

[illegible]

```
IIIIII  SSSSSSSS UU  UU  BBBB BBBB  SSSSSSSS
IIIIII  SSSSSSSS UU  UU  BBBB BBBB  SSSSSSSS
II  SS  UU  UU  BB  BB  SS  SSSSSSSS
II  SS  UU  UU  BB  BB  SS  SSSSSSSS
II  SS  UU  UU  BB  BB  SS  SSSSSSSS
II  SS  UU  UU  BB  BB  SS  SSSSSSSS
II  SS  UU  UU  BB  BB  SS  SSSSSSSS
II  SS  UU  UU  BB  BB  SS  SSSSSSSS
II  SS  UU  UU  BB  BB  SS  SSSSSSSS
IIIIII  SSSSSSSS UU  UU  BBBB BBBB  SSSSSSSS
IIIIII  SSSSSSSS UU  UU  BBBB BBBB  SSSSSSSS
UUUUUUUUUU BBBB BBBB  SSSSSSSS
UUUUUUUUUU BBBB BBBB  SSSSSSSS
```

....
....
....
....

```
LL  IIIIIII  SSSSSSSS
LL  IIIIIII  SSSSSSSS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SSSSSS
LL  II  SSSSSS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SS
LLLLLLLLLLL IIIIIII  SSSSSSSS
LLLLLLLLLLL IIIIIII  SSSSSSSS
```

```
1 0001 0 MODULE MAIL$ISAM_SUBS( %TITLE 'ISAM 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: Benn Schreiber, CREATION DATE: 22-Dec-1982
38 0038 1
39 0039 1 MODIFIED BY:
40 0040 1
41 0041 1 V03-022 ROP0031 Robert Posniak 02-AUG-1984
42 0042 1 Make sure do not copy deleted messages between
43 0043 1 folders during a COPY or MOVE command.
44 0044 1
45 0045 1 V03-021 ROP0027 Robert Posniak 23-JUL-1984
46 0046 1 Change read cluster size from 127 to 32.
47 0047 1
48 0048 1 V03-020 ROP0003 Robert Posniak 6-JUN-1984
49 0049 1 Check for oversized record. Fix COMPRESS
50 0050 1 command to strip version number when opening
51 0051 1 the newly created file. Also, set the protection
52 0052 1 on the .OLD file so that the owner can delete it.
53 0053 1
54 0054 1 V03-019 BLS0292 Benn Schreiber 8-APR-1984
55 0055 1 Pass accvios on in reclaim_handler
56 0056 1
57 0057 1 V03-018 BLS0263 Benn Schreiber 4-FEB-1984
```

: F

58	0058	1	Add COMPRESS command. Use LIB\$FIND_IMAGE_SYMBOL
59	0059	1	
60	0060	1	V03-017 BLS0255 Benn Schreiber 28-Dec-1983
61	0061	1	Global mail flags, sanitize flags to create_isam. Add
62	0062	1	support for exec mode logicals in create_isam. Don't quit
63	0063	1	on errors when emptying wastebasket. If filing new message,
64	0064	1	clear new message flag and decrement count (filed before
65	0065	1	it was read).
66	0066	1	
67	0067	1	V03-016 BLS0250 Benn Schreiber 8-Dec-1983
68	0068	1	Ensure users mail file is a disk file.
69	0069	1	
70	0070	1	V03-015 BLS0246 Benn Schreiber 7-Nov-1983
71	0071	1	Put new mail count in VMSMAIL.DAT. Confirm on non-existent
72	0072	1	folder creation. Remove check for uic ownership of external
73	0073	1	mail files.
74	0074	1	
75	0075	1	V03-014 BLS0238 Benn Schreiber 1-Sep-1983
76	0076	1	Check correct null parameter in mail\$check_date, also
77	0077	1	reinit descriptor before each fao
78	0078	1	
79	0079	1	V03-013 BLS0237 Benn Schreiber 31-Aug-1983
80	0080	1	Include end of file mark word in byte count.
81	0081	1	
82	0082	1	V03-012 BLS0235 Benn Schreiber 23-Aug-1983
83	0083	1	Have to read non-var files with \$get's, since there is
84	0084	1	no byte count. Do not set folder name if no messages found
85	0085	1	when selecting messages. Filter trailing cr/lf's in
86	0086	1	mail\$get_i_msg.
87	0087	1	
88	0088	1	V03-011 BLS0229 Benn Schreiber 16-Jul-1983
89	0089	1	New routines to handle new message count in mail file.
90	0090	1	Correct way messages are deleted on MOVE command. Change
91	0091	1	protection on external files and then \$erase. This then
92	0092	1	requires no SYSPRV, and works over the net.
93	0093	1	
94	0094	1	V03-010 BLS0227 Benn Schreiber 1-Jul-1983
95	0095	1	Remove system delete access from isam mail files.
96	0096	1	
97	0097	1	V03-009 BLS0224 Benn Schreiber 7-Jun-1983
98	0098	1	Signal default folder not found in select_msgs, and
99	0099	1	if /new, then go after default folder name, not MAIL.
100	0100	1	Sort messages in add_message. Enable DFW according to
101	0101	1	MAIL\$B_DFW (off for now). Get rid of MAIL\$GET_FILEDESC.
102	0102	1	
103	0103	1	V03-008 BLS0222 Benn Schreiber 7-May-1983
104	0104	1	Modify reclaim operation to be automatic with a message
105	0105	1	
106	0106	1	V03-007 BLS0221 Benn Schreiber 5-May-1983
107	0107	1	Remove temp message if \$erase fails
108	0108	1	
109	0109	1	V03-006 BLS0218 Benn Schreiber 18-Apr-1983
110	0110	1	Enhance check_date to also ensure file doesn't exist
111	0111	1	if optional 3rd dev/dir descr. is passed.
112	0112	1	
113	0113	1	V03-005 BLS0217 Benn Schreiber 12-Apr-1983
114	0114	1	Count bytes of deleted space, rather than number of messages.

```

: 115      0115 1 |
: 116      0116 1 |
: 117      0117 1 |
: 118      0118 1 |
: 119      0119 1 |
: 120      0120 1 |
: 121      0121 1 |
: 122      0122 1 |
: 123      0123 1 |
: 124      0124 1 |
: 125      0125 1 |
: 126      0126 1 |
: 127      0127 1 |
: 128      0128 1 |
: 129      0129 1 |
: 130      0130 1 |
: 131      0131 1 |
: 132      0132 1 |
: 133      0133 1 |

```

Correct signalling of CONVERT/RECLAIM errors. Tell
when new folder created with COPY command. Add external
messages (stored in another file)

V03-004 BLS0216 Benn Schreiber 1-Apr-1983
Use old message date time when copying messages

V03-003 BLS0215 Benn Schreiber 31-Mar-1983
Return 'TO' string from MAIL\$INFO_I_MSG

V03-002 BLS0214 Benn Schreiber 27-Mar-1983
Remove LASTREAD code. Implement new scheme of NEWMAIL
handling. Handle locked info records. Add number of
records info in msg header. Correct quoted sender name
parsing in MAIL\$INFO_I_MSG. Do not allow delete in waste
basket folder. Tell user if new folder created. Do not
update deleted msg count if number to add is 0. Correct
handling of last message being deleted in MAIL\$FIND_I_MSG.

```
135 0134 1 |
136 0135 1 | TABLE OF CONTENTS:
137 0136 1 |
138 0137 1 |
139 0138 1 | INCLUDE FILES:
140 0139 1 |
141 0140 1 |     LIBRARY 'SYSS$LIBRARY:LIB';
142 0141 1 |     REQUIRE 'SRC$:MAILREQ';
143 0287 1 |     LIBRARY 'LIB$:MAILDEF';
144 0288 1 |
145 0289 1 | FORWARD ROUTINE
146 0290 1 |     MAIL$SELECT_MSGS;                                !Select messages
147 0291 1 |
148 0292 1 | GLOBAL
149 0293 1 |     MAIL$DELMSGCNT,                                !Number deleted msgs
150 0294 1 |     MAIL$RECL_THRESHOLD : LONG INITIAL(32768), !Threshold before reclaim done
151 0295 1 |     MAIL$B_MSGTRAN : BYTE;                          !Transition for get_i_msg
152 0296 1 |
153 0297 1 | BIND
154 0298 1 |     MSGTRAN = MAIL$B_MSGTRAN : BYTE;
155 0299 1 |
156 0300 1 | GLOBAL
157 0301 1 |     MAIL$Q_CURFOLDER : DYN_DESC_DECL,               !Descriptor of current folder name
158 0302 1 |     MAIL$Q_WASTEDESC : DYN_DESC_DECL,               !String descriptor for wastebasket
159 0303 1 |     MAIL$MSGMAX,                                     !Maximum number of messages in vector
160 0304 1 |     MAIL$NMSG,                                       !Number of messages currently in vector
161 0305 1 |     MAIL$CURMSG : REF $BBLOCK,                      !Pointer to current message block
162 0306 1 |     MAIL$MSGTBL : REF VECTOR[,LONG];               !Pointer to message vector
163 0307 1 |
164 0308 1 | EXTERNAL
165 0309 1 |     MAIL$GL_BIGBUFFER,                               ! and address
166 0310 1 |     MAIL$MYUIC,                                     !My UIC from SYSUAF
167 0311 1 |     MAIL$MYPID,                                     !My process pid
168 0312 1 |     MAIL$SIZLIM,                                    !Size of msg to externalize
169 0313 1 |     MAIL$Q_MSG_FNM : $BBLOCK,                       !Descriptor of filename of mail file
170 0314 1 |     MAIL$B_DFW : BYTE,                              !True if should use FOP=DFW
171 0315 1 |     MAIL$G_FLAGS : $BBLOCK,                         !Mail control flags
172 0316 1 |     MAIL$Q_MAILDEF : $BBLOCK,                       !'dev:[dir]mail.mai' spec
173 0317 1 |     MAIL$Q_USERDEF : $BBLOCK,                       !Default file spec
174 0318 1 |     MAIL$MSG_NUM,                                   !Current msg number
175 0319 1 |     MAIL$MSG_STS,                                   !msg status
176 0320 1 |     MAIL$RECBUF,                                    !Pointer to user record buffer
177 0321 1 |     MAIL$W_MSG_RSIZ : WORD,                         !sequential msg size
178 0322 1 |     MAIL$MSG_RADR,                                  !seq. msg addr
179 0323 1 |     MAIL$Q_USERNAME : $BBLOCK,                     !Descriptor for username
180 0324 1 |     MAIL$RD_RAB : $BBLOCK,                          !RAB for reading input file
181 0325 1 |     MAIL$Q_RD_DESC : $BBLOCK;                      !Descriptor of currently open mail file
182 0326 1 |
183 0327 1 | EXTERNAL ROUTINE
184 0328 1 |     UTIL$GETFILENAME,
185 0329 1 |     LIB$FIND_IMAGE_SYMBOL,
186 0330 1 |     UTIL$REPORT_IO_ERROR,
187 0331 1 |     CLIS$PRESENT,
188 0332 1 |     CLIS$GET_VALUE,
189 0333 1 |     LIB$CVT_TIME,                                  !Convert date/time
190 0334 1 |     LIB$GET_VM,                                    !Allocate virtual memory
191 0335 1 |     LIB$PUT_OUTPUT,                                !Write to SYSS$OUTPUT
```



```
192 0336 1 LIB$FREE VM,
193 0337 1 LIB$RENAME FILE,
194 0338 1 LIB$COPY R DX,
195 0339 1 MAIL$CLOSE FILE,
196 0340 1 MAIL$CONFIRM,
197 0341 1 MAIL$DISABLE CTRL,
198 0342 1 MAIL$ENABLE CTRL,
199 0343 1 MAIL$FIND MSG,
200 0344 1 MAIL$FAO_GETMSG,
201 0345 1 MAIL$GET_MSG,
202 0346 1 MAIL$GETMSG,
203 0347 1 MAIL$INCR_NEWMAIL,
204 0348 1 MAIL$DECR_NEWMAIL,
205 0349 1 MAIL$INFO_MSG,
206 0350 1 MAIL$OPEN_FILE,
207 0351 1 MAIL$RENAME_ERROR,
208 0352 1 MAIL$RENAME_SUCCESS,
209 0353 1 MAIL$REWIND_MSG,
210 0354 1 MAIL$TEST_SEQ_NEW,
211 0355 1 MAIL$TEST_SUBJ,
212 0356 1 MAIL$WRITE_MSG,
213 0357 1 STR$APPEND,
214 0358 1 STR$COPY DX,
215 0359 1 STR$CONCAT,
216 0360 1 STR$FREE1_DX,
217 0361 1 STR$TRIM,
218 0362 1 SYSS$FAO,
219 0363 1 UTIL$GETFILENAME;
220 0364 1
221 0365 1 OWN
222 0366 1 YN_DESC : DYN_DESC DECL,
223 0367 1 SND_XABPRO : $XABPRO DECL,
224 0368 1 DYN_DESC : DYN_DESC DECL,
225 0369 1 DATEBUF : $BLOCK[32],
226 0370 1 DATEDESC : VECTOR[2,LONG];
227 0371 1
228 0372 1 Define string descriptors
229 0373 1
230 P 0374 1 SD
231 P 0375 1 ('NEWMAIL',
232 P 0376 1 'CONVSHR',
233 0377 1 'MAIL');
234 0378 1
235 0379 1 BIND
236 0380 1 SD_NL = $DESCRIPTOR('NLAO:');
237 0381 1
238 0382 1 GLOBAL BIND
239 0383 1 MAIL$SD_NEWMAIL = SD_NEWMAIL,
240 0384 1 MAIL$SD_MAIL = SD_MAIL,
241 0385 1 MAIL$SD_EXTMSGNAM = $DESCRIPTOR('!ASMAIL$!XL!XL.MAI');
242 0386 1
243 0387 1 Declare facility-specific versions of shared messages
244 0388 1
245 P 0389 1 $SHR_MSGDEF(MAIL,126,LOCAL,
246 P 0390 1 (COPIEDR,SUCCESS),
247 P 0391 1 (CREATED,SUCCESS),
248 P 0392 1 (INVQUAVAL,WARNING),
```

