADR = 0;
795 0929 2 M_ADR = 0;
796 0930 2 M/ [$L_MSG_NUM = 0;
797 0931 2 MAIL$W_MSG_RSIZ = 0;
798 0932 2 MAIL$L_MSG_RADR = 0;
799 0933 2 MAIL$L_MSG_STS = MAIL$ NOTREADIN;
800 0934 2 MAIL$G_FLAGS[MAIF_V_ISAM] = 0;
801 0935 2 MAIL$G_FLAGS[MAIF_V_FILOPN] = 0;
802 0936 2 MAIL$G_FLAGS[MAIF_V_DFMSG] = 0;
803 0937 1 END;

```

```

.EXTRN SYSSDCLEXH, SYSSWAIT
.EXTRN SYSSREWIND, SYSSPUT
.EXTRN SYSSCLOSE, SYSSCANEXH

```

			00FC 0000	.ENTRY MAIL\$CLOSE FILE, Save R2,R3,R4,R5,R6,R7	: 0849
	57	00000000G	00 9E 00002	MOVAB MAIL\$GL_FLAGS, R7	
	56	00000000'	00 9E 00009	MOVAB MAIL\$L_MSG_NUM, R6	
	55	00000000'	00 9E 00010	MOVAB MSG_BUF_SIZ, R5	
		FF66	C5 B5 00017	TSTW RD_FAB+2	: 0879
			01 12 0001B	BNEG 1\$	
			04 0001D	RET	
	1C		67 E9 0001E 1\$:	BLBC MAIL\$GL_FLAGS, 3\$: 0881
	50	00000000G	00 D0 00021	MOVL MAIL\$L_USRSYSBUF, R0	: 0883
07	1F	A0	02 E0 00028	BBS #2, 31(TRO), 2\$	
		00000000G	00 00 FB 0002D	CALLS #0, MAIL\$EMPTY_WASTEBASKET	: 0884
		00000000G	00 00 FB 00034 2\$:	CALLS #0, MAIL\$DESELECT_MSGS	: 0885
			74 11 0003B	BRB 7\$: 0881
	62	FEEB	C5 E9 0003D 3\$:	BLBC MODIFIED, 6\$: 0892
			65 D5 00042	TSTL MSG_BUF_SIZ	: 0893
			58 13 00044	BEQL 5\$	
		FEDB	C5 9F 00046	PUSHAB EXITBLOCK	: 0895
		00000000G	00 01 FB 0004A	CALLS #1, SYSSDCLEXH	

			B4	A5	9F	00051	PUSHAB	RD_RAB	: 0896	
	00000000G	00		01	FB	00054	CALLS	#1, SYSS\$WAIT		
			B4	A5	9F	0005B	FUSHAB	RD_RAB	: 0897	
	00000000G	00		01	FB	0005E	CALLS	#1, SYSS\$REWIND		
		53	FC	A5	D0	00065	MOVL	MSG_BUF_ADR, PTR	: 0898	
		52		65	D0	00069	MOVL	MSG_BUF_SIZ, SIZ	: 0899	
				52	D5	0006C	TSTL	SIZ	: 0900	
				34	15	0006E	BLEQ	6\$		
		31		50	E9	00070	BLBC	STATUS, 6\$		
	DC	A5	01	A3	9E	00073	MOVAB	1(R3), RD_RAB+40	: 0902	
		51		63	9A	00078	MOVZBL	(PTR), R1	: 0903	
54	D6	A5		51	B0	0007B	MOVW	R1, RD_RAB+34		
		52		51	C3	0007F	SUBL3	R1, SIZ, R4	: 0904	
		52	FF	A4	9E	00083	MOVAB	-1(R4), SIZ		
		53	01	A1	43	9E	00087	MOVAB	1(R1)[PTR], PTR	: 0905
			00000000G	00	9F	0008C	PUSHAB	UTIL\$REPORT_IO_ERROR	: 0906	
			B4	A5	9F	00092	PUSHAB	RD_RAB		
	00000000G	00		02	FB	00095	CALLS	#2, SYSS\$PUT		
				CE	11	0009C	BRB	4\$: 0900	
	FF69	C5	80	8F	88	0009E	BISB2	#128, RD_FAB+5	: 0910	
			FC	A5	9F	000A4	PUSHAB	MSG_BUF_ADR	: 0911	
			F8	A5	9F	000A7	PUSHAB	MSG_BUF_ASIZ		
	00000000G	00		02	FB	000AA	CALLS	#2, LIB\$FREE_VM		
				7E	D4	000B1	CLRL	-(SP)	: 0916	
			10000000	8F	DD	000B3	PUSHL	#268435456		
	00000000G	00		02	FB	000B9	CALLS	#2, MAIL\$UP_PRIV		
			FF64	C5	9F	000C0	PUSHAB	RD_FAB	: 0918	
	00000000G	00		01	FB	000C4	CALLS	#1, SYSS\$CLOSE		
		09		50	E8	000CB	BLBS	STATUS, 8\$		
				50	DD	000CE	PUSHL	STATUS		
	00000000G	00		01	FB	000D0	CALLS	#1, LIB\$SIGNAL		
	00000000G	00		00	FB	000D7	CALLS	#0, MAIL\$DOWN_PRIV	: 0919	
			FED8	C5	9F	000DE	PUSHAB	EXITBLOCK	: 0920	
	00000000G	00		01	FB	000E2	CALLS	#1, SYSS\$CANEXH		
	00000000G	00		00	FB	000E9	CALLS	#0, MAIL\$CLR_DIR_FLAG	: 0924	
			FEE8	C5	D4	000F0	CLRL	MODIFIED	: 0925	
				65	7C	000F4	CLRQ	MSG_BUF_SIZ	: 0927	
			F8	A5	7C	000F6	CLRQ	MSG_BUF_ASIZ	: 0926	
				66	D4	000F9	CLRL	MAIL\$MSG_NUM	: 0930	
			04	A6	B4	000FB	CLRW	MAIL\$MSG_RSIZ	: 0931	
			08	A6	D4	000FE	CLRL	MAIL\$MSG_RADR	: 0932	
	FC	A6	007E8022	8F	D0	00101	MOVL	#8290338, MAIL\$MSG_STS	: 0933	
		67	1801	8F	AA	00109	BICW2	#6145, MAIL\$GL_FLAGS	: 0936	
				04	00	0010E	RET		: 0937	

; Routine Size: 271 bytes, Routine Base: \$CODE\$ + 04EF

```

805 0938 1 ROUTINE OPEN_MAIL_FILE (SIGNAL_FLAG,RE_OPEN) =
806 0939 1 +-
807 0940 1 FUNCTIONAL DESCRIPTION:
808 0941 1
809 0942 1 This routine opens the mail file
810 0943 1
811 0944 1 Assumes RD_FAB initialized to open the file
812 0945 1 --
813 0946 2 BEGIN
814 0947 2
815 0948 2 LOCAL
816 0949 2 DESC : VECTOR[2,LONG],
817 0950 2 STATUS;
818 0951 2
819 0952 2 MAIL$GL_FLAGS[MAIF_V_ISAM] = 0;
820 0953 2 DESC[0] = DESC[1] = 0;
821 0954 2 STATUS = $OPEN(FAB = RD_FAB);
822 0955 2 CH$MOVE(DSC$C_S_BLN,UTIL$GETFILENAME(RD_FAB),DESC);
823 0956 2 STR$COPY_DX(RD_ESNAM_DESC,DESC);
824 0957 2 IF NOT .STATUS
825 0958 3 THEN BEGIN
826 0959 3 IF .STATUS EQL RMS$_SHR
827 0960 4 THEN BEGIN
828 0961 4 RD_FAB[FAB$B_FAC] = FAB$M_GET OR FAB$M_PUT OR FAB$M_TRN;
829 0962 4 RD_FAB[FAB$L_FOP] = FAB$M_TEF OR FAB$M_SQO;
830 0963 4 RD_FAB[FAB$B_SHR] = 0;
831 0964 4 RD_RAB[RAB$B_RAC] = RAB$C_SEQ;
832 0965 4 RD_RAB[RAB$L_ROP] = RAB$M_RAH OR RAB$M_WBH OR RAB$M_TPT;
833 0966 4 IF NOT .RE_OPEN
834 0967 4 THEN RETURN TRUE;
835 P 0968 4 IF _ERR($OPEN(FAB=RD_FAB));,
836 P 0969 4 IF .SIGNAL_FLAG
837 P 0970 4 THEN UTIL$REPORT_IO_ERROR(RD_FAB);
838 0971 4 RETURN .STATUS);
839 0972 4 END
840 0973 3 ELSE RETURN UTIL$REPORT_IO_ERROR(RD_FAB);
841 0974 2 END;
842 0975 2 IF .RD_FAB[FAB$B_ORG] EQL FAB$C_IDX
843 0976 3 THEN BEGIN
844 0977 3 MAIL$GL_FLAGS[MAIF_V_ISAM] = 1; !Flag ISAM file
845 0978 3 IF .MAIL$SL_RECBUF EQL 0
846 P 0979 3 THEN IF _ERR(LIB$GET_VM(%REF(MFR_C_MAXLEN),MAIL$SL_RECBUF));,
847 P 0980 3 IF .SIGNAL_FLAG THEN SIGNAL(.STATUS);
848 0981 3 RETURN .STATUS);
849 0982 3
850 0983 2 RD_RAB[RAB$L_UBF] = .MAIL$SL_RECBUF;
851 0984 2 END;
852 0985 2
853 P 0986 2 IF _ERR($CONNECT(RAB=RD_RAB));,
854 P 0987 2 IF .SIGNAL_FLAG
855 P 0988 2 THEN UTIL$REPORT_IO_ERROR(RD_RAB);
856 P 0989 2 $CLOSE(FAB=RD_FAB);
857 0990 2 RETURN .STATUS);
858 0991 2
859 0992 2 IF .MAIL$GL_FLAGS[MAIF_V_ISAM]
860 0993 2 THEN MAIL$ISAM_INIT();
861 0994 2

```


: 862
: 863

0995 2 RETURN TRUE
0996 1 END;

.EXTRN SYSS\$OPEN, SYSS\$CONNECT

				OFFC	00000	OPEN_MAIL	FILE:				
							.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	: 0938		
		5B	00000000G	00	9E	00002	MOVAB	SYSS\$OPEN, R11			
		5A	00000000G	00	9E	00009	MOVAB	MAIL\$GL_FLAGS, R10			
		59	00000000'	00	9E	00010	MOVAB	MAIL\$RECBUF, R9			
		58	00000000G	00	9E	00017	MOVAB	UTIL\$REPORT_IO_ERROR, R8			
		57	00000000'	00	9E	0001E	MOVAB	RD_FAB, R7			
		5E		0C	C2	00025	SUBL2	#12, SP			
		6A		01	8A	00028	BICB2	#1, MAIL\$GL_FLAGS	: 0952		
				04	AE	7C	0002B	CLRQ	DESC	: 0953	
					57	DD	0002E	PUSHL	R7	: 0954	
		6B		01	FB	00030	CALLS	#1, SYSS\$OPEN			
		56		50	D0	00033	MOVL	R0, STATUS			
					57	DD	00036	PUSHL	R7	: 0955	
		04	00000000G	00	01	FB	00038	CALLS	#1, UTIL\$GETFILENAME		
	04	AE		60	08	28	0003F	MOVCB	#8, (R0), DESC		
					04	AE	9F	00044	PUSHAB	DESC	: 0956
					98	A7	9F	00047	PUSHAB	RD_ESNAM_DESC	
			00000000G	00	02	FB	0004A	CALLS	#2, STR\$COPY_DX		
				41	56	E8	00051	BLBS	STATUS, 3\$: 0957	
			000186B4	8F	56	D1	00054	CMPL	STATUS, #100020	: 0959	
					32	12	0005B	BNEQ	2\$		
		04		A7	10000040	8F	D0	0005D	MOVL	#268435520, RD_FAB+4	: 0962
		16		A7		13	B0	00065	MOVW	#19, RD_FAB+22	: 0961
					6E	A7	94	00069	CLRB	RD_RAB+30	: 0964
		54		A7	0602	8F	3C	0006C	MOVZWL	#1538, RD_RAB+4	: 0965
				03	08	AC	E8	00072	BLBS	RE_OPEN, T\$: 0966
					0085	31	00076	BRW	9\$		
					57	DD	00079	1\$: PUSHL	R7	: 0971	
		6B			01	FB	0007B	CALLS	#1, SYSS\$OPEN		
		52			50	D0	0007E	MOVL	R0, STATUS		
		11			52	E8	00081	BLBS	STATUS, 3\$		
		68		04	AC	E9	00084	BLBC	SIGNAL_FLAG, 7\$		
					57	DD	00088	PUSHL	R7		
		68			01	FB	0008A	CALLS	#1, UTIL\$REPORT_IO_ERROR		
					61	11	0008D	BRB	7\$		
					57	DD	0008F	2\$: PUSHL	R7	: 0973	
		68			01	FB	00091	CALLS	#1, UTIL\$REPORT_IO_ERROR		
					04	04	00094	RET			
		20		1D	A7	91	00095	3\$: CMPB	RD_FAB+29, #32	: 0975	
					32	12	00099	BNEQ	5\$		
		6A			01	88	0009B	BISB2	#1, MAIL\$GL_FLAGS	: 0977	
					69	D5	0009E	TSTL	MAIL\$RECBUF	: 0978	
					27	12	000A0	BNEQ	4\$		
					59	DD	000A2	PUSHL	R9	: 0981	
		04		AE	0800	8F	3C	000A4	MOVZWL	#2048, 4(SP)	
					04	AE	9F	000AA	PUSHAB	4(SP)	
			00000000G	00	02	FB	000AD	CALLS	#2, LIB\$GET_VM		
				52	50	D0	000B4	MOVL	R0, STATUS		
				0F	52	E8	000B7	BLBS	STATUS, 4\$		

	32	04	AC	E9	000BA	BLBC	SIGNAL_FLAG, 7\$:	
			52	DD	000BE	PUSHL	STATUS	:	
00000000G	00		01	FB	000C0	CALLS	#1, LIB\$SIGNAL	:	
	74	A7	27	11	000C7	BRB	7\$:	
			69	D0	000C9	4\$:	MOVL	MAIL\$L_RECBUF, RD_RAB+36	0983
00000000G	00	50	A7	9F	000CD	5\$:	PUSHAB	RD_RAB	0990
			01	FB	000D0		CALLS	#1, SYSS\$CONNECT	
			50	D0	000D7		MOVL	R0, STATUS	
			52	E8	000DA		BLBS	STATUS, 8\$	
	06	04	AC	E9	000DD	BLBC	SIGNAL_FLAG, 6\$:	
		50	A7	9F	000E1	PUSHAB	RD_RAB	:	
	68		01	FB	000E4	CALLS	#1, UTIL\$REPORT_IO_ERROR	:	
00000000G	00		57	DD	000E7	6\$:	PUSHL	R7	
			01	FB	000E9		CALLS	#1, SYSS\$CLOSE	
			52	D0	000F0	7\$:	MOVL	STATUS, R0	
					04	000F3	RET	:	
00000000G	07		6A	E9	000F4	8\$:	BLBC	MAIL\$GL_FLAGS, 9\$	0992
			00	FB	000F7		CALLS	#0, MAIL\$ISAM_INIT	0993
			01	D0	000FE	9\$:	MOVL	#1, R0	0995
			04	00101			RET	:	0996

; Routine Size: 258 bytes, Routine Base: \$CODE\$ + 05FE

```

: 865      0997 1 GLOBAL ROUTINE MAIL$RENAME_SUCCESS (OLD_SPEC,NEW_SPEC) =
: 866      0998 1 |++
: 867      0999 1 | FUNCTIONAL DESCRIPTION:
: 868      1000 1 |
: 869      1001 1 |           This routine is called by LIB$RENAME_FILE on a successful RENAME
: 870      1002 1 | --
: 871      1003 2 BEGIN
: 872      1004 2
: 873      1005 2 MAP
: 874      1006 2     OLD_SPEC · REF $BBLOCK;
: 875      1007 2
: 876      1008 2 SIGNAL(MAIL$_RENAMED,2,.OLD_SPEC,.NEW_SPEC);
: 877      1009 2
: 878      1010 2 RETURN TRUE
: 879      1011 1 END;

```

```

                                0000 00000          .ENTRY MAIL$RENAME_SUCCESS, Save nothing
                                7E      04    AC  7D 00002          MOVQ  OLD_SPEC, -(SP)          : 0997
                                02      DD 00006          PUSHL #2                      : 1008
                                007E12A1 8F  DD 00008          PUSHL #8262305                :
                                00000000G 00  04  FB 0000E          CALLS #4, LIB$SIGNAL           :
                                50      01  D0 00015          MOVL  #1, R0                   : 1010
                                04      04  00018          RET                             : 1011

```

; Routine Size: 25 bytes, Routine Base: \$CODE\$ + 0700

```

: 881 1012 1 GLOBAL ROUTINE MAIL$RENAME_ERROR (OLD_SPEC,NEW_SPEC,RMSSTS,RMSSTV,ERROR_SOURCE) =
: 882 1013 1 |++
: 883 1014 1 | FUNCTIONAL DESCRIPTION:
: 884 1015 1 |
: 885 1016 1 | Error routine called by LIB$RENAME_FILE on an error
: 886 1017 1 |--
: 887 1018 2 BEGIN
: 888 1019 2 MAP
: 889 1020 2 RMSSTS : REF VECTOR[,LONG],
: 890 1021 2 RMSSTV : REF VECTOR[,LONG];
: 891 1022 2
: 892 1023 2 CASE .ERROR_SOURCE FROM 0 TO 2
: 893 1024 2 OF SET
: 894 1025 2
: 895 1026 2 [0] : !Error searching for old_file
: 896 1027 2 | IF .RMSSTS[0] EQL RMSS_FNF
: 897 1028 2 | THEN RETURN TRUE
: 898 1029 2 | ELSE
: 899 1030 2 | SIGNAL(MAIL$_SEARCHFAIL,1,.OLD_SPEC,
: 900 1031 2 | .RMSSTS[0],.RMSSTV[0]);
: 901 1032 2
: 902 1033 2 [1,2] : !Error parsing new file
: 903 1034 2 | SIGNAL(MAIL$_SEARCHFAIL,1,.NEW_SPEC,.RMSSTS[0],.RMSSTV[0]);
: 904 1035 2 TES;
: 905 1036 2
: 906 1037 2 RETURN TRUE
: 907 1038 1 END;

```

				0000 0000		.ENTRY MAIL\$RENAME_ERROR, Save nothing	: 1012
	02	00	14	AC CF 00002		CASEL ERROR_SOURCE, #0, #2	: 1023
	0011	0011	0006	00007	1\$:	.WORD 2\$-1\$,-	
						3\$-1\$,-	
						3\$-1\$	
			10	BC DD 0000D	2\$:	PUSHL @RMSSTV	: 1031
			0C	BC DD 00010		PUSHL @RMSSTS	
			04	AC DD 00013		PUSHL OLD_SPEC	: 1030
				09 11 00016		BRB 4\$	
			10	BC DD 00018	3\$:	PUSHL @RMSSTV	: 1034
			0C	BC DD 0001B		PUSHL @RMSSTS	
			08	AC DD 0001E		PUSHL NEW_SPEC	
				01 DD 00021	4\$:	PUSHL #1	
		007E123A		8F DD 00023		PUSHL #8262202	
	00000000G	00		05 FB 00029		CALLS #5, LIB\$SIGNAL	
		50		01 DD 00030		MOVL #1, R0	: 1037
				04 DD 00033		RET	: 1038

: Routine Size: 52 bytes, Routine Base: \$CODE\$ + 0719