!Deallocate it

```
!Close msg file
!Ask question, get y/n answer
!Disable CLI ctrl/y handling
!Enable CLI ctrl/y handling
!Find sequential message
!GETMSG followed by FAO
!Get sequential file msg record
!Get message text for message code
!Increment new mail count
!and decrement it
!Get sequential file msg info
!Open msg file
!Report error from lib$rename_file
!Report success
!Rewind sequential message
!Test if message is new message
!Test if subject present in seq. msg
!Write message to ISAM file
!String append
!String copy
!concatenate
!Deallocate dynamic string
!Trim trailing blanks
```

!Get file spec from FAB

```
!For yes/no answers
!Protection XAB
!Temp buf for date/time
!Descriptor for datebuf
```

```
: 249 P 0393 1 (FILNOTDEL,ERROR),
: 250 P 0394 1 (OPENOUT,ERROR),
: 251 P 0395 1 (OPENIN,WARNING),
: 252 P 0396 1 (READERR,WARNING),
: 253 P 0397 1 (WRITEERR,WARNING),
: 254 P 0398 1 (CLOSEDEL,ERROR),
: 255 0399 1 (CLOSEOUT,WARNING));
: 256 0400 1
: 257 0401 1 LITERAL
: 258 0402 1 PRO_SYS = $BITPOSITION(XAB$V_SYS;
: 259 0403 1 PRO_OWN = $BITPOSITION(XAB$V_OWN);
: 260 0404 1 PRO_GRP = $BITPOSITION(XAB$V_GRP);
: 261 0405 1 PRO_WLD = $BITPOSITION(XAB$V_WLD);
: 262 0406 1
: 263 0407 1 SND_PRO =
: 264 0408 1 ( XAB$M_NOEXE OR XAB$M_NODEL) ^ PRO_SYS OR
: 265 0409 1 ( XAB$M_NOEXE OR XAB$M_NODEL) ^ PRO_OWN OR
: 266 0410 1 (XAB$M_NOREAD OR XAB$M_NOWRITE OR XAB$M_NOEXE OR XAB$M_NODEL) ^ PRO_GRP OR
: 267 0411 1 (XAB$M_NOREAD OR XAB$M_NOWRITE OR XAB$M_NOEXE OR XAB$M_NODEL) ^ PRO_WLD;
: 268 0412 1 DELETE_PRO = ! to delete a file
: 269 0413 1 ( XAB$M_NOEXE ) ^ PRO_SYS OR
: 270 0414 1 ( XAB$M_NOEXE ) ^ PRO_OWN OR
: 271 0415 1 (XAB$M_NOREAD OR XAB$M_NOWRITE OR XAB$M_NOEXE OR XAB$M_NODEL) ^ PRO_GRP OR
: 272 0416 1 (XAB$M_NOREAD OR XAB$M_NOWRITE OR XAB$M_NOEXE OR XAB$M_NODEL) ^ PRO_WLD;
```



```
274 0417 1 ROUTINE GET_MEM(SIZE,RETADR) =
275 0418 1 ++
276 0419 1 FUNCTIONAL DESCRIPTION:
277 0420 1
278 0421 1 Allocate memory, signal error if unavailable and return status
279 0422 1
280 0423 1 --
281 0424 2 BEGIN
282 0425 2
283 0426 2 MAP
284 0427 2 RETADR : REF VECTOR[,LONG];
285 0428 2
286 0429 2 LOCAL
287 0430 2 STATUS;
288 0431 2
289 0432 3 IF NOT (STATUS = LIB$GET_VM(SIZE,RETADR[0]))
290 0433 2 THEN SIGNAL(.STATUS);
291 0434 2
292 0435 2 RETURN .STATUS
293 0436 1 END;
```

```
.TITLE MAIL$ISAM_SUBS ISAM MAIL MESSAGE SUBROUTINES
.IDENT \V04-000\

.PSECT $CODE$,NOWRT,2

4C 49 41 4D 57 45 4E 00000 P.AAB: .ASCII \NEWMAIL\
00007 00007 P.AAB: .BLKB 1
00000007 00008 P.AAA: .LONG 7
00000000' 0000C P.AAA: .ADDRESS P.AAB
52 48 53 56 4E 4F 43 00010 P.AAD: .ASCII \CONVSHR\
00017 00017 P.AAD: .BLKB 1
00000007 00018 P.AAC: .LONG 7
00000000' 0001C P.AAC: .ADDRESS P.AAD
4C 49 41 4D 00020 P.AAF: .ASCII \MAIL\
00000004' 00024 P.AAE: .LONG 4
00000000' 00028 P.AAE: .ADDRESS P.AAF
3A 30 41 4C 4E 0002C P.AAH: .ASCII \NLAO:\
00031 00031 P.AAH: .BLKB 3
00000005 00034 P.AAG: .LONG 5
00000000' 00038 P.AAG: .ADDRESS P.AAH
2E 4C 58 21 4C 58 21 24 4C 49 41 4D 53 41 21 0003C P.AAJ: .ASCII \!ASMAIL$!XL!XL.MAI\
49 41 4D 0004B 0004B P.AAJ: .BLKB 2
00000012 00050 P.AAI: .LONG 18
00000000' 00054 P.AAI: .ADDRESS P.AAJ

.PSECT $OWNS$,NOEXE,2

00# 00000 YN_DESC: .BYTE 0[2]
02 0E 00002 .BYTE 14, 2
00004 .BLKB 4
00008 SND_XABPRO:
00008 .BLKB 88
00# 00060 DYN_DESC:
00060 .BYTE 0[2]
```

```

02 0E 00062 .BYTE 14, 2
      00064 .BLKB 4
      00068 DATEBUF:.BLKB 32
      00088 DATEDESC:
      .BLKB 8
      .PSECT $GLOBAL$,NOEXE,2

00000 MAIL$ _DELMSGCNT::
      .BLKB 4
00008000 00004 MAIL$ _RECL_THRESHOLD::
      .LONG 32768
      00008 MAIL$ _MSGTRAN::
      .BLKB 1
      00009 .BLKB 3
      00# 0000C MAIL$ _CURFOLDER::
      .BYTE 0[2]
02 0E 0000E .BYTE 14, 2
      00010 .BLKB 4
      00# 00014 MAIL$ _WASTEDESC::
      .BYTE 0[2]
02 0E 00016 .BYTE 14, 2
      00018 .BLKB 4
      0001C MAIL$ _MSGMAX::
      .BLKB 4
      00020 MAIL$ _NMSG::
      .BLKB 4
      00024 MAIL$ _CURMSG::
      .BLKB 4
      00028 MAIL$ _MSGTBL::
      .BLKB 4

```

```

MSGTRAN= MAIL$ _MSGTRAN
SD_NEWMAIL= P.AAA
SD_CONVSHR= P.AAC
SD_MAIL= P.AAE
SD_NL= P.AAG
MAIL$SD_NEWMAIL== P.AAA
MAIL$SD_MAIL== P.AAF
MAIL$SD_EXTMSGNAM== P.AAI
      .EXTRN MAIL$ _BIGBUFFER
      .EXTRN MAIL$ _MYUIC, MAIL$ _MYPID
      .EXTRN MAIL$ _SIZLIM, MAIL$ _MSG_FNM
      .EXTRN MAIL$ _DFW, MAIL$ _GL_FLAGS
      .EXTRN MAIL$ _MAILDEF, MAIL$ _USERDEF
      .EXTRN MAIL$ _MSG_NUM, MAIL$ _MSG_STS
      .EXTRN MAIL$ _RECBUF, MAIL$ _MSG_RSIZ
      .EXTRN MAIL$ _MSG_RADR
      .EXTRN MAIL$ _USERNAME
      .EXTRN MAIL$ _RD_RAB, MAIL$ _RD_DESC
      .EXTRN UTIL$GETFILENAME
      .EXTRN LIB$FIND_IMAGE_SYMBOL
      .EXTRN UTIL$REPORT_IO_ERROR
      .EXTRN CLISP$PRESENT, CLISP$GET_VALUE
      .EXTRN LIB$CVT_TIME, LIB$GET_VM
      .EXTRN LIB$PUT_OUTPUT, LIB$FREE_VM
      .EXTRN LIB$RENAME_FILE

```

```
.EXTRN LIB$SCOPY R DX, MAIL$CLOSE_FILE
.EXTRN MAIL$CONFIRM, MAIL$DISABLE_CTRL
.EXTRN MAIL$ENABLE_CTRL
.EXTRN MAIL$FIND MSG, MAIL$FAO GETMSG
.EXTRN MAIL$GET MSG, MAIL$GETMSG
.EXTRN MAIL$INCR_NEWMAIL
.EXTRN MAIL$DECR_NEWMAIL
.EXTRN MAIL$INFO MSG, MAIL$OPEN_FILE
.EXTRN MAIL$RENAME_ERROR
.EXTRN MAIL$RENAME_SUCCESS
.EXTRN MAIL$REWIND MSG
.EXTRN MAIL$TEST_SEQ NEW
.EXTRN MAIL$TEST_SUBJ, MAIL$WRITE_MSG
.EXTRN STR$APPEND, STR$COPY DX
.EXTRN STR$CONCAT, STR$FREE DX
.EXTRN STR$TRIM, SYSS$FAO
```

.PSECT \$CODE\$,NOWRT,2

		0004 00000	GET_MEM:	.WORD	Save R2	: 0417
	08	AC DD 00002		PUSHL	RETADR	: 0432
	04	AC 9F 00005		PUSHAB	SIZE	:
00000000G	00	02 FB 00008		CALLS	#2, LIB\$GET_VM	:
	52	50 D0 0000F		MOVL	R0, STATUS	:
	09	52 E8 00012		BLBS	STATUS, 1\$	:
		52 DD 00015		PUSHL	STATUS	: 0433
00000000G	00	01 FB 00017		CALLS	#1, LIB\$SIGNAL	:
	50	52 D0 0001E	1\$:	MOVL	STATUS, R0	: 0435
		04 00021		RET		: 0436