```

1039 1 GLOBAL ROUTINE MAIL$OPEN_FILE(FILE_DESC) =
1040 1  +-
1041 1  FUNCTIONAL DESCRIPTION:
1042 1
1043 1      This routine opens the named message file and reads it into the
1044 1      message buffer.
1045 1
1046 1  !INPUTS:
1047 1
1048 1      FILE_DESC = Descriptor of file to open. if 0, default mail file opened
1049 1      MAIL$T_MAILDEF = User's default device/directory string
1050 1
1051 1  !OUTPUTS:
1052 1
1053 1      MSG_BUF_ASIZ = Size of allocated message buffer.
1054 1      MSG_BUF_SIZ = Size of text in message buffer.
1055 1      MSG_BUF_ADR = Address of text in message buffer.
1056 1      MSG_ADR = Address of first new message in buffer.
1057 1      MAIL$Q_MSG_FILE = Message file name desc.
1058 1      MAIL$Q_MSG_NUM = Current message number.
1059 1
1060 1      All failures are signalled as well as returned.
1061 1  --
1062 2 BEGIN
1063 2
1064 2 MAP
1065 2      FILE_DESC : REF $BLOCK;
1066 2
1067 2 LOCAL
1068 2      DESC : VECTOR[2, LONG],
1069 2      DESC_1 : VECTOR[2, LONG],
1070 2      NEW_FORMAT,           ! file is ISAM
1071 2      PREV_REC,             ! Previous record address
1072 2      SPACE_LEFT,         ! Space left in buffer
1073 2      STATUS;             ! Routine status
1074 2
1075 2 IF .RD_RSNAM_BUF EQL 0
1076 2 P THEN IF_ERR(LIB$GET_VM(%REF(NAM$C_MAXRSS), RD_RSNAM_BUF));
1077 2      RETURN SIGNAL(.STATUS);
1078 2 IF .RD_ESNAM_BUF EQL 0
1079 2 P THEN IF_ERR(LIB$GET_VM(%REF(NAM$C_MAXRSS), RD_ESNAM_BUF));
1080 2      RETURN SIGNAL(.STATUS);
1081 2
1082 2      Close any currently open file and zero all context.
1083 2
1084 2 MAIL$CLOSE_FILE();
1085 2
1086 2      Initialize RMS data structures
1087 2
1088 2 P $FAB_INIT(FAB=RD_FAB,
1089 2 P      FNS=.FILE_DESC[DSC$W_LENGTH],
1090 2 P      FNA=.FILE_DESC[DSC$A_POINTER],
1091 2 P      DNS = .MAIL$Q_MAILDEF[DSC$W_LENGTH],
1092 2 P      DNA = .MAIL$Q_MAILDEF[DSC$A_POINTER],
1093 2 P      CTX = MAIL$OPENIN,
1094 2 P      FAC=(DEL,GET,PUT,UPD),
1095 2 P      SHR=(DEL,GET,PUT,UPD),

```

```

: 966      1096      2          NAM=RD_NAM);
: 967      1097      2
: 968      1098      2 RD_FAB[FAB$V_DFW] = .MAIL$B_DFW;
: 969      1099      2
: 970      P 1100      2 $NAM_INIT(NAM=RD_NAM,
: 971      P 1101      2          RSS=NAM$C_MAXRSS,
: 972      P 1102      2          RSA=.RD_RSNAME_BUF,
: 973      P 1103      2          ESS=NAM$C_MAXRSS,
: 974      1104      2          ESA=.RD_ESNAME_BUF);
: 975      1105      2
: 976      P 1106      2 $RAB_INIT(RAB=RD_RAB,
: 977      P 1107      2          FAB=RD_FAB,
: 978      P 1108      2          CTX=MAIL$_OPENIN,
: 979      P 1109      2          RAC=KEY,
: 980      1110      2          USZ=MFR_C_MAXLEN);
: 981      1111      2
: 982      1112      2 IF .FILE_DESC[DSC$W_LENGTH] EQL 0
: 983      1113      2     OR CR$EQL(.FILE_DESC[DSC$W_LENGTH],.FILE_DESC[DSC$A_POINTER],
: 984      1114      2         .MAIL$Q_MSGNAME[DSC$W_LENGTH],.MAIL$Q_MSGNAME[DSC$A_POINTER],0)
: 985      1115      2 THEN BEGIN
: 986      1116      2
: 987      1117      2     Open the new message file. Call routine to concatenate new msgs together
: 988      1118      2     if sequential file. if isam file, always just open.
: 989      1119      2
: 990      1120      2     STR$COPY_DX(MAIL$Q_MSG_FILE,.MAIL$Q_MSGNAME);
: 991      1121      2     RD_FAB[FAB$B_FNS] = .MAIL$Q_MSGNAME[DSC$W_LENGTH];
: 992      1122      2     RD_FAB[FAB$L_FNA] = .MAIL$Q_MSGNAME[DSC$A_POINTER];
: 993      1123      2     STATUS = OPEN_NEWFILE();
: 994      1124      2     IF .STATUS EQ[ RMS$_FNF                                ! If file not found
: 995      1125      2     THEN BEGIN
: 996      1126      2         SIGNAL(MAIL$_NOMSGS);                                ! Signal no messages
: 997      1127      2         RETURN MAIL$_NOMSGS                                    ! Return
: 998      1128      2     END
: 999      1129      2     ELSE IF NOT .STATUS
: 1000     1130      2     THEN RETURN .STATUS;
: 1001     1131      2     MAIL$GL_FLAGS[MAIF_V_DFMSG] = 1;
: 1002     1132      2     END
: 1003     1133      2 ELSE BEGIN
: 1004     1134      2     STR$COPY_DX(MAIL$Q_MSG_FILE,.FILE_DESC);
: 1005     1135      2     STATUS = OPEN_MAIL_FILE(1,1);
: 1006     1136      2     IF NOT .STATUS THEN RETURN .STATUS;
: 1007     1137      2     END;
: 1008     1138      2     MAIL$Q_MSG_FNM[DSC$W_LENGTH] = .RD_NAM[NAM$B_NAME];
: 1009     1139      2     MAIL$Q_MSG_FNM[DSC$A_POINTER] = .RD_NAM[NAM$C_NAME];
: 1010     1140      2
: 1011     1141      2     IF NOT .MAIL$GL_FLAGS[MAIF_V_ISAM]                        !Isam handled differently, thanks
: 1012     1142      2     THEN BEGIN
: 1013     1143      2
: 1014     1144      2     Allocate a buffer to read-in the message text
: 1015     1145      2
: 1016     1146      2     IF .RD_FAB[FAB$L_ALQ] EQL 0                                ! Return if file empty
: 1017     1147      2     THEN
: 1018     1148      2         (SIGNAL(MAIL$_NOMSGS); $CLOSE(FAB=RD_FAB); RETURN MAIL$_NOMSGS);
: 1019     1149      2     MSG_BUF_ASIZ = .RD_FAB[FAB$L_ALQ] * 512;                    ! Compute size in bytes
: 1020     P 1150      2     IF_ERR(-LIB$GET_VMT(MSG_BUF_ASIZ,MSG_BUF_ADR));
: 1021     P 1151      2     SIGNAL(.STATUS);
: 1022     P 1152      2     $CLOSE(FAB = RD_FAB);
```

```
1023 1153 3 RETURN .STATUS);
1024 1154 3 |
1025 1155 3 | Read all the text records from the file into the message buffer
1026 1156 3 |
1027 1157 3 SPACE_LEFT = .MSG_BUF_ASIZ;
1028 1158 3 PREV_REC = .MSG_BUF_ADR; .PREV_REC = 0;
1029 1159 3 RD_RAB[RAB$L_UBF] = .MSG_BUF_ADR + RECSIZ_SIZ;
1030 1160 3 RD_RAB[RAB$W_USZ] = MIN(.SPACE_LEFT - RECSIZ_SIZ,MAXRECSIZ);
1031 1161 3 RD_RAB[RAB$L_CTX] = MAIL$READERR;
1032 1162 4 WHILE (STATUS = $GET(RAB = RD_RAB,ERR=UTIL$REPORT_IO_ERROR))
1033 1163 4 AND (.SPACE_LEFT GTR 0)
1034 1164 4 DO BEGIN
1035 1165 4 RECSIZ(.RD_RAB[RAB$L_UBF] - RECSIZ_SIZ) = .RD_RAB[RAB$W_RSZ];
1036 1166 4 |
1037 1167 4 | If first new message, save address of message
1038 1168 4 |
1039 1169 5 IF MSGDELIM(.PREV_REC)
1040 1170 4 THEN
1041 1171 4 IF .MSG_ADR EQL 0
1042 1172 5 THEN BEGIN
1043 1173 5 MAIL$L_MSG_NUM = .MAIL$L_MSG_NUM + 1;
1044 1174 5 IF CHRCHAR(NEWMSGFLAG(.PREV_REC)) EQL '*'
1045 1175 5 THEN
1046 1176 5 MSG_ADR = .PREV_REC;
1047 1177 4 END;
1048 1178 4 PREV_REC = .RD_RAB[RAB$L_RBF] - RECSIZ_SIZ;
1049 1179 4 |
1050 1180 4 | Compute space left and address for next record
1051 1181 4 |
1052 1182 4 SPACE_LEFT = .SPACE_LEFT - .RD_RAB[RAB$W_RSZ] - RECSIZ_SIZ;
1053 1183 4 MSG_BUF_SIZ = .MSG_BUF_SIZ + .RD_RAB[RAB$W_RSZ] + RECSIZ_SIZ;
1054 1184 4 RD_RAB[RAB$L_UBF] = .RD_RAB[RAB$L_UBF] +
1055 1185 4 .RD_RAB[RAB$W_RSZ] + RECSIZ_SIZ;
1056 1186 4 RD_RAB[RAB$W_USZ] = MIN(.SPACE_LEFT - RECSIZ_SIZ,MAXRECSIZ);
1057 1187 4 END;
1058 1188 3 IF .STATUS NEQ RMSS_EOF
1059 1189 4 THEN BEGIN
1060 1190 4 MAIL$CLOSE FILE();
1061 1191 4 RETURN .STATUS;
1062 1192 3 END;
1063 1193 3 |
1064 1194 3 | If no new messages found, set current message to first message. Also, if
1065 1195 3 | this is the new message file, clear the user's new message flag.
1066 1196 3 |
1067 1197 3 IF .MSG_ADR EQL 0 ! If no new messages,
1068 1198 4 THEN BEGIN ! Position to first one
1069 1199 4 MSG_ADR = .MSG_BUF_ADR;
1070 1200 4 MAIL$L_MSG_NUM = 1;
1071 1201 3 END;
1072 1202 3 %IF NOT LEAVE_SEQ
1073 1203 3 %THEN
1074 1204 3 IF .MAIL$GL_FLAGS[MAIF_V_DFMSG]
1075 1205 4 THEN BEGIN
1076 1206 4 DESC[0] = .RD_NAM[NAM$B_NAME];
1077 1207 4 DESC[1] = .RD_NAM[NAM$L_NAME];
1078 1208 4 DESC_1[0] = .RD_NAM[NAM$B_RSL] - .RD_NAM[NAM$B_VER];
1079 1209 4 DESC_1[1] = .RD_NAM[NAM$L_RSA];
```

```

: 1080      1210 4      MAIL$COPY MSGS(DESC_1,DESC,(MCR_M_NIL OR MCR_M_ALL),TELL CVTFILE);
: 1081      1211 4      MAIL$CLOSE_FILE(); !Close the sequential file
: 1082      1212 4      LIB$RENAME_FILE(MAIL$Q RD_DESC, !Rename the old sequential file
: 1083      1213 4      $DESCRIPTOR('OLD'),0,0,0,
: 1084      1214 4      MAIL$RENAME_SUCCESS,MAIL$RENAME_ERROR);
: 1085      1215 4      MAIL$GET_INPUT(DYN_DESC,MAIL$GETMSG(MAIL$RETCONT));
: 1086      1216 4      RETURN MAIL$OPEN_FILE(.FILE_DESC); !and open the isam version
: 1087      1217 3      END;
: 1088      1218 3      %FI
: 1089      1219 2      END;
: 1090      1220 2      MAIL$GL_FLAGS[MAIF_V_FILOPN] = 1; !Sequential mail file
: 1091      1221 2      RETURN TRUE !Flag file open for business
: 1092      1222 1      END;

```

```

44 4C 4F 2E 0074D P.AAG: .ASCII \.OLD\
00000004 00751 .BLKB 3
00000000' 00754 P.AAF: .LONG 4
00000000' 00758 .ADDRESS P.AAG

```

```

$RMS_PTR= RD_FAB
$RMS_PTR= RD_NAM
$RMS_PTR= RD_RAB
.EXTRN SYSSGET

```

OFFC 00000

```

.ENTRY MAIL$OPEN FILE, Save R2,R3,R4,R5,R6,R7,R8,- : 1039
R9,R10,R11

```

```

5B 00000000G 00 9E 00002 MOVAB LIB$SIGNAL, R11
5A 00000000' 00 9E 00009 MOVAB MAIL$Q MSG_FILE, R10
59 FD7F CF 9E 00010 MOVAB MAIL$CLOSE_FILE, R9
58 00000000' 00 9E 00015 MOVAB $RMS_PTR, R8
5E 14 C2 0001C SUBL2 #20, SP
90 A8 D5 0001F TSTL RD_RSNAME_BUF
15 12 00022 BNEQ 1$
90 A8 9F 00024 PUSHAB RD_RSNAME_BUF
04 AE FF 8F 9A 00027 MOVZBL #255, 4(SP)
00000000G 00 04 AE 9F 0002C PUSHAB 4(SP)
1A 50 FB 0002F CALLS #2, LIB$GET_VM
94 A8 D5 00039 1$: TSTL RD_ESNAME_BUF
1B 12 0003C BNEQ 3$
94 A8 9F 0003E PUSHAB RD_ESNAME_BUF
04 AE FF 8F 9A 00041 MOVZBL #255, 4(SP)
00000000G 00 04 AE 9F 00046 PUSHAB 4(SP)
06 50 FB 00049 CALLS #2, LIB$GET_VM
50 DD 00053 2$: BLBS STATUS, 3$
6B 01 FB 00055 PUSHL STATUS
04 00058 CALLS #1, LIB$SIGNAL
RET
69 00 FB 00059 3$: CALLS #0, MAIL$CLOSE_FILE
0050 8F 00 00 2C 0005C MOVCS #0, (SP), #0, #80, $RMS_PTR
68 00063
16 A8 5003 8F B0 00064 MOVW #20483, $RMS_PTR
18 A8 0F0F 8F B0 00069 MOVW #3855, $RMS_PTR+22
007E109A 8F D0 0006F MOVL #8261786, $RMS_PTR+24
1F A8 02 90 00077 MOVAB #2, $RMS_PTR+3T

```