; Routine Size: 34 bytes, Routine Base: \$CODE\$ + 0058

```
295 0437 1 ROUTINE ADD_MESSAGE(MSGADR) =
296 0438 1 **
297 0439 1 FUNCTIONAL DESCRIPTION:
298 0440 1
299 0441 1 Insert the message block into the message table. If this message
300 0442 1 causes the table to overflow, double the size of the table.
301 0443 1
302 0444 1 INPUTS:
303 0445 1 MSGADR is address of message descriptor block
304 0446 1
305 0447 1 --
306 0448 2 BEGIN
307 0449 2
308 0450 2 MAP
309 0451 2 MSGADR : REF $BBLOCK;
310 0452 2
311 0453 2 LOCAL
312 0454 2 NMSG,
313 0455 2 DATE1 : REF VECTOR[,LONG],
314 0456 2 DATE2 : REF VECTOR[,LONG],
315 0457 2 PTR : REF $BBLOCK;
316 0458 2
317 0459 2 IF .MAIL$NMSG EQL .MAIL$MSGMAX
318 0460 3 THEN BEGIN
319 0461 3 IF .MAIL$MSGMAX NEQ 0 !If this is not first time...
320 0462 4 THEN BEGIN
321 0463 4
322 0464 4 Allocate new table 2* size of old one
323 0465 4
324 P 0466 4 IF_ERR(GET_MEM((8*.MAIL$MSGMAX),PTR));,
325 0467 4 RETURN .STATUS);
326 0468 4 CH$MOVE(.MAIL$MSGMAX*4,.MAIL$MSGTBL,PTR);
327 0469 4 LIB$FREE_VM(%REF(4*.MAIL$MSGMAX),MAIL$MSGTBL);
328 0470 4 MAIL$MSGTBL = .PTR;
329 0471 4 MAIL$MSGMAX = 2*.MAIL$MSGMAX;
330 0472 4 END
331 0473 4 ELSE BEGIN
332 P 0474 4 IF_ERR(GET_MEM(128*4,MAIL$MSGTBL));,
333 0475 4 RETURN .STATUS);
334 0476 4 MAIL$MSGMAX = 128;
335 0477 3 END;
336 0478 2 END;
337 0479 2
338 0480 2 !
339 0481 2 Assume message will go at the end of the vector list
340 0482 2
341 0483 2 DATE1 = MSGADR[MMB T HDREC];
342 0484 2 MAIL$MSGTBL[.MAIL$NMSG] = .MSGADR; !Preset in new slot in table
343 0485 2 MAIL$NMSG = .MAIL$NMSG + 1;
344 0486 2 IF .MAIL$NMSG GTRU T
345 0487 3 THEN BEGIN
346 0488 3 PTR = .MAIL$MSGTBL[.MAIL$NMSG-2];
347 0489 3 DATE2 = PTR[MMB T HDREC];
348 0490 4 IF (.DATE1[1] GTRU .DATE2[1])
349 0491 5 OR ((.DATE1[1] EQL .DATE2[1])
350 0492 4 AND (.DATE1[0] GTRU .DATE2[0]))
351 0493 3 THEN RETURN TRUE;
```

; R

```
352 0494 2 END;
353 0495 2
354 0496 2 It didn't. Run through the vector and find the message it goes
355 0497 2 in front of.
356 0498 2
357 0499 2 INCR I FROM 0 TO .MAIL$ _NMSG-2
358 0500 3 DO BEGIN
359 0501 3 PTR = .MAIL$ _MSGTBL[.I];
360 0502 3 DATE2 = PTR[MMB T HDREC];
361 0503 4 IF (.DATE1[1] LSSU .DATE2[1])
362 0504 5 OR ((.DATE1[1] EQL .DATE2[1])
363 0505 4 AND (.DATE1[0] LSSU .DATE2[0]))
364 0506 4 THEN BEGIN
365 0507 4 LOCAL J;
366 0508 4
367 0509 4 J = .MAIL$ _NMSG - .I;
368 0510 4 CH$MOVE(.J*4,MAIL$ _MSGTBL[.I],MAIL$ _MSGTBL[.I+1]);
369 0511 4 MAIL$ _MSGTBL[.I] = .MSGADR;
370 0512 4 EXITLOOP;
371 0513 3 END;
372 0514 2 END;
373 0515 2
374 0516 2 RETURN TRUE
375 0517 1 END;
```

OFFC 00000 ADD_MESSAGE:						
	5E		08 C2 00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	0437
	50 00000000'		00 D0 00005	SUBL2	#8, SP	
	50 00000000'		00 D1 0000C	MOVL	MAIL\$ _MSGMAX, R0	0459
			67 12 00013	CMPL	MAIL\$ _NMSG, R0	
			50 D5 00015	BNEQ	4\$	
			47 13 00017	TSTL	R0	0461
			AE 9F 00019	BEQL	1\$	
7E	50	04	03 78 0001C	PUSHAB	PTR	0467
	BA AF		02 FB 00020	ASHL	#3, R0, -(SP)	
	49		50 E9 00024	CALLS	#2, GET_MEM	
56 00000000'	00		02 78 00027	BLBC	STATUS, -2\$	
	50 00000000'		00 D0 0002F	ASHL	#2, MAIL\$ _MSGMAX, R6	0468
04 BE	60		56 28 00036	MOVL	MAIL\$ _MSGTBL, R0	
			00 9F 0003B	MOVC3	R6, (R0), @PTR	
	04 AE		56 D0 00041	PUSHAB	MAIL\$ _MSGTBL	0469
			AE 9F 00045	MOVL	R6, 4(SP)	
		04	02 FB 00048	PUSHAB	4(SP)	
00000000G	00		02 FB 00048	CALLS	#2, LIB\$FREE VM	
00000000'	00	04	AE D0 0004F	MOVL	PTR, MAIL\$ _MSGTBL	0470
00000000'	00		02 C4 00057	MULL2	#2, MAIL\$ _MSGMAX	0471
			1C 11 0005E	BRB	4\$	0461
			00 9F 00060	PUSHAB	MAIL\$ _MSGTBL	0475
	7E	0200	8F 3C 00066	MOVZWL	#512, -(SP)	
FF6E	CF		02 FB 0006B	CALLS	#2, GET_MEM	
	01		50 E8 00070	BLBS	STATUS, -3\$	
			04 00073	RET		
00000000'	00	80	8F 9A 00074	MOVZBL	#128, MAIL\$ _MSGMAX	0476

58	04	AC	00000047	8F	C1	0007C	4\$:	ADDL3	#71, MSGADR, DATE1	:	0483	
		5A	00000000	00	D0	00085		MOVL	MAIL\$\$_MSGTBL, R10	:	0484	
		50	00000000	00	D0	0008C		MOVL	MAIL\$\$_NMSG, R0	:		
	6A40		04	AC	D0	00093		MOVL	MSGADR, (R10)[R0]	:		
			00000000	00	D6	00098		INCL	MAIL\$\$_NMSG	:	0485	
		59	00000000	00	D0	0009E		MOVL	MAIL\$\$_NMSG, R9	:	0486	
		01		59	D1	000A5		CMPL	R9, #1	:		
				1D	1B	000A8		BLEQU	5\$	:		
	04	AE		FB	AA49	D0	000AA	MOVL	-8(R10)[R9], PTR	:	0488	
57	04	AE	00000047	8F	C1	000B0		ADDL3	#71, PTR, DATE2	:	0489	
	04	A7		04	A8	D1	000B9	CMPL	4(DATE1), 4(DATE2)	:	0490	
				49	1A	000BE		BGTRU	9\$	:		
				05	12	000C0		BNEQ	5\$	:	0491	
	67			68	D1	000C2		CMPL	(DATE1), (DATE2)	:	0492	
				42	1A	000C5		BGTRU	9\$	:		
		5B		FE	A9	9E	000C7	5\$:	MOVAB	-2(R9), R11	:	0499
		56			01	CE	000CB	MNEGL	#1, I	:	0503	
				35	11	000CE		BRB	8\$	:		
	04	AE		6A46	D0	000D0	6\$:	MOVL	(R10)[I], PTR	:	0501	
57	04	AE	00000047	8F	C1	000D5		ADDL3	#71, PTR, DATE2	:	0502	
	04	A7		04	A8	D1	000DE	CMPL	4(DATE1), 4(DATE2)	:	0503	
				07	1F	000E3		BLSSU	7\$	:		
				1E	12	000E5		BNEQ	8\$	:	0504	
	67			68	D1	000E7		CMPL	(DATE1), (DATE2)	:	0505	
				19	1E	000EA		BGEQU	8\$	:		
50		59		56	C3	000EC	7\$:	SUBL3	I, R9, J	:	0509	
		50		04	C4	000F0		MULL2	#4, R0	:	0510	
			04	AA46	DF	000F3		PUSHAL	4(R10)[I]	:		
			6A46	DF	000F7			PUSHAL	(R10)[I]	:		
9E		9E		50	28	000FA		MOVCL	R0, @ (SP)+, @ (SP)+	:		
	6A46		04	AC	D0	000FE		MOVL	MSGADR, (R10)[I]	:	0511	
				04	11	00103		BRB	9\$	:	0506	
C7		56		5B	F3	00105	8\$:	AOBLEQ	R11, I, 6\$	:	0499	
		50		01	D0	00109	9\$:	MOVL	#1, R0	:	0516	
				04	0010C			RET		:	0517	

; Routine Size: 269 bytes, Routine Base: \$CODE\$ + 007A


```
377 0518 1 GLOBAL ROUTINE MAIL$FORM_EXTFILE (DATIM,DESC) : NUVALUE =
378 0519 1 ++
379 0520 1 FUNCTIONAL DESCRIPTION:
380 0521 1
381 0522 1 Make the external file name
382 0523 1
383 0524 1 --
384 0525 2 BEGIN
385 0526 2 MAP
386 0527 2 DESC : REF $BBLOCK;
387 0528 2
388 0529 2 LOCAL
389 0530 2 FDESC : VECTOR[2, LONG],
390 0531 2 TMPFAB : $FAB_DECL,
391 0532 2 TMPNAM : $NAM_DECL,
392 0533 2 NAMBUF : $BBLOCK[NAM$C_MAXRSS];
393 0534 2
394 P 0535 2 $FAB_INIT(FAB=TMPFAB,
395 P 0536 2 FNS=.MAIL$Q_RD_DESC[DSC$W_LENGTH],
396 P 0537 2 FNA=.MAIL$Q_RD_DESC[DSC$A_POINTER],
397 0538 2 NAM=TMPNAM);
398 0539 2
399 P 0540 2 $NAM_INIT(NAM=TMPNAM,
400 P 0541 2 ESS=NAM$C_MAXRSS,
401 P 0542 2 ESA=NAMBUF,
402 P 0543 2 RSS=NAM$C_MAXRSS,
403 0544 2 RSA=NAMBUF);
404 0545 2
405 0546 2 $PARSE(FAB=TMPFAB);
406 0547 2 FDESC[0] = .TMPNAM[NAM$B_NODE] + .TMPNAM[NAM$B_DEV] + .TMPNAM[NAM$B_DIR];
407 0548 3 FDESC[1] = (IF .TMPNAM[NAM$S_NODE] NEQ 0
408 0549 3 THEN .TMPNAM[NAM$S_NODE]
409 0550 3 ELSE .TMPNAM[NAM$S_DEV]);
410 0551 2
411 0552 2 SYSS$FAO(MAIL$SD_EXTMSGNAM,.DESC,.DESC,FDESC,
412 0553 2 .(.DATIM)<32,32,0>;
413 0554 2 .(.DATIM)<0,32,0>);
414 0555 2 RETURN;
415 0556 1 END;
```

.EXTRN SYSS\$PARSE

```
.ENTRY MAIL$FORM_EXTFILE, Save R2,R3,R4,R5
MOVAB -440(SP), SP
MOVCS #0, (SP), #0, #80, $RMS_PTR

MOVW #20483, $RMS_PTR
MOVB #2, $RMS_PTR+22
MOVB #2, $RMS_PTR+31
MOVAB TMPNAM, $RMS_PTR+40
MOVL MAIL$Q_RD_DESC+4, $RMS_PTR+44
MOVB MAIL$Q_RD_DESC, $RMS_PTR+52
MOVCS #0, (SP), #0, #96, $RMS_PTR

MOVW #24578, $RMS_PTR
```

```
0050 8F 00 5E FE48 CE 9E 00002
6E AD 00 2C 00007
A8 AD 0000E
BE AD 5003 8F B0 00010
C7 AD 02 90 00016
D0 AD FF48 CD 9E 0001E
D4 AD 00000000G 00 D0 00024
DC AD 00000000G 00 90 0002C
0060 8F 00 6E 00 2C 00034
FF48 CD FF48 CD 0003B
6002 8F B0 0003E
```

```
: 0518
: 0538
:
:
:
: 0544
:
```

FF4A	CD	01	8E	00045	MNEGB	#1, \$RMS_PTR+2	:	
FF4C	CD	6E	9E	0004A	MOVAB	NAMBUF, \$RMS_PTR+4	:	
FF52	CD	01	8E	0004F	MNEGB	#1, \$RMS_PTR+10	:	
FF54	CD	6E	9E	00054	MOVAB	NAMBUF, \$RMS_PTR+12	:	
		A8	AD	9F	PUSHAB	TMPFAB	:	0546
00000000G	00	01	FB	0005C	CALLS	#1, SYSSPARSE	:	
	50	80	AD	9A	MOVZBL	TMPNAM+56, R0	:	0547
	51	81	AD	9A	MOVZBL	TMPNAM+57, R1	:	
	50	51	C0	0006B	ADDL2	R1, R0	:	
	52	82	AD	9A	MOVZBL	TMPNAM+58, R2	:	
F8 AD	50	52	C1	00072	ADDL3	R2, R0, FDESC	:	
		88	AD	D5	TSTL	TMPNAM+64	:	0548
		06	13	0007A	BEQL	1\$	:	
	50	88	AD	D0	MOVL	TMPNAM+64, R0	:	0549
		04	11	00080	BRB	2\$	:	
	50	8C	AD	D0	MOVL	TMPNAM+68, R0	:	0550
FC	AD	50	D0	00086	MOVL	R0, FDESC+4	:	0548
	50	04	AC	D0	MOVL	DATIM, R0	:	0554
		60	DD	0008E	PUSHL	(R0)	:	
		04	A0	DD	PUSHL	4(R0)	:	0553
		F8	AD	9F	PUSHAB	FDESC	:	0552
		08	AC	DD	PUSHL	DESC	:	
		08	AC	DD	PUSHL	DESC	:	
		FE29	CF	9F	PUSHAB	MAIL\$SD_EXTMSGNAM	:	
00000000G	00	06	FB	000A0	CALLS	#6, SYSSFA0	:	
		04	00	000A7	RET		:	0556