```

: 1075
: 1077
: 1078
: 1080
: 1084
: 1096

```


			28	A8	A0	A8	9E	0007B	MOVAB	RD NAM, \$RMS_PTR+40		
				56	04	AC	DO	00080	MOVL	FILE_DESC, R6		
			2C	A8	04	A6	DO	00084	MOVL	4(R6), \$RMS_PTR+44		
			30	A8	00000000G	00	DO	00089	MOVL	MAIL\$Q_MAILDEF+4, \$RMS_PTR+48		
			34	A8		66	90	00091	MOVB	(R6), \$RMS_PTR+52		
			35	A8	00000000G	00	90	00095	MOVB	MAIL\$Q_MAILDEF, \$RMS_PTR+53		
04	AB	01		05	00000000G	00	FO	0009D	INSV	MAIL\$B_DFW, #5, #1, RD FAB+4	1098	
0060	BF	00		6E		00	2C	000A7	MOVCS	#0, (SP), #0, #96, \$RMS_PTR	1104	
					A0	A8		000AE				
			A0	A8	6002	8F	BO	000B0	MOVW	#24578, \$RMS_PTR		
			A2	A8		01	8E	000B6	MNEGB	#1, \$RMS_PTR+2		
			A4	A8	90	AB	DO	000BA	MOVL	RD_RSNAM_BUF, \$RMS_PTR+4		
			AA	A8		01	8E	000BF	MNEGB	#1, \$RMS_PTR+10		
0044	BF	00		AC	94	A8	DO	000C3	MOVL	RD_ESNAM_BUF, \$RMS_PTR+12		
				6E		00	2C	000C8	MOVCS	#0, (SP), #0, #68, \$RMS_PTR	1110	
					50	A8		000CF				
			50	A8	4401	8F	BO	000D1	MOVW	#17409, \$RMS_PTR		
			63	A8	007E109A	8F	DO	000D7	MOVL	#8261786, \$RMS_PTR+24		
			6E	A8		01	90	000DF	MOVB	#1, \$RMS_PTR+30		
			70	A8	0800	8F	BO	000E3	MOVW	#2048, \$RMS_PTR+32		
			008C	C8		68	9E	000E9	MOVAB	RD FAB, \$RMS_PTR+60		
						66	B5	000EE	TSTW	(R6)	1112	
						10	13	000F0	BEQL	4\$		
F7B5	CF	00	04	50	F7C2	CF	DO	000F2	MOVL	MAIL\$Q_MSGNAME+4, R0	1114	
				66		2D	000F7	CMPCS	(R6), #4(R6), #0, MAIL\$Q_MSGNAME, (R0)	1113		
						60		000FF				
						43	12	00100	BNEQ	6\$		
					F7AE	CF	9F	00102	4\$: PUSHAB	MAIL\$Q_MSGNAME	1120	
						5A	DD	00106	PUSHL	R10		
			00000000G	00		02	FB	00108	CALLS	#2, STR\$COPY_DX		
				34	F7A1	CF	90	0010F	MOVB	MAIL\$Q_MSGNAME, RD FAB+52	1121	
				2C	F79F	CF	DO	00115	MOVL	MAIL\$Q_MSGNAME+4, RD_FAB+44	1122	
			00000000V	00		00	FB	0011B	CALLS	#0, OPEN_NEWFILE	1123	
				57		50	DO	00122	MOVL	R0, STATUS		
			00018292	8F		57	D1	00125	CMPL	STATUS, #98962	1124	
						0B	12	0012C	BNEQ	5\$		
					007E804A	8F	DD	0012E	PUSHL	#8290378	1126	
				6B		01	FB	00134	CALLS	#1, LIB\$SIGNAL		
						54	11	00137	BRB	10\$	1127	
				20		57	E9	00139	5\$: BLBC	STATUS, 7\$	1129	
			00000000G	00		10	88	0013C	BISB2	#16, MAIL\$GL_FLAGS+1	1131	
						1D	11	00143	BRB	8\$	1112	
						56	DD	00145	6\$: PUSHL	R6	1134	
						5A	DD	00147	PUSHL	R10		
			00000000G	00		02	FB	00149	CALLS	#2, STR\$COPY_DX		
						01	DD	00150	PUSHL	#1	1135	
						01	DD	00152	PUSHL	#1		
			010F	C9		02	FB	00154	CALLS	#2, OPEN_MAIL_FILE		
				57		50	DO	00159	MOVL	R0, STATUS		
				03		57	E8	0015C	7\$: BLBS	STATUS, 8\$	1136	
						0138	31	0015F	BRW	19\$		
				08	DB	A8	9B	00162	8\$: MOVZBW	RD_NAM+59, MAIL\$Q_MSG_FNM	1138	
				0C	EC	A8	DO	00167	MOVL	RD_NAM+76, MAIL\$Q_MSG_FNM+4	1139	
				03	00000000G	00	E9	0016C	BLBC	MAIL\$GL_FLAGS, 9\$	1141	
						01AD	31	00173	BRW	22\$		
						10	A8	00176	9\$: TSTL	RD FAB+16	1146	
						1A	12	00179	BNEQ	11\$		

			007E804A	8F	DD	0017B		PUSHL	#8290378		1148
	6B			01	FB	00181		CALLS	#1, LIB\$SIGNAL		
		00000000G	00	58	DD	00184		PUSHL	R8		
			50	01	FB	00186		CALLS	#1, SYSS\$CLOSE		
			007E804A	8F	D0	0018D	10\$:	MOVL	#8290378, R0		
				04	00194			RET			
0094	C8	10	A8	09	78	00195	11\$:	ASHL	#9, RD FAB+16, MSG_BUF_ASIZ		1149
			0098	C8	9F	0019C		PUSHAB	MSG_BUF_ADR		1153
			0094	C8	9F	001A0		PUSHAB	MSG_BUF_ASIZ		
		00000000G	00	02	FB	001A4		CALLS	#2, LIB\$GET_VM		
			52	50	D0	001AB		MOVL	R0, STATUS		
			12	52	E8	001AE		BLBS	STATUS, 12\$		
				52	DD	001B1		PUSHL	STATUS		
			6B	01	FB	001B3		CALLS	#1, LIB\$SIGNAL		
				58	DD	001B6		PUSHL	R8		
		00000000G	00	01	FB	001B8		CALLS	#1, SYSS\$CLOSE		
			50	52	D0	001BF		MOVL	STATUS, R0		
				04	001C2			RET			
			52	0094	C8	D0	001C3	12\$:	MOVL	MSG_BUF_ASIZ, SPACE_LEFT	1157
			50	0098	C8	D0	001C8		MOVL	MSG_BUF_ADR, R0	1158
			53	50	D0	001CD		MOVL	R0, PREV_REC		
				63	D4	001D0		CLRL	(PREV_REC)		
	74	A8	01	A0	9E	001D2		MOVAB	1(R0), RD RAB+36		1159
				50	A2	9E	001D7		MOVAB	-1(R2), R0	1160
		000000FF	8F	50	D1	001DB		CPL	R0, #255		
				04	15	001E2		BLEQ	13\$		
			50	FF	8F	9A	001E4		MOVZBL	#255, R0	
	70	A8		50	B0	001E8	13\$:	MOVW	R0, RD RAB+32		
	68	A8	007E10B2	8F	D0	001EC		MOVL	#82618T0, RD RAB+24		1161
			00000000G	00	9F	001F4	14\$:	PUSHAB	UTIL\$REPORT_TO_ERROR		1162
				50	A8	9F	001FA		PUSHAB	RD_RAB	
		00000000G	00	02	FB	001FD		CALLS	#2, SYSS\$GET		
			57	50	D0	00204		MOVL	R0, STATUS		
			03	57	E8	00207		BLBS	STATUS, 15\$		
				0081	31	0020A		BRW	18\$		
				52	D5	0020D	15\$:	TSTL	SPACE_LEFT		1163
				7D	15	0020F		BLEQ	18\$		
			54	74	A8	D0	00211		MOVL	RD_RAB+36, R4	1165
			51	72	A8	3C	00215		MOVZWL	RD_RAB+34, R1	
	FF	A4		51	9C	00219		MOVB	R1, -1(R4)		
			50	63	9A	0021D		MOVZBL	(PREV_REC), R0		1169
			01	50	91	00220		CMPB	R0, #T		
				32	12	00223		BNEQ	16\$		
			0C	01	A3	91	00225		CMPB	1(PREV_REC), #12	
				2C	12	00229		BNEQ	16\$		
			04	02	A3	91	0022B		CMPB	2(PREV_REC), #4	
				26	1B	0022F		BLEQU	16\$		
	6D6F7246	8F		03	A3	D1	00231		CPL	3(PREV_REC), #1836020294	
				1C	12	00239		BNEQ	16\$		
				00A0	C8	D5	0023B		TSTL	MSG_ADR	1171
				16	12	0023F		BNEQ	16\$		
				14	AA	D6	00241		INCL	MAIL\$ MSG_NUM	1173
			50	01	A043	9E	00244		MOVAB	1(R0)[PREV_REC], PTR	1174
			55	60	9A	00249		MOVZBL	(PTR), R5		
			2A	6540	91	0024C		CMPB	(R5)[PTR], #42		
				05	12	00250		BNEQ	16\$		
	00A0	C8		53	D0	00252		MOVL	PREV_REC, MSG_ADR		1176

53	78	A8	01	C3	00257	16\$:	SUBL3	#1, RD_RAB+40, PREV_REC	1178		
50		52	51	C3	0025C		SUBL3	R1, SPACE_LEFT, R0	1182		
		52	FF	A0	9E	00260	MOVAB	-1(R0), SPACE_LEFT			
		50	009C	C8	D0	00264	MOVL	MSG_BUF_SIZ, R0	1183		
	009C	C8	01	A140	9E	00269	MOVAB	1(RT)[R0], MSG_BUF_SIZ			
	74	A8	01	A144	9E	00270	MOVAB	1(R1)[R4], RD_RAB+36	1185		
		50	FF	A2	9E	00276	MOVAB	-1(R2), R0	1186		
	000000FF	8F	50	D1	0027A		CMPL	R0, #255			
			04	15	00281		BLEQ	17\$			
		50	FF	8F	9A	00283	MOVZBL	#255, R0			
	70	A8	50	B0	00287	17\$:	MOVW	R0, RD_RAB+32			
			FF	66	31	0028B	BRW	14\$	1162		
	0001827A	8F	57	D1	0028E	18\$:	CMPL	STATUS, #98938	1188		
			07	13	00295		BEQL	20\$			
		69	00	FB	00297		CALLS	#0, MAIL\$CLOSE_FILE	1190		
		50	57	D0	0029A	19\$:	MOVL	STATUS, R0	1191		
				04	0029D		RET				
			00A0	C8	D5	0029E	20\$:	TSTL	MSG_ADR	1197	
				0B	12	002A2	BNEQ	21\$			
	00A0	C8	0098	C8	D0	002A4	MOVL	MSG_BUF_ADR, MSG_ADR	1199		
	14	AA		01	D0	002AB	MOVL	#1, MAIL\$MSG_NUM	1200		
	6C 00000000G	00		04	E1	002AF	21\$:	BBC	#4, MAIL\$C_FLAGS+1, 22\$	1204	
		0C		DB	A8	9A	002B7	MOVZBL	RD_NAM+59, DESC	1206	
		10		EC	A8	D0	002BC	MOVL	RD_NAM+76, DESC+4	1207	
				A3	A8	9A	002C1	MOVZBL	RD_NAM+3, R0	1208	
				DD	A8	9A	002C5	MOVZBL	RD_NAM+61, R1		
	04 AE	50		51	C3	002C9	SUBL3	R1, R0, DESC 1			
		08		A4	A8	D0	002CE	MOVL	RD_NAM+4, DESC_1+4	1209	
				E5	A9	9F	002D3	PUSHAB	TECL_CVTFILE	1210	
					0A	DD	002D6	PUSHL	#10		
				14	AE	9F	002D8	PUSHAB	DESC		
				10	AE	9F	002DB	PUSHAB	DESC 1		
	00000000G	00		04	FB	002DE	CALLS	#4, MAIL\$COPY_MSGS			
		69		00	FB	002E5	CALLS	#0, MAIL\$CLOSE_FILE	1211		
				022A	C9	9F	002E8	PUSHAB	MAIL\$RENAME_ERROR	1212	
				0211	C9	9F	002EC	PUSHAB	MAIL\$RENAME_SUCCESS		
					7E	7C	002F0	CLRQ	-(SP)		
					7E	D4	002F2	CLRL	-(SP)		
				FD00	CF	9F	002F4	PUSHAB	P.AAF	1213	
				98	A8	9F	002F8	PUSHAB	MAIL\$RD_DESC	1212	
	00000000G	00		07	FB	002FB	CALLS	#7, LIB\$RENAME_FILE			
				007EDF08	8F	DD	00302	PUSHL	#8314632	1215	
	00000000G	00		01	FB	00308	CALLS	#1, MAIL\$GETMSG			
				50	D3	0030F	PUSHL	R0			
				88	A8	9F	00311	PUSHAB	DYN_DESC		
	00000000G	00		02	FB	00314	CALLS	#2, MAIL\$GET_INPUT			
				56	DD	0031B	PUSHL	R6		1216	
	026D	C9		01	FB	0031D	CALLS	#1, MAIL\$OPEN_FILE			
					04	00322	RET				
	00000000G	00		08	88	00323	22\$:	BISB2	#8, MAIL\$GL_FLAGS+1	1220	
		50		01	D0	0032A	MOVL	#1, R0	1221		
				04	0032D		RET		1222		

: Routine Size: 814 bytes, Routine Base: \$CODE\$ + 075C

: 1093 1223 1

```

1095 1224 1 ROUTINE OPEN_NEWFILE =
1096 1225 1 +-
1097 1226 1 FUNCTIONAL DESCRIPTION:
1098 1227 1
1099 1228 1 This routine is called to open the new mail message file by first
1100 1229 1 concatenating all the new mail message files into one file. This
1101 1230 1 is because there is no file sharing, and if the new mail message
1102 1231 1 file is in use, the mail that is being sent must be put in a different
1103 1232 1 file. This routine copies all the extra files into one file.
1104 1233 1
1105 1234 1 INPUTS:
1106 1235 1
1107 1236 1 RD_FAB[FAB$B_DNS] and RD_FAB[FAB$L_DNA] set to user's default
1108 1237 1 directory specification.
1109 1238 1
1110 1239 1 OUTPUTS:
1111 1240 1
1112 1241 1 RD_FAB is open and RD_RAB is connected to the file.
1113 1242 1 --
1114 1243 2 BEGIN
1115 1244 2 LITERAL
1116 1245 2 FIDS = 64; ! Allow 64 file ID's
1117 1246 2 STRUCTURE
1118 1247 2 FIDVECTOR[FIDNUM,FIDFIELD;N] = ! FID vector
1119 1248 2 [N*6] (FIDVECTOR+((FIDNUM-1)*6)+(FIDFIELD*2)) <0,16,0>;
1120 1249 2 LOCAL
1121 1250 2 STATUS, ! Routine status
1122 1251 2 DESC: VECTOR[2, LONG],
1123 1252 2 FID: FIDVECTOR[FIDS], ! Message file ID's
1124 1253 2 CUR_FID: LONG, ! Current ID
1125 1254 2 FAB: BLOCK[FAB$C_BLN,BYTE], ! Temporary FAB
1126 1255 2 RAB: BLOCK[RAB$C_BLN,BYTE], ! Temporary RAB
1127 1256 2 NAM: $BLOCK[NAM$C_BLN], ! Temporary NAM
1128 1257 2 NAMRSS: $BLOCK[NAM$C_MAXRSS],
1129 1258 2 NAMESS: $BLOCK[NAM$C_MAXRSS],
1130 1259 2 BUFF: BLOCK[MAIL$K_INBUFFSZ,BYTE]; ! Temporary I/O buffer
1131 1260 2
1132 1261 2 STATUS = OPEN_MAIL_FILE(0,0);
1133 1262 2 IF NOT .STATUS
1134 1263 2 THEN RETURN .STATUS;
1135 1264 2 IF .MAIL$GL_FLAGS[MAIF_V_ISAM]
1136 1265 2 THEN RETURN TRUE;
1137 1266 2
1138 1267 2 $NAM_INIT(NAM=RD_NAM,
1139 1268 2 RSS=NAM$C_MAXRSS,
1140 1269 2 RSA=.RD_RSNAME_BUF,
1141 1270 2 ESS=NAM$C_MAXRSS,
1142 1271 2 ESA=.RD_ESNAME_BUF);
1143 1272 2
1144 1273 2 !
1145 1274 2 ! Find all the newmail message files.
1146 1275 2 !
1147 1276 2 CUR_FID = 0; ! Init file ID index
1148 1277 2 RD_FAB[FAB$B_FNS] = %CHARCOUNT(DEFFILENAM,DEFFILETYP,':;*');
1149 1278 2 RD_FAB[FAB$L_FNA] = UPLIT_BYTE(DEFFILENAM,DEFFILETYP,':;*');
1150 1279 2 IF_ERR( $PARSE(FAB = RD_FAB);.RETURN .STATUS); ! Parse the file name
1151 1280 2 !

```

```

1152 1281 2 : Search for all the files and save the file IDs
1153 1282 2
1154 1283 2 WHILE (STATUS = $SEARCH(FAB = RD_FAB)) AND (.CUR_FID LSS FIDS)
1155 1284 2 DO BEGIN
1156 1285 2   STR$COPY_DX(MAIL$Q_RD_DESC,UTIL$GETFILENAME(RD_FAB)); !So we get oldest
1157 1286 2   CUR_FID = .CUR_FID + 1;
1158 1287 2   FID[.CUR_FID,0] = .RD_NAM[NAM$W_FID_NUM];
1159 1288 2   FID[.CUR_FID,1] = .RD_NAM[NAM$W_FID_SEQ];
1160 1289 2   FID[.CUR_FID,2] = .RD_NAM[NAM$W_FID_RVN];
1161 1290 2   END;
1162 1291 2 IF NOT .STATUS AND (.STATUS NEQ RMSS_NMF) THEN RETURN .STATUS;
1163 1292 2
1164 1293 2 : Open the oldest file (the last one found).
1165 1294 2
1166 1295 2 RD_NAM[NAM$W_FID_NUM] = .FID[.CUR_FID,0];           ! Set file ID
1167 1296 2 RD_NAM[NAM$W_FID_SEQ] = .FID[.CUR_FID,1];
1168 1297 2 RD_NAM[NAM$W_FID_RVN] = .FID[.CUR_FID,2];
1169 1298 2 RD_FAB[FAB$V_NAM] = 1;           ! Open by file ID
1170 1299 2 STATUS = $OPEN(FAB = RD_FAB);     ! Open file
1171 1300 2 RD_FAB[FAB$V_NAM] = 0;           ! Clear open by file ID
1172 1301 2 IF NOT .STATUS THEN RETURN .STATUS; ! Check open status
1173 1302 2 IF .RD_FAB[FAB$B_ORG] NEQ FAB$C_SEQ
1174 1303 2 THEN ($CLOSE(FAB=RD_FAB);
1175 1304 2   RETURN SIGNAL(MAIL$ FILECONFUSED,1,MAIL$Q_RD_DESC));
1176 1305 2 MAIL$GL_FLAGS[MAIF_V_FILOPN] = 1;
1177 1306 2 CUR_FID = .CUR_FID - 1;           ! Decrement file ID count
1178 1307 2 RD_RAB[RAB$V_EOF] = 1;           ! Set to end-of-file
1179 1308 2 STATUS = $CONNECT(RAB = RD_RAB); ! Connect stream
1180 1309 2 IF NOT .STATUS
1181 1310 2 THEN ($CLOSE(FAB=RD_FAB); RETURN .STATUS);
1182 1311 2 RD_RAB[RAB$V_EOF] = 0;           ! Clear EOF flag
1183 1312 2
1184 1313 2
1185 1314 2 : If there are any more files, open them and append their contents to the
1186 1315 2 : end of the last file.
1187 1316 2
1188 1317 2 IF .CUR_FID GTR 0
1189 1318 2 THEN BEGIN
1190 1319 2   $FAB_INIT( FAB=FAB,           ! Create a FAB
1191 1320 2             FAC=GET,           ! GET access only
1192 1321 2             FOP=<SQO,         ! Sequential only
1193 1322 2             NAM>,           ! Open by NAM block
1194 1323 2             NAM=NAM);         ! NAM block addr
1195 1324 2   $RAB_INIT( RAB=RAB,         ! Create a RAB
1196 1325 2             FAB=FAB,         ! FAB address
1197 1326 2             ROP=<RAH,         ! Read-ahead
1198 1327 2             LOC>,           ! Locate mode
1199 1328 2             UBF=BUFF,         ! Record buffer addr
1200 1329 2             USZ=MAIL$K_INBUFFSZ); ! Record buffer size
1201 1330 2   CH$MOVE(NAM$C_BLN,RD_NAM,NAM);
1202 1331 2   NAM[NAM$S_L_RSA] = NAMRSS;
1203 1332 2   NAM[NAM$S_L_ESA] = NAMESS;
1204 1333 2   CH$MOVE(.RD_NAM[NAM$B_RSL],.RD_NAM[NAM$S_L_RSA],NAMRSS);
1205 1334 2   CH$MOVE(.RD_NAM[NAM$B_ESL],.RD_NAM[NAM$S_L_ESA],NAMESS);
1206 1335 2
1207 1336 2 : Open the next file
1208 1337 2

```