; Routine Size: 168 bytes, Routine Base: \$CODE\$ + 0187


```
417 0557 1 ROUTINE READ_SAVE_EXTMSG =
418 0558 1 ++
419 0559 1 FUNCTIONAL DESCRIPTION:
420 0560 1
421 0561 1 Read a message which is contained in an external message file
422 0562 1
423 0563 1 --
424 0564 2 BEGIN
425 0565 2 LOCAL
426 0566 2 STATUS,
427 0567 2 TMPFAB : $FAB_DECL,
428 0568 2 TMPRAB : $RAB_DECL,
429 0569 2 TMPNAM : $NAM_DECL,
430 0570 2 NAMBUF : $BBLOCK[NAM$C_MAXRSS],
431 0571 2 TMPXAB : $XABFHC_DECL,
432 0572 2 TMPXAB1 : $XABPRO_DECL,
433 0573 2 LASTBLOCK : REF $BBLOCK,
434 0574 2 PTR : REF $BBLOCK,
435 0575 2 DESC : VECTOR[2,LONG],
436 0576 2 BUFSIZ,
437 0577 2 NBYTES,
438 0578 2 NBLOCKS;
439 0579 2
440 0580 2 MAIL$CURMSG[MMB_V_EXTFNF] = FALSE;
441 0581 2 DESC[0] = NAM$C_MAXRSS;
442 0582 2 DESC[1] = NAMBUF;
443 0583 2 MAIL$FORM_EXTFILE(MAIL$_CURMSG[MMB_Q_DATID],DESC);
444 0584 2
445 0585 2 Open the external message file
446 0586 2
447 P 0587 2 $FAB_INIT(FAB=TMPFAB,
448 P 0588 2 FAC=(GET,BRO),
449 P 0589 2 FOP=SQO,
450 P 0590 2 FNS=.DESC[0],
451 P 0591 2 FNA=.DESC[1],
452 P 0592 2 CTX=MAIL$OPENIN,
453 P 0593 2 XAB=TMPXAB,
454 0594 2 NAM=TMPNAM);
455 0595 2
456 P 0596 2 $XABFHC_INIT(XAB=TMPXAB,
457 0597 2 NXT=TMPXAB1);
458 0598 2
459 P 0599 2 $XABPRO_INIT(XAB=TMPXAB1,
460 0600 2 NXT=0);
461 0601 2
462 P 0602 2 $RAB_INIT(RAB=TMPRAB,
463 P 0603 2 ROP=BIO,
464 0604 2 FAB=TMPFAB);
465 0605 2
466 P 0606 2 $NAM_INIT(NAM=TMPNAM,
467 P 0607 2 ESS=NAM$C_MAXRSS,
468 P 0608 2 ESA=NAMBUF,
469 P 0609 2 RSS=NAM$C_MAXRSS,
470 0610 2 RSA=NAMBUF);
471 0611 2
472 P 0612 2 IF_ERR($OPEN(FAB=TMPFAB,ERR=UTIL$REPORT_IO_ERROR);,
473 P 0613 2 MAIL$_CURMSG[MMB_V_EXTFNF] = TRUE;
```

: R

```
474 0614 2 RETURN .STATUS);
475 0615 2
476 0616 2
477 0617 2 If file is not variable length sequential, then we have to
478 0618 2 read in records
479 0619 2
480 0620 2 IF .TMPFAB[FAB$B_RFM] NEQ FAB$C_VAR
481 0621 2 OR .TMPFAB[FAB$B_ORG] NEQ FAB$C_SE2
482 0622 2 THEN MAIL$CURMSG[MMB_V_EXTNSTD] = NOT (TMPRAB[RAB$V_BIO] = FALSE);
483 0623 2
484 P 0624 2 IF_ERR($CONNECT(RAB=TMPRAB,ERR=UTIL$REPORT_IO_ERROR);,
485 P 0625 2 $CLOSE(FAB=TMPFAB);
486 P 0626 2 MAIL$CURMSG[MMB_V_EXTFNF] = TRUE;
487 0627 2 RETURN .STATUS);
488 0628 2
489 0629 2 IF .TMPRAB[RAB$V_BIO]
490 0630 3 THEN BEGIN
491 0631 3 BUFS'Z = (.TMPXAB[XAB$L_EBK]*512);
492 0632 3 MAIL$CURMSG[MMB_L_NBLKS] = .BUFSIZ; !Compute bytes, converted later
493 0633 3 NBYTES = ((.TMPXAB[XAB$L_EBK]-1)*512)+.TMPXAB[XAB$W_FFB];
494 0634 3 IF .BUFSIZ EQL 0
495 0635 4 THEN ($CLOSE(FAB=TMPFAB);
496 0636 3 RETURN TRUE);
497 0637 3
498 P 0638 3 IF_ERR(GET MEM(TXT_C_LENGTH+.BUFSIZ,PTR);,
499 P 0639 3 $CLOSE(FAB=TMPFAB);
500 P 0640 3 MAIL$CURMSG[MMB_V_EXTFNF] = TRUE;
501 0641 3 RETURN .STATUS);
502 0642 3
503 0643 3 PTR[TXT_L_NEXT] = 0;
504 0644 3 MAIL$CURMSG[MMB_L_TXTLST] = .PTR;
505 0645 3 PTR[TXT_L_TEXTLEN] = .NBYTES;
506 0646 3 PTR[TXT_L_ALLOC] = .BUFSIZ;
507 0647 3 BUFSIZ = .BUFSIZ/512;
508 0648 3 TMPRAB[RAB$L_CTX] = MAIL$READERR;
509 0649 3 TMPRAB[RAB$L_UBF] = PTR[TXT_T_TEXT];
510 0650 3 WHILE .BUFSIZ GTR 0
511 0651 4 DO BEGIN
512 0652 4 NBLOCKS = MINU(32,.BUFSIZ);
513 0653 4 TMPRAB[RAB$W_USZ] = .NBLOCKS*512;
514 P 0654 4 IF_ERR($READ(RAB=TMPRAB,ERR=UTIL$REPORT_IO_ERROR);,
515 P 0655 4 $CLOSE(FAB=TMPFAB);
516 P 0656 4 IF .NBYTES NEQ 0
517 P 0657 4 THEN MAIL$CURMSG[MMB_V_EXTFNF] = TRUE;
518 0658 4 RETURN .STATUS);
519 0659 4 TMPRAB[RAB$L_UBF] = .TMPRAB[RAB$L_UBF] + .TMPRAB[RAB$W_RSZ];
520 0660 4 BUFSIZ = .BUFSIZ - .NBLOCKS;
521 0661 3 END;
522 0662 3 END
523 0663 3 ELSE BEGIN
524 0664 3
525 0665 3 Record format is not variable length, read file in record mode
526 0666 3
527 0667 3 LASTBLOCK = MAIL$CURMSG[MMB_L_TXTLST];
528 0668 3 MAIL$CURMSG[MMB_L_NBLKS] = 0;
529 0669 3 TMPRAB[RAB$L_UBF] = .MAIL$GL_BIGBUFFER;
530 0670 3 TMPRAB[RAB$W_USZ] = MAIL$K_INBUFSZ;
```

```
531 0671 3 WHILE (STATUS = $GET(RAB=TMPRAB,ERR=UTIL$REPORT_IO_ERROR)) NEQ RMSS_EOF
532 0672 4 DO BEGIN
533 P 0673 4 IF_ERR(GET MEM(TXT C LENGTH+TMPRAB[RAB$W_RSZ],PTR);,
534 P 0674 4 SIGNAL(.STATUS);
535 P 0675 4 MAIL$L_CURMSG[MMB V_EXTFNF] = TRUE;
536 P 0676 4 $CLOSE(FAB=TMPFAB);
537 0677 4 RETURN .STATUS);
538 0678 4 PTR[TXT_L_NEXT] = 0;
539 0679 4 LASTBLOCK[TXT_L_NEXT] = .PTR;
540 0680 4 LASTBLOCK = .PTR;
541 0681 4 MAIL$L_CURMSG[MMB_L_NBLKS] = .MAIL$L_CURMSG[MMB_L_NBLKS] +
542 0682 4 .TMPRAB[RAB$W_RSZ] + 2;
543 0683 4 PTR[TXT_L_TEXTLEN] = .TMPRAB[RAB$W_RSZ];
544 0684 4 PTR[TXT_L_ALLOC] = .PTR[TXT_L_TEXTLEN];
545 0685 4 CH$MOVE(TMPRAB[RAB$W_RSZ],TMPRAB[RAB$L_RBF],PTR[TXT_T_TEXT]);
546 0686 3 END;
547 0687 2 END;
548 0688 2 $CLOSE(FAB=TMPFAB);
549 0689 2 RETURN TRUE
550 0690 1 END;
```

```
.EXTRN SYSSOPEN, SYSSCONNECT
.EXTRN SYSSCLOSE, SYSSREAD
.EXTRN SYSSGET
```

OFFC 00000 READ_SAVE_EXTMSG:

```
.WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11
MOVAB GET MEM, R11
MOVAB UTIL$REPORT_IO_ERROR, R10
MOVAB SYSSCLOSE, R9
MOVAB MAIL$L_CURMSG, R8
MOVAB -644(SP), SP
MOVAB MAIL$L_CURMSG, R0
BICB2 #16, 10(R0)
MOVZBL #255, DESC
MOVAB NAMBUF, DESC+4
PUSHAB DESC
PUSHAB 55(R0)
CALLS #2, MAIL$FORM_EXTFILE
MOVCS #0, (SP), #0, #80, $RMS_PTR
```

```
MOVW #20483, $RMS_PTR
MOVZBL #64, $RMS_PTR+4
MOVW #66, $RMS_PTR+22
MOVL #8261784, $RMS_PTR+24
MOVW #2, $RMS_PTR+32
MOVAB TMPXAB, $RMS_PTR+36
MOVAB TMPNAM, $RMS_PTR+40
MOVL DESC+4, $RMS_PTR+44
MOVW DESC, $RMS_PTR+52
MOVCS #0, (SP), #0, #44, $RMS_PTR
```

```
MOVW #11293, $RMS_PTR
MOVAB TMPXAB1, $RMS_PTR+4
MOVCS #0, (SP), #0, #88, $RMS_PTR
```

```
5B FE23 CF 9E 00002
5A 00000000G 00 9E 00007
59 00000000G 00 9E 0000E
58 00000000' 00 9E 00015
5E FD7C CE 9E 0001C
50 68 D0 00021
0A A0 10 8A 00024
04 AE FF 8F 9A 00028
08 AE 0090 CE 9E 0002D
04 AE 9F 00033
37 A0 9F 00036
012F CB 02 FB 00039
6E 00 2C 0003E
B0 AD 00045
B0 AD 5003 8F B0 00047
B4 AD 40 8F 9A 0004D
C6 AD 42 8F 90 00052
C8 AD 007E1098 8F D0 00057
CF AD 02 90 0005F
D4 AD 64 AE 9E 00063
D8 AD FF0C CD 9E 00068
DC AD 08 AE D0 0006E
E4 AD 04 AE 90 00073
2C 6E 00 2C 00078
64 AE 0007D
68 AE 2C1D 8F B0 0007F
AE 9E 00085
6E 00 2C 0008A
```

0557

0580

0581

0582

0583

0594

0597

0600

0044	BF	00	OC	AE	5813	8F	BO	00091	MOVW	#22547, SRMS_PTR	
				6E		00	2C	00093	MOVCS	#0, (SP), #0, #68, SRMS_PTR	0604
			FF6C	CD	4401	8F	BO	000A0			
			FF70	CD	0800	8F	3C	000A3	MOVW	#17409, SRMS_PTR	
0060	8F	00	A8	AD	BO	AD	9E	000B1	MOVZWL	#2048, SRMS_PTR+4	
				6E		00	2C	000B6	MOVAB	TMPFAB, SRMS_PTR+60	0610
			FF0C	CD	6002	00	2C	000BD	MOVCS	#0, (SP), #0, #96, SRMS_PTR	
			FF0E	CD		8F	BO	000C0	MOVW	#24578, SRMS_PTR	
			FF10	CD	0090	01	8E	000C7	MNEGB	#1, SRMS_PTR+2	
			FF16	CD	0090	CE	9E	000CC	MOVAB	NAMBUF, SRMS_PTR+4	
			FF18	CD	0090	01	8E	000D3	MNEGB	#1, SRMS_PTR+10	
						CE	9E	000D8	MOVAB	NAMBUF, SRMS_PTR+12	
					BO	5A	DD	000DF	PUSHL	R10	0614
		00000000G				AD	9F	000E1	PUSHAB	TMPFAB	
			08			02	FB	000E4	CALLS	#2, SYSS\$OPEN	
			51			50	E8	000EB	BLBS	STATUS, 1\$	
		OA	A1			68	D0	000EE	MOVL	MAIL\$L_CURMSG, R1	
						10	88	000F1	BISB2	#16, 10(R1)	
			02		CF	04	00	00F5	RET		
						AD	91	000F6	CMPB	TMPFAB+31, #2	0620
						05	12	000FA	BNEQ	2\$	
					CD	AD	95	000FC	TSTB	TMPFAB+29	0621
						0D	13	000FF	BEQL	3\$	
			50			68	D0	00101	MOVL	MAIL\$L_CURMSG, R0	0622
		FF71	CD			08	8A	00104	BICB2	#8, TMPRAB+5	
		OA	A0		40	8F	88	00109	BISB2	#64, 10(R0)	
						5A	DD	0010E	PUSHL	R10	0627
					FF6C	CD	9F	00110	PUSHAB	TMPRAB	
		00000000G				02	FB	00114	CALLS	#2, SYSS\$CONNECT	
			52			50	D0	0011B	MOVL	R0, STATUS	
			10			52	E8	0011E	BLBS	STATUS, 4\$	
					BO	AD	9F	00121	PUSHAB	TMPFAB	
			69			01	FB	00124	CALLS	#1, SYSS\$CLOSE	
			50			68	D0	00127	MOVL	MAIL\$L_CURMSG, R0	
		OA	A0			10	88	0012A	BISB2	#16, 10(R0)	
						010F	31	0012E	BRW	16\$	
			50			68	D0	00131	MOVL	MAIL\$L_CURMSG, R0	0632
03	FF71	CD				03	E0	00134	BBS	#3, TMPRAB+5, 5\$	0629
						00AD	31	0013A	BRW	14\$	
52	74	AE				09	78	0013D	ASHL	#9, TMPXAB+16, BUFSIZ	0631
	3F	A0				52	D0	00142	MOVL	BUFSIZ, 63(R0)	0632
53	74	AE				09	78	00146	ASHL	#9, TMPXAB+16, R3	0633
		50			78	AE	3C	0014B	MOVZWL	TMPXAB+20, R0	
		53				50	C0	0014F	ADDL2	R0, R3	
		53			FE00	C3	7F	00152	MOVAB	-512(R3), NBYTES	
						52	D5	00157	TSTL	BUFSIZ	0634
						42	13	00159	BEQL	8\$	
						5E	DD	0015B	PUSHL	SP	0641
					OC	A2	9F	0015D	PUSHAB	12(BUFSIZ)	
			68			02	FB	00160	CALLS	#2, GET MEM	
			54			50	D0	00163	MOVL	R0, STATUS	
			08			54	E8	00166	BLBS	STATUS, 6\$	
					BO	AD	9F	00169	PUSHAB	TMPFAB	
			69			01	FB	0016C	CALLS	#1, SYSS\$CLOSE	
						61	11	0016F	BRB	11\$	

	50	6E	D0	00171	6\$:	MOVL	PTR, R0	:	0643	
		60	D4	00174		CLRL	(R0)	:		
	51	68	D0	00176		MOVL	MAIL\$L CURMSG, R1	:	0644	
06	A1	50	D0	00179		MOVL	R0, 6(R1)	:		
04	A0	53	D0	0017D		MOVL	NBYTES, 4(R0)	:	0645	
08	A0	52	D0	00181		MOVL	BUFSIZ, 8(R0)	:	0646	
	52	8F	C6	00185		DIVL2	#512, BUFSIZ	:	0647	
84	AD	007E10B0	8F	D0	0018C	MOVL	#8261808, TMPRAB+24	:	0648	
90	AD	0C	A0	9E	00194	MOVAB	12(R0), TMPRAB+36	:	0649	
			52	D5	00199	7\$:	TSTL	BUFSIZ	:	0650
			03	14	0019B		BGTR	9\$	:	
			00D2	31	0019D	8\$:	BRW	18\$	:	
	50		52	D0	001A0	9\$:	MOVL	BUFSIZ, R0	:	0652
	20		50	D1	001A3		CMPL	R0, #32	:	
			03	1B	001A6		BLEQU	10\$	:	
	50		20	D0	001A8		MOVL	#32, R0	:	
	55		50	D0	001AB	10\$:	MOVL	R0, NBLOCKS	:	
8C	AD		8F	A5	001AE		MULW3	#512, NBLOCKS, TMPRAB+32	:	0653
		0200	5A	DD	001B5		PUSHL	R10	:	0658
		FF6C	CD	9F	001B7		PUSHAB	TMPRAB	:	
	00000000G		02	FB	001BB		CALLS	#2, SYSS\$READ	:	
			50	D0	001C2		MOVL	R0, STATUS	:	
			54	E8	001C5		BLBS	STATUS, 13\$	:	
		B0	AD	9F	001C8		PUSHAB	TMPIAB	:	
			01	FB	001CB		CALLS	#1, SYSS\$CLOSE	:	
			53	D5	001CE		TSTL	NBYTES	:	
			07	13	001D0		BEQL	12\$	:	
	50		68	D0	001D2	11\$:	MOVL	MAIL\$L CURMSG, R0	:	
0A	A0		10	88	001D5		BISB2	#16, 10(R0)	:	
	50		54	D0	001D9	12\$:	MOVL	STATUS, R0	:	
			04	001DC		RET		:		
	50	8E	AD	3C	001DD	13\$:	MOVZWL	TMPRAB+34, R0	:	0659
	AD		50	C0	001E1		ADDL2	R0, TMPRAB+36	:	
	52		55	C2	001E5		SUBL2	NBLOCKS, BUFSIZ	:	0660
			AF	11	001E8		BRB	7\$	:	0650
	56	06	A0	9E	001EA	14\$:	MOVAB	6(R0), LASTBLOCK	:	0667
		3F	A0	D4	001EE		CLRL	63(R0)	:	0668
90	AD	00000000G	00	D0	001F1		MOVL	MAIL\$GL BIGBUFFER, TMPRAB+36	:	0669
8C	AD	0200	8F	B0	001F9		MOVW	#512, TMPRAB+32	:	0670
			5A	DD	001FF	15\$:	PUSHL	R10	:	0671
		FF6C	CD	9F	00201		PUSHAB	TMPRAB	:	
	00000000G		02	FB	00205		CALLS	#2, SYSS\$GET	:	
			50	D0	0020C		MOVL	R0, STATUS	:	
	0001827A		57	D1	0020F		CMPL	STATUS, #98938	:	
			5A	13	00216		BEQL	18\$	:	
			5E	DD	00218		PUSHL	SP	:	0677
	7E	8E	AD	3C	0021A		MOVZWL	TMPRAB+34, -(SP)	:	
	6E		0C	C0	0021E		ADDL2	#12, (SP)	:	
	68		02	FB	00221		CALLS	#2, GET MEM	:	
	52		50	D0	00224		MOVL	R0, STATUS	:	
	1A		52	E8	00227		BLBS	STATUS, 17\$	:	
			52	DD	0022A		PUSHL	STATUS	:	
	00000000G		01	FB	0022C		CALLS	#1, LIB\$SIGNAL	:	
			68	D0	00233		MOVL	MAIL\$L CURMSG, R0	:	
	0A		10	88	00236		BISB2	#16, 10(R0)	:	
		B0	AD	9F	0023A		PUSHAB	TMPIAB	:	
			01	FB	0023D		CALLS	#1, SYSS\$CLOSE	:	

	50		52	DU	0C	40	16\$:	MOVL	STATUS, R0	:	
				04	00	243		RET		:	
	51		6E	D0	00	244	17\$:	MOVL	PTR, R1	:	0678
			61	D4	00	247		CLRL	(R1)	:	
	66		51	D0	00	249		MOVL	R1, (LASTBLOCK)	:	0679
	56		51	D0	00	24C		MOVL	R1, LASTBLOCK	:	0680
	50		68	D0	00	24F		MOVL	MAIL\$ CURMSG, R0	:	0681
	52	8E	AD	3C	00	252		MOVZWL	TMPRAB+34, R2	:	0682
	52	3F	A0	C0	00	256		ADDL2	63(R0), R2	:	
3F	A0	02	A2	9E	00	25A		MOVAB	2(R2), 63(R0)	:	
04	A1	8E	AD	3C	00	25F		MOVZWL	TMPRAB+34, 4(R1)	:	0683
08	A1	04	A1	D0	00	264		MOVL	4(R1), 8(R1)	:	0684
OC	A1	94	BD	8E	AD	28	00269	MOV3	TMPRAB+34, @TMPRAB+40, 12(R1)	:	0685
				8D	11	00270		BRB	15\$	:	0671
			B0	AD	9F	00272	18\$:	PUSHAB	TMPFAB	:	0688
	69			01	FB	00275		CALLS	#1, SYS\$CLOSE	:	
	50			01	D0	00278		MOVL	#1, R0	:	0689
				04	00	27B		RET		:	0690

; Routine Size: 636 bytes, Routine Base: \$CODE\$ + 022F


```
552 0691 1 ROUTINE READ_SAVE_MSG =
553 0692 1 ++
554 0693 1 FUNCTIONAL DESCRIPTION:
555 0694 1
556 0695 1 Read the text records of the current message and stash them in
557 0696 1 memory
558 0697 1
559 0698 1 --
560 0699 2 BEGIN
561 0700 2
562 0701 2 LOCAL
563 0702 2 LASTBLOCK : REF $BBLOCK,
564 0703 2 PTR : REF $BBLOCK,
565 0704 2 RECLN,
566 0705 2 STATUS;
567 0706 2
568 0707 2 MAIL$$_CURMSG[MMB_L_NBLKS] = 0;
569 0708 2
570 0709 2 IF .MAIL$$_CURMSG[MMB_V_EXTMSG]
571 0710 2 THEN RETURN READ_SAVE_EXTMSG();
572 0711 2
573 0712 2 MAIL$$_RD_RAB[RAB$$_ROP] = RAB$$_LIM OR RAB$$_LOC OR RAB$$_NLK;
574 0713 2 MAIL$$_RD_RAB[RAB$$_KBF] = MAIL$$_CURMSG[MMB_Q_DATID];
575 0714 2 MAIL$$_RD_RAB[RAB$$_KSZ] = 8;
576 0715 2 MAIL$$_RD_RAB[RAB$$_KRF] = 0;
577 0716 2 MAIL$$_RD_RAB[RAB$$_RAC] = RAB$$_KEY;
578 0717 2 MAIL$$_RD_RAB[RAB$$_CTX] = MAIL$$_READERR;
579 0718 2 MAIL$$_RD_RAB[RAB$$_UBF] = .MAIL$$_RECBUF;
580 P 0719 2 IF _ERR($FIND(RAB=MAIL$$_RD_RAB));
581 P 0720 2 IF .STATUS EQL RMSS_RNF
582 P 0721 2 THEN RETURN TRUE;
583 P 0722 2 UTIL$REPORT IO_ERROR(MAIL$$_RD_RAB);
584 0723 2 RETURN .STATUS;
585 0724 2 MAIL$$_RD_RAB[RAB$$_RAC] = RAB$$_SEQ;
586 0725 2
587 0726 2 Read in all records of this message
588 0727 2
589 0728 2 LASTBLOCK = MAIL$$_CURMSG[MMB_L_TXTLST];
590 0729 2 WHILE 1
591 0730 3 DO BEGIN
592 0731 3 STATUS = $GET(RAB=MAIL$$_RD_RAB);
593 0732 3 IF NOT .STATUS
594 0733 4 THEN BEGIN
595 0734 4 IF .STATUS NEQ RMSS_RNF
596 0735 4 AND .STATUS NEQ RMSS_EOF
597 0736 4 THEN RETURN UTIL$REPORT_IO_ERROR(MAIL$$_RD_RAB);
598 0737 4 EXITLOOP;
599 0738 3 END;
600 0739 3 IF .STATUS EQL RMSS_OK_LIM
601 0740 3 THEN EXITLOOP;
602 0741 3 IF .MAIL$$_RD_RAB[RAB$$_RSZ] GTRU 8 !Skip marker records
603 0742 4 THEN BEGIN
604 0743 4 RECLN = .MAIL$$_RD_RAB[RAB$$_RSZ] - $BYTEOFFSET(MFR_T_DATA);
605 P 0744 4 IF _ERR(GET_MEM(TXT T LENGTH+.RECLN,PTR));
606 0745 4 RETURN .STATUS;
607 0746 4 PTR[TXT_L_NEXT] = .LASTBLOCK[TXT_L_NEXT];
608 0747 4 LASTBLOCK[TXT_L_NEXT] = .PTR;
```

```

: 609      0748 4      LASTBLOCK = .PTR;
: 610      0749 4      MAIL$$_CURMSG[MMB_L_NBLKS] = .MAIL$$_CURMSG[MMB_L_NBLKS] + .RECLEN + 2;
: 611      0750 4      PTR[TEXT_L_TEXTLEN] = .RECLEN;
: 612      0751 4      PTR[TEXT_L_ALLOC] = .RECLEN;
: 613      0752 4      CH$MOVE(.RECLEN, .MAIL$$_RD_RAB[RAB$$_RBF] + $BYTEOFFSET(MFR_T_DATA),
: 614      0753 4      PTR[TEXT_T_TEXT]);
: 615      0754 3      END;
: 616      0755 2      END;
: 617      0756 2
: 618      0757 2      RETURN TRUE
: 619      0758 1      END;
```