```

: 1209      1338  4      DO      BEGIN
: 1210      1339  4      NAM[NAM$W_FID_NUM] = .FID[.CUR_FID,0]; ! Set file ID
: 1211      1340  4      NAM[NAM$W_FID_SEQ] = .FID[.CUR_FID,1];
: 1212      1341  4      NAM[NAM$W_FID_RVN] = .FID[.CUR_FID,2];
: 1213      1342  4      IF_ERR($OPEN(FAB = FAB);,$CLOSE(FAB = RD_FAB); RETURN .STATUS);
: 1214      1343  4      IF .FAB[FAB$B_ORG] NEQ FAB$C_SEQ
: 1215      1344  5          THEN BEGIN
: 1216      1345  5              LOCAL DEV_DESC : VECTOR[2,LONG],
: 1217      1346  5                  FILE_DESC : DYN_DESC_DECL;
: 1218      1347  5
: 1219      1348  5              $CLOSE(FAB=FAB);
: 1220      1349  5              $CLOSE(FAB=RD_FAB);
: 1221      1350  5              DEV_DESC[0] = .NAM[NAM$B_DEV];
: 1222      1351  5              DEV_DESC[1] = .NAM[NAM$L_DEV];
: 1223      1352  5              LIB$FID TO NAME(DEV_DESC,FID[.CUR_FID,0],FILE_DESC);
: 1224      1353  5              SIGNAL(MAIL$_FILECONFUSED,1,
: 1225      1354  6                  (IF .FILE_DESC[DSC$W_LENGTH] NEQ 0
: 1226      1355  6                      THEN FILE_DESC
: 1227      1356  5                          ELSE UTIL$GETFILENAME(FAB)));
: 1228      1357  5              STR$FREE1 DX(FILE_DESC);
: 1229      1358  5              RETURN MAIL$_FILECONFUSED;
: 1230      1359  4          END;
: 1231      1360  4      $CONNECT(RAB = RAB);
: 1232      1361  4      RD_FAB[FAB$L_ALQ] = .FAB[FAB$L_ALQ] + ! Add file's size
: 1233      1362  4                  .RD_FAB[FAB$L_ALQ];
: 1234      1363  4      !
: 1235      1364  4      ! Append all the records to the end of the file
: 1236      1365  4      !
: 1237      1366  5      WHILE STATUS = $GET(RAB = RAB)
: 1238      1367  5      DO      BEGIN
: 1239      1368  5          RD_RAB[RAB$W_RSZ] = .RAB[RAB$W_RSZ];
: 1240      1369  5          RD_RAB[RAB$L_RBF] = .RAB[RAB$L_RBF];
: 1241      1370  5          STATUS = $PUT(RAB = RD_RAB);
: 1242      1371  5          IF NOT .STATUS THEN EXITLOOP;
: 1243      1372  4          END;
: 1244      1373  4
: 1245      1374  4      IF .STATUS NEQ RM$S_EOF ! Check for failure
: 1246      1375  4      THEN ($CLOSE(FAB = FAB); $CLOSE(FAB = RD_FAB); RETURN .STATUS);
: 1247      1376  4      $FLUSH(RAB = RD_RAB); ! Force disk update
: 1248      1377  4      FAB[FAB$V_DLT] = 1; ! Delete other file
: 1249      1378  4      MAIL$UP_PRIV(1*$BITPOSITION(PRV$V_SYSPRV),0); ! Enable system priv
: 1250      1379  4      $CLOSE(FAB = FAB); ! Close and delete
: 1251      1380  4      MAIL$DOWN_PRIV(); ! Disable system uic
: 1252      1381  4      FAB[FAB$V_DLT] = 0;
: 1253      1382  4      END
: 1254      1383  3      WHILE (CUR_FID = .CUR_FID - 1) GTR 0; ! Open all files
: 1255      1384  2      END;
: 1256      1385  2      $REWIND(RAB = RD_RAB); ! Rewind message file
: 1257      1386  2      RETURN TRUE
: 1258      1387  1      END;

```

```

4C 49 41 4D 00A8A P.AAH: .ASCII \MAIL\
49 41 4D 2E 00A8E .ASCII \.MAIL\
2A 3B 00A92 .ASCII \;*

```


				FE72	CD40	9F	000D1		PUSHAB	FID-6[RO]		
	C4	A8				9E	B0	000D6	MOVW	@(SP)+, RD_NAM+36		
				FE74	CD40	9F	000DA		PUSHAB	FID-4[RO]	1296	
	C6	A8				9E	B0	000DF	MOVW	@(SP)+, RD_NAM+38		
				FE76	CD40	9F	000E3		PUSHAB	FID-2[RO]	1297	
	C8	A8				9E	B0	000E8	MOVW	@(SP)+, RD_NAM+40		
	07	A8				01	88	000EC	BISB2	#1, RD_FAB+7	1298	
						58	DD	000F0	PUSHL	R8	1299	
		6B				01	FB	000F2	CALLS	#1, SYSSOPEN		
		57				50	D0	000F5	MOVL	RO, STATUS		
	07	A8				01	8A	000F8	BICB2	#1, RD_FAB+7	1300	
		CB				57	E9	000FC	BLBC	STATUS, 6\$	1301	
				1D		A8	95	000FF	TSTB	RD_FAB+29	1302	
						18	13	00102	BEQL	8\$		
						58	DD	00104	PUSHL	R8	1303	
		69				01	FB	00106	CALLS	#1, SYSSCLOSE		
				98		A8	9F	00109	PUSHAB	MAIL\$Q_RD_DESC	1304	
						01	DD	0010C	PUSHL	#1		
				007E8122		8F	DD	0010E	PUSHL	#8290594		
	00000000G	00				03	FB	00114	CALLS	#3, LIB\$SIGNAL		
						04	00	0011B	RET			
	00000000G	00				08	88	0011C	8\$: BISB2	#8, MAIL\$GL_FLAGS+1	1305	
						56	D7	00123	DECL	CUR_FID	1306	
	55	A8				01	88	00125	BISB2	#1, RD_RAB+5	1307	
				50		A8	9F	00129	PUSHAB	RD_RAB	1308	
	00000000G	00				01	FB	0012C	CALLS	#1, SYSSCONNECT		
						50	D0	00133	MOVL	RO, STATUS		
						57	E8	00136	BLBS	STATUS, 9\$	1309	
				0188		31	00	00139	BRW	18\$		
	55	A8				01	8A	0013C	9\$: BICB2	#1, RD_RAB+5	1311	
						56	D5	00140	TSTL	CUR_FID	1317	
						03	14	00142	BGTR	10\$		
				01C1		31	00	00144	BRW	22\$		
	0050	8F	00			00	2C	00147	10\$: MOVCS	#0, (SP), #0, #80, \$RMS_PTR	1323	
				FE28	CD			0014E				
				5003		8F	B0	00151	MOVW	#20483, \$RMS_PTR		
				01000040		8F	D0	00158	MOVL	#16777280, \$RMS_PTR+4		
						02	90	00161	MOVW	#2, \$RMS_PTR+22		
						02	90	00166	MOVW	#2, \$RMS_PTR+31		
				FD84	CD	9E	00	00168	MOVAB	NAM, \$RMS_PTR+40		
	0044	8F	00			00	2C	00172	MOVCS	#0, (SP), #0, #68, \$RMS_PTR	1329	
								00179				
				FDE4	CD	8F	B0	0017C	MOVW	#17409, \$RMS_PTR		
				4401		8F	D0	00183	MOVL	#66048, \$RMS_PTR+4		
				00010200		8F	B0	0018C	MOVW	#512, \$RMS_PTR+32		
				0200		AE	9E	00193	MOVAB	BUFF, \$RMS_PTR+36		
				10		9E	00	00199	MOVAB	FAB, \$RMS_PTR+60		
				FE28	CD	9E	00	001A0	MOVCS	#96, RD_NAM, NAM	1330	
	FD84	CD		0060		8F	28	001A0	MOVCS	NAMRSS, NAM+4	1331	
				0310		CE	9E	001A9	MOVAB	NAMRSS, NAM+4	1332	
				0210		CE	9E	001B0	MOVAB	NAMRSS, NAM+12	1333	
				A3		A8	9A	001B7	MOVZBL	RD_NAM+3, R1		
				A4		A8	D0	001BB	MOVL	RD_NAM+4, R0		
				51		28	00	001BF	MOVCS	R1, (R0), NAMRSS		
	0310	CE		60		A8	9A	001C5	MOVZBL	RD_NAM+11, R1	1334	
				51		A8	D0	001C9	MOVL	RD_NAM+12, R0		
				50		51	28	001CD	MOVCS	R1, (R0), NAMRSS		
	0210	CE		60		06	C5	001D3	11\$: MULL3	#6, CUR_FID, R2	1339	
		52		56								

		FE72	CD42	9F	001D7		PUSHAB	FID-6[R2]		
FDA8	CD		9E	B0	001DC		MOVW	@(SP)+, NAM+36		
		FE74	CD42	9F	001E1		PUSHAB	FID-4[R2]		1340
FDA4	CD		9E	B0	001E6		MOVW	@(SP)+, NAM+38		
		FE76	CD42	9F	001EB		PUSHAB	FID-2[R2]		1341
FDAC	CD		9E	B0	001F0		MOVW	@(SP)+, NAM+40		
		FE28	CD	9F	001F5		PUSHAB	FAB		1342
	6B		01	FB	001F9		CALLS	#1, SYSSOPEN		
	53		50	DO	001FC		MOVL	RO, STATUS		
	09		53	E8	001FF		BLBS	STATUS, 12\$		
			58	DD	00202		PUSHL	R8		
	69		01	FB	00204		CALLS	#1, SYSSCLOSE		
	50		53	DO	00207		MOVL	STATUS, RO		
			04	0020A			RET			
		FE45	CD	95	0020B	12\$:	TSTB	FAB+29		1343
			65	13	0020F		BEQL	15\$		
	6C	020E0000	8F	DO	00211		MOVL	#34471936, FILE_DESC		1346
			04	AE	D4	00218	CLRL	FILE_DESC+4		
		FE28	CD	9F	0021B		PUSHAB	FAB		1348
	69		01	FB	0021F		CALLS	#1, SYSSCLOSE		
			58	DD	00222		PUSHL	R8		1349
	69		01	FB	00224		CALLS	#1, SYSSCLOSE		
	08	AE	FDBD	CD	9A	00227	MOVZBL	NAM+57, DEV_DESC		1350
	OC	AE	FDC8	CD	DO	0022D	MOVL	NAM+68, DEV_DESC+4		1351
			5E	DD	00233		PUSHL	SP		1352
		FE72	CD42	9F	00235		PUSHAB	FID-6[R2]		
			10	AE	9F	0023A	PUSHAB	DEV_DESC		
00000000G	00		03	FB	0023D		CALLS	#3, LIB\$FID_TO_NAME		
			6E	B5	00244		TSTW	FILE_DESC		1354
			05	13	00246		BEQL	13\$		
	50		6E	9E	00248		MOVAB	FILE_DESC, RO		
			07	11	0024B		BRB	14\$		
		FE28	CD	9F	0024D	13\$:	PUSHAB	FAB		1356
	6A		01	FB	00251		CALLS	#1, UTIL\$GETFILENAME		
			50	DD	00254	14\$:	PUSHL	RO		
			01	DD	00256		PUSHL	#1		1353
		007E8122	8F	DD	00258		PUSHL	#8290594		
00000000G	00		03	FB	0025E		CALLS	#3, LIB\$SIGNAL		1357
			5E	DD	00265		PUSHL	SP		
00000000G	00		01	FB	00267		CALLS	#1, STR\$FREE1_DX		
	50	007E8122	8F	DO	0026E		MOVL	#8290594, RO		1358
			04	00275			RET			
		FDE4	CD	9F	00276	15\$:	PUSHAB	RAB		1360
	00		01	FB	0027A		CALLS	#1, SYSSCONNECT		
	10	A8	FE38	CD	CO	00281	ADDL2	FAB+16, RD_FAB+16		1362
			FDE4	CD	9F	00287	PUSHAB	RAB		1366
00000000G	00		01	FB	0028B		CALLS	#1, SYSSGET		
	57		50	DO	00292		MOVL	RO, STATUS		
	1C		57	E9	00295		BLBC	STATUS, 17\$		
	72	A8	FE06	CD	B0	00298	MOVW	RAB+34, RD_RAB+34		1368
	78	A8	FE0C	CD	DO	0029E	MOVL	RAB+40, RD_RAB+40		1369
			50	A8	9F	002A4	PUSHAB	RD_RAB		1370
00000000G	00		01	FB	002A7		CALLS	#1, SYSSPUT		
	57		50	DO	002AE		MOVL	RO, STATUS		
	D3		57	E8	002B1		BLBS	STATUS, 16\$		1371
0001827A	8F		57	D1	002B4		CMPL	STATUS, #98938		1374
			10	13	002BB		BEQL	20\$		

		FE28	CD	9F	002BD		PUSHAB	FAB	:	1375
	69		01	FB	002C1		CALLS	#1, SYSS\$CLOSE	:	
			58	DD	002C4	18\$:	PUSHL	R8	:	
	69		01	FB	002C6		CALLS	#1, SYSS\$CLOSE	:	
	50		57	DD	002C9	19\$:	MOVL	STATUS, R0	:	
				04	002CC		RET		:	
		50	A8	9F	002CD	20\$:	PUSHAB	RD_RAB	:	1376
00000000G	00		01	FB	002D0		CALLS	#1, SYSS\$FLUSH	:	
FE2D	CD	80	8F	88	002D7		BISB2	#128, FAB+5	:	1377
			7E	D4	002DD		CLRL	-(SP)	:	1378
		10000000	8F	DD	002DF		PUSHL	#268435456	:	
00000000G	00		02	FB	002E5		CALLS	#2, MAIL\$UP_PRIV	:	
		FE28	CD	9F	002EC		PUSHAB	FAB	:	1379
	69		01	FB	002F0		CALLS	#1, SYSS\$CLOSE	:	
00000000G	00		00	FB	002F3		CALLS	#0, MAIL\$DOWN_PRIV	:	1380
FE2D	CD	80	8F	8A	002FA		BICB2	#128, FAB+5	:	1381
	02		56	F5	00300		SOBGTR	CUR_FID, 21\$:	1383
			03	11	00303		BRB	22\$:	
			FECB	31	00305	21\$:	BRW	11\$:	
		50	A8	9F	00308	22\$:	PUSHAB	RD_RAB	:	1385
00000000G	00		01	FB	0030B		CALLS	#1, SYSS\$REWIND	:	
	50		01	DD	00312	23\$:	MOVL	#1, R0	:	1386
			04	00315			RET		:	1387

: Routine Size: 790 bytes, Routine Base: \$CODE\$ + 0A94