.EXTRN SYSS\$FIND

```

OFFC 00000 READ_SAVE MSG:
: 0691      5B 00000000G 00 9E 00002      .WORD      Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11
:          5A 00000000' 00 9E 00009      MOVAB      UTIL$REPORT_IO_ERROR, R11
:          59 00000000G 00 9E 00010      MOVAB      MAIL$$_CURMSG, R10
:          5E          04 C2 00017      MOVAB      MAIL$$_RD_RAB, R9
:          50          6A D0 0001A      SUBL2      #4, SP
:          3F          A0 D4 0001D      MOVL       MAIL$$_CURMSG, R0
:          06          0A A0 03 E1 00020      CLRL      63(R0)
:          FD5A        CF 00 FB 00025      BBC       #3, 10(R0), 1$
:          04          A9 00114000 8F D0 0002B 1$:      CALLS    #0, READ_SAVE_EXTMSG
:          30          A9          37 C1 00033      RET
:          34          A9          08 B0 0003B      MOVL     #1130496, MAIL$$_RD_RAB+4
:          1E          A9          01 90 0003C      ADDL3    #55, MAIL$$_CURMSG, MAIL$$_RD_RAB+48
:          18          A9 007E10B0 8F D0 00040      MOVW     #8, MAIL$$_RD_RAB+52
:          24          A9 00000000G 00 D0 00048      MOV      #1, MAIL$$_RD_RAB+30
:          59          DD 00050      MOV      #8261808, MAIL$$_RD_RAB+24
:          00000000G 00          59 DD 00050      MOVL     MAIL$$_RECBUF, MAIL$$_RD_RAB+36
:          52          01 FB 00052      PUSHL    R9
:          12          50 D0 00059      CALLS    #1, SYSS$FIND
:          000182B2 8F          52 E8 0005C      MOVL     R0, STATUS
:          3E          13 00066      BLBS     STATUS, 2$
:          59          DD 00068      CMPL     STATUS, #98994
:          68          01 FB 0006A      BEQL     5$
:          50          52 D0 0006D      PUSHL    R9
:          1E          A9 94 00071 2$:      CALLS    #1, UTIL$REPORT_IO_ERROR
:          58          6A          06 C1 00074      MOVL     STATUS, R0
:          59          DD 00078 3$:      RET
:          00000000G 00          01 FB 0007A      CLRB     MAIL$$_RD_RAB+30
:          57          50 D0 00081      ADDL3    #6, MAIL$$_CURMSG, LASTBLOCK
:          18          57 E8 00084      PUSHL    R9
:          000182B2 8F          57 D1 00087      CALLS    #1, SYSS$GET
:          0001827A 8F          5F 13 0008E      MOVL     R0, STATUS
:          56          13 00097      BLBS     STATUS, 4$
:          68          01 FB 0009B      CMPL     STATUS, #98994
:          00018051 8F          57 D1 0009F 4$:      BEQL     6$
:          57          DD 00099      CMPL     STATUS, #98938
:          59          DD 00099      BEQL     6$
:          57          DD 00099      BEQL     6$
:          68          01 FB 0009B      PUSHL    R9
:          00018051 8F          57 D1 0009F 4$:      CALLS    #1, UTIL$REPORT_IO_ERROR
:          57          DD 00099      RET
:          57          D1 0009F 4$:      CMPL     STATUS, #98385
```


		50	22	47	13	000A6	5\$:	BEQL	6\$	:	
		08		A9	3C	000A8		MOVZWL	MAIL\$T_RD_RAB+34, R0	:	0741
				50	B1	000AC		CMPW	R0, #8	:	
		56	D0	C7	1B	000AF		BLEQU	3\$	:	
				A0	9E	000B1		MOVAB	-48(R0), RECLN	:	0743
			0C	5E	DD	000B5		PUSHL	SP	:	0745
	FAEE	CF		A6	9F	000B7		PUSHAB	12(RECLN)	:	
		30		02	FB	000BA		CALLS	#2, GET_MEM	:	
		51		50	E9	000BF		BLBC	STATUS, -7\$	:	
		61		6E	D0	000C2		MOVL	PTR, R1	:	0746
		68		68	D0	000C5		MOVL	(LASTBLOCK), (R1)	:	
		58		51	D0	000C8		MOVL	R1, (LASTBLOCK)	:	0747
		50		51	D0	000CB		MOVL	R1, LASTBLOCK	:	0748
		56	3F	6A	D0	000CE		MOVL	MAIL\$L_CURMSG, R0	:	0749
52		56		A0	C1	000D1		ADDL3	63(R0), RECLN, R2	:	
	3F	A0	02	A2	9E	000D6		MOVAB	2(R2), 63(R0)	:	
	04	A1		56	D0	000DB		MOVL	RECLN, 4(R1)	:	0750
	08	A1		56	D0	000DF		MOVL	RECLN, 8(R1)	:	0751
		50	28	A9	D0	000E3		MOVL	MAIL\$T_RD_RAB+40, R0	:	0752
0C	A1	30		56	28	000E7		MOVC3	RECLN, 48(R0), 12(R1)	:	0753
		50		89	11	000ED		BRB	3\$	:	0729
				01	D0	000EF	6\$:	MOVL	#1, R0	:	0757
				04	000F2	7\$:		RET		:	0758

; Routine Size: 243 bytes, Routine Base: \$CODE\$ + 04AB

```
0759 1 ROUTINE DEALLOC_MSG =
0760 1 **
0761 1 FUNCTIONAL DESCRIPTION:
0762 1
0763 1 Deallocate all storage used for current msg
0764 1
0765 1 IMPLICIT INPUTS:
0766 1
0767 1 MAIL$$_CURMSG points at message descriptor block
0768 1 --
0769 2 BEGIN
0770 2
0771 2 LOCAL
0772 2   PTR : REF $BLOCK,
0773 2   BUFSIZ,
0774 2   PTR1;
0775 2
0776 2 IF .MAIL$$_CURMSG EQL 0
0777 2 THEN RETURN TRUE;
0778 2
0779 2 PTR = .MAIL$$_CURMSG[MMB_L_TXTLST];
0780 2 IF_ERR(LIB$FREE_VM(%REF(MMB_C_LENGTH+.MAIL$$_CURMSG[MMB_W_HDRECLN])),
0781 2   MAIL$$_CURMSG);
0782 2 SIGNAL((.STATUS AND NOT STS$M_SEVERITY) OR STS$K_ERROR));
0783 2 MAIL$$_CURMSG = 0;
0784 2 IF .PTR NEQ 0
0785 2 THEN WHILE .PTR NEQ 0
0786 2 DO BEGIN
0787 2   PTR1 = .PTR[txt_L_NEXT];
0788 2   BUFSIZ = txt_C_LENGTH+.PTR[txt_L_TEXTLEN];
0789 2   IF_ERR(LIB$FREE_VM(%REF(txt_C_LENGTH+.PTR[txt_L_ALLOC]),PTR);,
0790 2     SIGNAL((.STATUS AND NOT STS$M_SEVERITY) OR STS$K_ERROR));
0791 2   PTR = .PTR1;
0792 2 END;
0793 2 RETURN TRUE
0794 1 END;
```

```
007C 00000 DEALLOC_MSG:
56 00000000G 00 9E 00002 .WORD Save R2,R3,R4,R5,R6
55 00000000G 00 9E 00009 MOVAB LIB$$_SIGNAL, R6
54 00000000' 00 9E 00010 MOVAB LIB$$_FREE_VM, R5
5E 08 C2 00017 MOVAB MAIL$$_CURMSG, R4
50 64 D0 0001A SUBL2 #8, SP
5E 13 0001D MOVL MAIL$$_CURMSG, R0
04 AE 06 A0 D0 0001F BEQL 4$
54 DD 00024 MOVL 6(R0), PTR
04 AE 11 A0 3C 00026 PUSHL R4
04 AE 0000047 8F C0 0002B MOVZWL 17(R0), 4(SP)
65 02 FB 00033 ADDL2 #71, 4(SP)
0A 50 E8 00036 PUSHAB 4(SP)
50 07 8A 0003C CALLS #2, LIB$$_FREE_VM
BLBS STATUS, 1$
BICB2 #7, R0
```

0759

0776

0779

0782

1
1

; Routine Size: 129 bytes, Routine Base: \$CODE\$ + 059E

```
658 0795 1 ROUTINE FIND_FILE_TYPE (IFAB) =
659 0796 1 **
660 0797 1 FUNCTIONAL DESCRIPTION:
661 0798 1
662 0799 1 This routine returns the file type in IFAB[FAB$B_ORG]. This is needed
663 0800 1 if RMS returns RMSS_FLK.
664 0801 1
665 0802 1 Inputs:
666 0803 1
667 0804 1 IFAB = addr of FAB after $OPEN has been tried (must have an attached
668 0805 1 NAM block
669 0806 1 --
670 0807 2 BEGIN
671 0808 2 MAP
672 0809 2 IFAB : REF $BBLOCK;
673 0810 2
674 0811 2 BIND
675 0812 2 INAM = .IFAB[FAB$L_NAM] : $BBLOCK;
676 0813 2
677 0814 2 LOCAL
678 0815 2 STATUS,
679 0816 2 SAVE_STS,
680 0817 2 SAVE_STV,
681 0818 2 IOSB : VECTOR[4,WORD],
682 0819 2 CHANNEL : WORD,
683 0820 2 FIB_DESC : VECTOR[2, LONG],
684 0821 2 FIB : $BBLOCK[FIB$C_ACCDATA],
685 0822 2 ATR : BLOCKVECTOR[1,8,BYTE],
686 0823 2 FILATR : $BBLOCK[ATR$S_PECATTR];
687 0824 2
688 0825 2 SAVE_STS = .IFAB[FAB$L_STS];
689 0826 2 SAVE_STV = .IFAB[FAB$L_STV];
690 0827 3 IF (STATUS = $PARSE(FAB=.IFAB))
691 0828 2 THEN (STATUS = $SEARCH(FAB=.IFAB));
692 0829 2 IFAB[FAB$L_STS] = .SAVE_STS;
693 0830 2 IFAB[FAB$L_STV] = .SAVE_STV;
694 0831 2 IF NOT .STATUS
695 0832 2 THEN RETURN .STATUS;
696 0833 2
697 0834 2 Assign a channel to the device
698 0835 2
699 0836 3 FIB_DESC[0] = (IF .INAM[NAM$B_RSL] NEQ 0
700 0837 3 THEN .INAM[NAM$B_RSL]
701 0838 2 ELSE .INAM[NAM$B_ESL]);
702 0839 2 FIB_DESC[1] = .INAM[NAM$L_RSA];
703 P 0840 3 IF NOT (STATUS = $ASSIGN(DEVNAM=FIB_DESC,
704 0841 3 CHAN=CHANNE[))
705 0842 2 THEN RETURN .STATUS;
706 0843 2
707 0844 2 CH$FILL(0,FIB$C_ACCDATA,FIB);
708 0845 2 FIB[FIB$L_ACCTL] = 0; !No special options
709 0846 2 FIB[FIB$W_FID_NUM] = .INAM[NAM$W_FID_NUM];
710 0847 2 FIB[FIB$W_FID_SEQ] = .INAM[NAM$W_FID_SEQ];
711 0848 2 FIB[FIB$W_FID_RVN] = .INAM[NAM$W_FID_RVN];
712 0849 2
713 0850 2 FIB_DESC[0] = FIB$C_ACCDATA;
714 0851 2 FIB_DESC[1] = FIB;
```

; R

```

715 0852 2
716 0853 2 ATR[0,ATR$W_TYPE] = ATR$C_RECATTR;
717 0854 2 ATR[0,ATR$W_SIZE] = ATR$S_RECATTR;
718 0855 2 ATR[0,ATR$W_ADDR] = FILATR;
719 0856 2
720 P 0857 2 STATUS = $QIOW(FUNC=IOS_ACCESS,
721 P 0858 2 CHAN=.CHANNEL,
722 P 0859 2 EFN=0,
723 P 0860 2 IOSB=IOSB,
724 P 0861 2 P1=FIB_DESC,
725 0862 2 P5=ATR);
726 0863 2 IF .STATUS
727 0864 2 THEN STATUS = .IOSB[0];
728 0865 2
729 0866 2 $DASSGN(CHAN=.CHANNEL);
730 0867 2
731 0868 2 IF NOT .STATUS
732 0869 2 THEN RETURN .STATUS;
733 0870 2
734 0871 2 IFAB[FAB$B_ORG] = 0;
735 0872 2 IF .FILATR[FAT$V_FILEORG] EQL FAT$C_INDEXED
736 0873 2 THEN IFAB[FAB$B_ORG] = FAB$C_IDX
737 0874 2 ELSE IF .FILATR[FAT$V_FILEORG] EQL FAT$C_SEQUENTIAL
738 0875 2 THEN IFAB[FAB$B_ORG] = FAB$C_SEQ;
739 0876 2
740 0877 2 RETURN TRUE
741 0878 1 END;
```

```

.EXTRN SYS$SEARCH, SYS$ASSIGN
.EXTRN SYS$QIOW, SYS$DASSGN
```

01FC 00000 FIND_FILE TYPE:

	5E	B8	AE	9E	00002	WORD	Save R2,R3,R4,R5,R6,R7,R8	0795
	56	04	AC	D0	00006	MOVAB	-72(SP), SP	
	57	28	A6	D0	0000A	MOVL	IFAB, R6	0812
	53	08	A6	D0	0000E	MOVL	40(R6), R7	
	52	0C	A6	D0	00012	MOVL	8(R6), SAVE_STS	0825
			56	DD	00016	MOVL	12(R6), SAVE_STV	0826
			01	FB	00018	PUSHL	R6	0827
00000000G	00		50	D0	0001F	CALLS	#1, SYS\$PARSE	
	58		58	E9	00022	MOVL	R0, STATUS	
	0C		56	DD	00025	BLBC	STATUS, 1\$	
			01	FB	00027	PUSHL	R6	0828
00000000G	00		50	D0	0002E	CALLS	#1, SYS\$SEARCH	
	58		53	D0	00031	MOVL	R0, STATUS	
08	A6		52	D0	00035	MOVL	SAVE_STS, 8(R6)	0829
0C	A6		58	E9	00039	MOVL	SAVE_STV, 12(R6)	0830
	2A		03	A7	95	BLBC	STATUS, 4\$	0831
			06	13	0003F	TSTB	3(R7)	0836
	50		03	A7	9A	BEQL	2\$	
			04	11	00045	MOVZBL	3(R7), R0	0837
	50		08	A7	9A	BRB	3\$	
38	AE		50	D0	0004B	MOVZBL	11(R7), R0	0838
3C	AE		04	A7	D0	MOVL	R0, FIB_DESC	0836
						MOVL	4(R7), FIB_DESC+4	0839

0A	00	00000000G	00	08	7E	7C	00054	CLRQ	-(SP)	0841
			58	44	AE	9F	00056	PUSHAB	CHANNEL	
			61		AE	9F	00059	PUSHAB	FIB_DESC	
			6E		04	FB	0005C	CALLS	#4, SYSS\$ASSIGN	
					50	D0	00063	MOVL	R0, STATUS	
					58	E9	00066	BLBC	STATUS, 6\$	
					00	2C	00069	MOVCS	#0, (SP), #0, #10, FIB	0844
				2C	AE		0006E			
				2C	AE	D4	00070	CLRL	FIB	0845
		30	AE	24	A7	D0	00073	MOVL	36(R7), FIB+4	0846
		34	AE	28	A7	B0	00078	MOVW	40(R7), FIB+8	0848
		38	AE		0A	D0	0007D	MOVL	#10, FIB_DESC	0850
		3C	AE	2C	AE	9E	00081	MOVAB	FIB, FIB_DESC+4	0851
		24	AE	20	8F	D0	00086	MOVL	#262176, ATR	0854
		28	AE	04	AE	9E	0008E	MOVAB	FILATR, ATR+4	0855
					7E	D4	00093	CLRL	-(SP)	0862
				28	AE	9F	00095	PUSHAB	ATR	
					7E	7C	00098	CLRQ	-(SP)	
					7E	D4	0009A	CLRL	-(SP)	
				4C	AE	9F	0009C	PUSHAB	FIB_DESC	
					7E	7C	0009F	CLRQ	-(SP)	
				60	AE	9F	000A1	PUSHAB	IOSB	
					32	DD	000A4	PUSHL	#50	
			7E	28	AE	3C	000A6	MOVZWL	CHANNEL, -(SP)	
					7E	D4	000AA	CLRL	-(SP)	
		00000000G	00		0C	FB	000AC	CALLS	#12, SYSS\$Q10W	
			58		50	D0	000B3	MOVL	R0, STATUS	
			04		58	E9	000B6	BLBC	STATUS, 5\$	0863
			58	40	AE	3C	000B9	MOVZWL	IOSB, STATUS	0864
			7E		6E	3C	000BD	MOVZWL	CHANNEL, -(SP)	0866
		00000000G	00		01	FB	000C0	CALLS	#1, SYSS\$DASSGN	
			04		58	E8	000C7	BLBS	STATUS, 7\$	0868
			50		58	D0	000CA	MOVL	STATUS, R0	0869
					04	000CD		RET		
02	04	AE	04	1D	A6	94	000CE	CLRB	29(R6)	0871
					04	ED	000D1	CMPZV	#4, #4, FILATR, #2	0872
					06	12	000D7	BNEQ	8\$	
		1D	A6		20	90	000D9	MOVB	#32, 29(R6)	0873
					0A	11	000DD	BRB	9\$	
		F0	8F	04	AE	93	000DF	BITB	FILATR, #240	0874
					03	12	000E4	BNEQ	9\$	
				1D	A6	94	000E6	CLRB	29(R6)	0875
			50		01	D0	000E9	MOVL	#1, R0	0877
					04	000EC		RET		0878

; Routine Size: 237 bytes, Routine Base: \$CODE\$ + 061F


```
743 0879 1 GLOBAL ROUTINE MAIL$CREATE_ISAM (IFAB,IRAB,INAM,NAMDESC,UIC,FLAGS,DEFDESC) =
744 0880 1 ++
745 0881 1 FUNCTIONAL DESCRIPTION:
746 0882 1
747 0883 1 Create/open the named ISAM file
748 0884 1
749 0885 1 INPUTS:
750 0886 1
751 0887 1 IFAB is fab address
752 0888 1 IRAB is rab address
753 0889 1 INAM is nam block address
754 0890 1 NAMDESC is descriptor for file spec
755 0891 1 UIC is desired owner, or 0 if current
756 0892 1 FLAGS are: MCR_V_CIF to create if, 0 to unconditionally create
757 0893 1 MCR_V_NIC to create with no sharing
758 0894 1 MCR_V_EXEC to restrict to exec mode logicals
759 0895 1 DEFDESC (optional) address of descriptor for default spec. If
760 0896 1 not specified, MAIL$Q_MAILDEF is used.
761 0897 1
762 0898 1 ROUTINE VALUE:
763 0899 1
764 0900 1 RMSS_SHR - file is sequential
765 0901 1 RMSS_FLK - isam file is locked
766 0902 1 1 - SUCCESS
767 0903 1 --
768 0904 2 BEGIN
769 0905 2 BUILTIN
770 0906 2 NULLPARAMETER;
771 0907 2
772 0908 2 MAP
773 0909 2 IFAB : REF $BBLOCK,
774 0910 2 IRAB : REF $BBLOCK,
775 0911 2 INAM : REF $BBLOCK,
776 0912 2 NAMDESC : REF $BBLOCK,
777 0913 2 FLAGS : $BBLOCK,
778 0914 2 DEFDESC : REF $BBLOCK;
779 0915 2
780 0916 2 LOCAL
781 0917 2 STATUS,
782 0918 2 FILE_TYPE,
783 0919 2 DEFLN,
784 0920 2 DEFADR,
785 0921 2 XABALL0 : $XABALL_DECL,
786 0922 2 XABALL1 : $XABALL_DECL,
787 0923 2 XABALL2 : $XABALL_DECL,
788 0924 2 XABKEY0 : $XABKEY_DECL,
789 0925 2 XABKEY1 : $XABKEY_DECL;
790 0926 2
791 0927 2 DEFLN = .MAIL$Q_MAILDEF[DSC$W_LENGTH];
792 0928 2 DEFADR = .MAIL$Q_MAILDEF[DSC$A_POINTER];
793 0929 2 IF NOT NULLPARAMETER(7)
794 0930 3 THEN BEGIN
795 0931 3 DEFLN = .DEFDESC[DSC$W_LENGTH];
796 0932 3 DEFADR = .DEFDESC[DSC$A_POINTER];
797 0933 2 END;
798 P 0934 2 $FAB_INIT(FAB=.IFAB,
P 0935 2 FNS=.NAMDESC[DSC$W_LENGTH],
```

```
800 P 0936 2 FNA=.NAMDESC[DSCSA_POINTER],
801 P 0937 2 DNS=.DEFLN,
802 P 0938 2 DNA=.DEFADR,
803 P 0939 2 FAC=(DEL,PUT,GET,UPD),
804 P 0940 2 NAM=.INAM,
805 P 0941 2 CTX=MAIL$OPENOUT,
806 P 0942 2 SHR=(DEL,PUT,GET,UPD),
807 P 0943 2 RFM=VAR,
808 P 0944 2 ORG=IDX,
809 P 0945 2 MRS=MFR_C MAXLEN,
810 0946 2 XAB=SND_XABPRO);
811 0947 2
812 0948 2 IFAB[FAB$V_DFW] = .MAIL$B_DFW;
813 0949 2 IFAB[FAB$V_CIF] = .FLAGS[MCR_V_CIF];
814 0950 2 IF .FLAGS[MCR_V_NIL]
815 0951 2 THEN IFAB[FAB$B_SHR] = FAB$M_NIL;
816 0952 2 IF .FLAGS[MCR_V_EXEC]
817 0953 2 THEN IFAB[FAB$V_LNM_MODE] = PSL$C_EXEC;
818 0954 2
819 P 0955 2 $XABPRO_INIT(XAB=SND_XABPRO,
820 0956 2 NXT=XABALL0);
821 0957 2 SND_XABPRO[XAB$L_UIC]=.UIC;
822 0958 2 SND_XABPRO[XAB$W_PRO]=SND_PRO;
823 0959 2
824 P 0960 2 $XABALL_INIT(XAB=XABALL0,
825 P 0961 2 AID=0,
826 P 0962 2 AOP=CBT,
827 P 0963 2 BKZ=5,
828 P 0964 2 DEQ=15,
829 0965 2 NXT=XABALL1);
830 0966 2
831 P 0967 2 $XABALL_INIT(XAB=XABALL1,
832 P 0968 2 AID=1,
833 P 0969 2 AOP=CBT,
834 P 0970 2 BKZ=5,
835 P 0971 2 DEQ=5,
836 0972 2 NXT=XABALL2);
837 0973 2
838 P 0974 2 $XABALL_INIT(XAB=XABALL2,
839 P 0975 2 AID=2,
840 P 0976 2 AOP=CBT,
841 P 0977 2 BKZ=3,
842 P 0978 2 DEQ=3,
843 0979 2 NXT=XABKEY0);
844 0980 2
845 P 0981 2 $XABKEY_INIT(XAB=XABKEY0,
846 P 0982 2 KREF=0,
847 P 0983 2 FLG=DUP,
848 P 0984 2 DTP=BN8,
849 P 0985 2 POS=0,
850 P 0986 2 SIZ=8,
851 P 0987 2 DAN=0,
852 P 0988 2 IAN=1,
853 P 0989 2 LAN=1,
854 0990 2 NXT=XABKEY1);
855 0991 2
856 P 0992 2 $XABKEY_INIT(XAB=XABKEY1,
```

```
! Use owner's UIC
! Set file protection
```



```

857 P 0993 2 KREF=1,
858 P 0994 2 FLG=(DUP,CHG,NUL),
859 P 0995 2 DTP=STG,
860 P 0996 2 POSO=$BYTEOFFSET(MFR_T_FILEKEY),
861 P 0997 2 SIZE=MFR_S_FILEKEY,
862 P 0998 2 NUL=0,
863 P 0999 2 DAN=2,
864 P 1000 2 IAN=2,
865 P 1001 2 LAN=2,
866 1002 2 NXT=0);
867 1003 2
868 1004 2 STATUS = $CREATE(FAB=.IFAB);
869 1005 2 IF .STATUS
870 1006 2 AND .UIC NEQ 0
871 1007 2 AND .SND_XABPRO[XAB$L_UIC] NEQ .UIC
872 1008 2 THEN BEGIN
873 1009 2 $CLOSE(FAB=.IFAB);
874 1010 2 STATUS = RMSS$PRV;
875 1011 2 IFAB[FAB$L_STS] = RMSS$PRV;
876 1012 2 IFAB[FAB$L_STV] = SSS$NOPRIV;
877 1013 2 END;
878 1014 2 SND_XABPRO[XAB$L_NXT] = 0; !Disconnect other XAB's
879 1015 2 IF NOT .STATUS
880 1016 2 THEN BEGIN
881 1017 2 IF .STATUS EQL RMSS$FLK
882 1018 2 THEN IF FIND FILE TYPE(.IFAB)
883 1019 2 THEN IF .IFAB[FAB$B_ORG] EQL FAB$C_SEQ
884 1020 2 THEN STATUS = RMSS$SHR; !If sequential, return RMSS$SHR
885 1021 2 RETURN .STATUS
886 1022 2 END;
887 1023 2
888 1024 2 IF NOT .IFAB[$FAB_DEV(DIR)]
889 1025 2 OR .IFAB[$FAB_DEV(FOR)]
890 1026 2 THEN BEGIN
891 1027 2 $CLOSE(FAB=.IFAB);
892 1028 2 STATUS = SSS$IVDEVNAM;
893 1029 2 IFAB[FAB$L_STS] = SSS$IVDEVNAM;
894 1030 2 IFAB[FAB$L_STV] = 0;
895 1031 2 RETURN .STATUS
896 1032 2 END;
897 1033 2
898 P 1034 2 $RAB_INIT(RAB=.IRAB,
899 P 1035 2 FAB=.IFAB,
900 1036 2 RAC=KEY);
901 1037 2
902 P 1038 2 IF_ERR($CONNECT(RAB=.IRAB);,
903 P 1039 2 IFAB[FAB$L_STS] = .IRAB[RAB$L_STS];
904 P 1040 2 IFAB[FAB$L_STV] = .IRAB[RAB$L_STV];
905 1041 2 RETURN .STATUS);
906 1042 2 RETURN TRUE
907 1043 1 END;
```