```

: 1260      1388 1 ROUTINE DELFILE (FAB) =
: 1261      1389 2 BEGIN
: 1262      1390 2 |++
: 1263      1391 2 | FUNCTIONAL DESCRIPTION:
: 1264      1392 2 |
: 1265      1393 2 |           This action routine is called by LIB$FILE_SCAN for each file which
: 1266      1394 2 |           satisfies the file specification match. The file is deleted
: 1267      1395 2 |
: 1268      1396 2 |--
: 1269      1397 2 MAP
: 1270      1398 2     FAB : REF $BLOCK;
: 1271      1399 2
: 1272      1400 2 LOCAL
: 1273      1401 2     STATUS;
: 1274      1402 2
: 1275      1403 2 STATUS = $ERASE(FAB=.FAB);
: 1276      1404 2
: 1277      1405 2 RETURN .STATUS
: 1278      1406 1 END;

```

```

                                .EXTRN  SYSSERASE
                                0000 0000 DELFILE: .WORD  Save nothing
                                04   AC  DD 00002  PUSHL   FAB
                                01   FB 00005  CALLS   #1, SYSSERASE
                                04   0000C  RET
                                : 1388
                                : 1403
                                : 1406

```

: Routine Size: 13 bytes, Routine Base: \$CODE\$ + ODAA

```

: 1280      1407 1 GLOBAL ROUTINE MAIL$DEL_TMPF : NOVALUE =
: 1281      1408 2 BEGIN
: 1282      1409 2 |**
: 1283      1410 2 | FUNCTIONAL DESCRIPTION:
: 1284      1411 2 |
: 1285      1412 2 |         This routine deletes all the MAIL.TMP files that may be in
: 1286      1413 2 |         the user's login directory. This is done to prevent inadvertently
: 1287      1414 2 |         sending an old message when a quit is done from the editor.
: 1288      1415 2 |
: 1289      1416 2 | --
: 1290      1417 2 |
: 1291      1418 2 LOCAL
: 1292      1419 2     FAB : $FAB_DECL,
: 1293      1420 2     NAM : $NAM_DECL,
: 1294      1421 2     RSBUF : $BBLOCK[NAM$C_MAXRSS],
: 1295      1422 2     ESBUF : $BBLOCK[NAM$C_MAXRSS];
: 1296      1423 2
: 1297      1424 2 $FAB_INIT(FAB=FAB,
: 1298      1425 2     NAM=NAM,
: 1299      1426 2     FNS=.MAIL$Q_TMPFILE_DESC[DSC$W_LENGTH],
: 1300      1427 2     FNA=.MAIL$Q_TMPFILE_DESC[DSC$A_POINTER],
: 1301      1428 2     DNS=%CHARCOUNT(';*'),
: 1302      1429 2     DNA=UPLIT(%ASCII ';*');
: 1303      1430 2
: 1304      1431 2 $NAM_INIT(NAM=NAM,
: 1305      1432 2     ESS=NAM$C_MAXRSS,
: 1306      1433 2     ESA=ESBUF,
: 1307      1434 2     RSS=NAM$C_MAXRSS,
: 1308      1435 2     RSA=RSBUF);
: 1309      1436 2
: 1310      1437 2 LIB$FILE_SCAN(FAB,DELFILE,0);
: 1311      1438 2 RETURN;
: 1312      1439 1 END;

```

```

00 00 2A 3B 00DB7 P.AAI: .BLKB 1
00DB8 .ASCII \;*\<0><0>

```

				007C 00000	.ENTRY MAIL\$DEL_TMPF, Save R2,R3,R4,R5,R6	: 1407
		56	F7	AF 9E 00002	MOVAB P.AAI, R6	
0050	8F	00	FD50	CE 9E 00006	MOVAB -688(SP), SP	
		6E		00 2C 0000B	MOVCS #0, (SP), #0, #80, \$RMS_PTR	: 1429
			B0	AD 00012		
		B0	AD 5003	8F B0 00014	MOVW #20483, \$RMS_PTR	
		C6	AD	02 90 0001A	MOVW #2, \$RMS_PTR+22	
		CF	AD	02 90 0001E	MOVW #2, \$RMS_PTR+31	
		D8	AD FF50	CD 9E 00022	MOVAB NAM, \$RMS_PTR+40	
		DC	AD 0000000G	00 D0 00028	MOVL MAIL\$Q_TMPFILE_DESC+4, \$RMS_PTR+44	
		E0	AD	66 9E 00030	MOVAB P.AAI, \$RMS_PTR+48	
		E4	AD 0000000G	00 90 00034	MOVW MAIL\$Q_TMPFILE_DESC, \$RMS_PTR+52	
		E5	AD	02 90 0003C	MOVW #2, \$RMS_PTR+53	
0060	8F	00	6E	00 2C 00040	MOVCS #0, (SP), #0, #96, \$RMS_PTR	: 1435
			FF50	CD 00047		
		FF50	CD 6002	8F B0 0004A	MOVW #24578, \$RMS_PTR	

MAIL\$MSGSUBS
V04-000

MAIL MESSAGE SUBROUTINES

L 11
16-Sep-1984 01:08:07
14-Sep-1984 12:42:28

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MSGSUBS.B32;1 (19)

MA
VO

FF52	CD		01	8E	00051	MNEGB	#1, \$RMS_PTR+2	:	
FF54	CD	0100	CE	9E	00056	MOVAB	RSBUF, \$RMS_PTR+4	:	
FF5A	CD		01	8E	0005D	MNEGB	#1, \$RMS_PTR+10	:	
FF5C	CD		6E	9E	00062	MOVAB	ESBUF, \$RMS_PTR+12	:	
			7E	D4	00067	CLRL	-(SP)	:	1437
			82	AF	9F	PUSHAB	DELFILE	:	
			B0	AD	9F	PUSHAB	FAB	:	
00000000G	00		03	FB	0006F	CALLS	#3, LIB\$FILE_SCAN	:	
			04	00076	RET			:	1439

; Routine Size: 119 bytes, Routine Base: \$CODE\$ + ODBC

: 1314 1440 1 END
: 1315 1441 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	308	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$GLOBALS	36	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODES	3635	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	148	1	581	00:00.7
_\$255\$DUA28:[MAIL.OBJ]MAILDEF.L32;1	457	22	4	26	00:00.2

COMMAND QUALIFIERS

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

: Size: 3574 code + 405 data bytes
: Run Time: 00:41.3
: Elapsed Time: 02:43.2
: Lines/CPU Min: 2092
: Lexemes/CPU-Min: 40214
: Memory Used: 298 pages
: Compilation Complete