\$RMS_PTR= SND_XABPRO
 .EXTRN SYSS\$CREATE

				07FC 00000	.ENTRY	MAIL\$CREATE_ISAM, Save R2,R3,R4,R5,R6,R7,-		
				5A 00000000G	00 9E 00002	MOVAB	R8,R9,R10	0879
				59 00000000'	00 9E 00009	MOVAB	SYSSCLOSE, R10	
				5E FF08	CE 9E 00010	MOVAB	SND_XABPRO, R9	
				58 00000000G	00 3C 00015	MOVZWL	-248(SP), SP	0927
				57 00000000G	00 D0 0001C	MOVZWL	MAIL\$Q_MAILDEF, DEFLEN	0928
				07	6C 91 00023	MOVZWL	MAIL\$Q_MAILDEF+4, DEFADR	0929
					10 1F 00026	CMPB	(AP), #7	
					1C AC D5 00028	BLSSU	1\$	
					0B 13 0002B	TSTL	28(AP)	
				50 1C	AC D0 0002D	BEQL	1\$	0931
				58	60 3C 00031	MOVZWL	DEFDESC, R0	
				57 04	A0 D0 00034	MOVZWL	(R0), DEFLEN	0932
				56 04	AC D0 00038	MOVZWL	4(R0), DEFADR	0946
0050	8F		00	6E	00 2C 0003C	MOVZWL	IFAB, R6	
					66 00043	MOVZWL	#0, (SP), #0, #80, (R6)	
				66 5003	8F B0 00044	MOVW	#20483, (R6)	
				16 A6 0F0F	8F B0 00049	MOVW	#3855, 22(R6)	
				18 A6 007E10A2	8F D0 0004F	MOVW	#8261794, 24(R6)	
				1D A6	20 90 00057	MOVW	#32, 29(R6)	
				1F A6	02 90 0005B	MOVW	#2, 31(R6)	
				24 A6	69 9E 0005F	MOVAB	SND_XABPRO, 36(R6)	
				28 A6 0C	AC D0 00063	MOVZWL	INAM, 40(R6)	
				50 10	AC D0 00068	MOVZWL	NAMDESC, R0	
				2C A6 04	A0 D0 0006C	MOVZWL	4(R0), 44(R6)	
				30 A6	57 D0 00071	MOVZWL	DEFADR, 48(R6)	
				34 A6	60 90 00075	MOVZWL	(R0), 52(R6)	
				35 A6	58 90 00079	MOVZWL	DEFLEN, 53(R6)	
				36 A6 0800	8F B0 0007D	MOVW	#2048, 54(R6)	
				05 00000000G	00 F0 00083	INSV	MAIL\$B_DFW, #5, #1, 4(R6)	0948
				01	02 EF 0008D	EXTZV	#2, #1, FLAGS, R0	0949
				01	50 F0 00093	INSV	R0, #1, #1, 7(R6)	
				18 AC	03 E1 00099	BBC	#3, FLAGS, 2\$	0950
				17 A6	20 90 0009E	MOVZWL	#32, 23(R6)	0951
				19 AC	01 E1 000A2	BBC	#1, FLAGS+1, 3\$	0952
				00	01 F0 000A7	INSV	#1, #0, #2, 74(R6)	0953
0058	8F		00	6E	00 2C 000AD	MOVZWL	#0, (SP), #0, #88, \$RMS_PTR	0956
					69 000B4			
				69 5813	8F B0 000B5	MOVW	#22547, \$RMS_PTR	
				04 A9 E0	AD 9E 000BA	MOVAB	XABALL0, \$RMS_PTR+4	
				0C A9 14	AC D0 000BF	MOVZWL	UIC, SND_XABPRO+12	0957
				08 A9	34 AE 000C4	MNEGW	#52, SND_XABPRO+8	0958
				6E	00 2C 000C8	MOVZWL	#0, (SP), #0, #32, \$RMS_PTR	0965
					AD 000CD			
				E0 AD 2014	8F B0 000CF	MOVW	#8212, \$RMS_PTR	
				E4 AD C0	AD 9E 000D5	MOVAB	XABALL1, \$RMS_PTR+4	
				E8 AD	20 90 000DA	MOVZWL	#32, \$RMS_PTR+8	
				F4 AD	0F B0 000DE	MOVW	#15, \$RMS_PTR+20	
				F6 AD	05 90 000E2	MOVZWL	#5, \$RMS_PTR+22	
				6E	00 2C 000E6	MOVZWL	#0, (SP), #0, #32, \$RMS_PTR	0972
					AD 000EB			
				C0 AD 2014	8F B0 000ED	MOVW	#8212, \$RMS_PTR	
				C4 AD A0	AD 9E 000F3	MOVAB	XABALL2, \$RMS_PTR+4	
				C8 AD	20 90 000F8	MOVZWL	#32, \$RMS_PTR+8	
				D4 AD 01050005	8F D0 000FC	MOVZWL	#17104901, \$RMS_PTR+20	
				6E	00 2C 00104	MOVZWL	#0, (SP), #0, #32, \$RMS_PTR	0979

004C	8F	00	A0	AD	2014	AD	00109	MOVW	#8212, \$RMS_PTR	:	
			A4	AD	4C	8F	B0 0010B	MOVAB	XABKEY0, \$RMS_PTR+4	:	
			A8	AD		AE	9E 00111	MOVW	#32, \$RMS_PTR+8	:	
			B4	AD	02030003	20	90 00116	MOVW	#33751043, \$RMS_PTR+20	:	
				6E		8F	D0 0011A	MOVW	#0, (SP), #0, #76, \$RMS_PTR	:	0990
						00	2C 00122			:	
						AE	00129			:	
			4C	AE	4C15	8F	B0 0012B	MOVW	#19477, \$RMS_PTR	:	
			50	AE		6E	9E 00131	MOVAB	XABKEY1, \$RMS_PTR+4	:	
			54	AE	0101	8F	B0 00135	MOVW	#257, \$RMS_PTR+8	:	
			5E	AE	0701	8F	B0 0013B	MOVW	#1793, \$RMS_PTR+18	:	
004C	8F	00	7A	AE		08	90 00141	MOVW	#8, \$RMS_PTR+46	:	
				6E		00	2C 00145	MOVW	#0, (SP), #0, #76, \$RMS_PTR	:	1002
						6E	0014C			:	
				6E	4C15	8F	B0 0014D	MOVW	#19477, \$RMS_PTR	:	
			08	AE	0202	8F	B0 00152	MOVW	#514, \$RMS_PTR+8	:	
			0A	AE		02	90 00158	MOVW	#2, \$RMS_PTR+10	:	
			12	AE		07	B0 0015C	MOVW	#7, \$RMS_PTR+18	:	
			17	AE		01	90 00160	MOVW	#1, \$RMS_PTR+23	:	
			1E	AE		09	B0 00164	MOVW	#9, \$RMS_PTR+30	:	
			2E	AE		27	90 00168	MOVW	#39, \$RMS_PTR+46	:	
						56	DD 0016C	PUSHL	R6	:	1004
		00000000G		00		01	FB 0016E	CALLS	#1, SYSS\$CREATE	:	
				52		50	D0 00175	MOVW	R0, STATUS	:	
				24		52	E9 00178	BLBC	STATUS, 4\$	:	1005
					14	AC	D5 0017B	TSTL	UIC	:	1006
						1F	13 0017E	BEQL	4\$	:	
			14	AC	0C	A9	D1 00180	CMPL	SND_XABPRO+12, UIC	:	1007
						18	13 00185	BEQL	4\$	:	
						56	DD 00187	PUSHL	R6	:	1009
				6A		01	FB 00189	CALLS	#1, SYSS\$CLOSE	:	
				52	0001829A	8F	D0 0018C	MOVW	#98970, STATUS	:	1010
			08	A6	0001829A	8F	D0 00193	MOVW	#98970, 8(R6)	:	1011
			0C	A6		24	D0 0019B	MOVW	#36, 12(R6)	:	1012
					04	A9	D4 0019F	CLRL	SND_XABPRO+4	:	1014
				21		52	E8 001A2	BLBS	STATUS, 5\$	:	1015
		0001828A		8F		52	D1 001A5	CMPL	STATUS, #98954	:	1017
						34	12 001AC	BNEQ	7\$	:	
						56	DD 001AE	PUSHL	R6	:	1018
			FD5E	CF		01	FB 001B0	CALLS	#1, FIND_FILE_TYPE	:	
				2A		50	E9 001B5	BLBC	R0, 7\$	:	
					1D	A6	95 001B8	TSTB	29(R6)	:	1019
						25	12 001BB	BNEQ	7\$	:	
				52	000186B4	8F	D0 001BD	MOVW	#100020, STATUS	:	1020
						1C	11 001C4	BRB	7\$	:	1021
		04		40	A6	03	E1 001C6	BBC	#3, 64(R6), 6\$	:	1024
				17		A6	E9 001CB	BLBC	67(R6), 8\$	:	1025
						56	DD 001CF	PUSHL	R6	:	1027
						01	FB 001D1	CALLS	#1, SYSS\$CLOSE	:	
				6A		8F	3C 001D4	MOVZWL	#324, STATUS	:	1028
			08	A6	0144	8F	3C 001D9	MOVZWL	#324, 8(R6)	:	1029
					0C	A6	D4 001DF	CLRL	12(R6)	:	1030
				50		52	D0 001E2	MOVW	STATUS, R0	:	1031
						04	001E5	RET		:	
				57	08	AC	D0 001E6	MOVW	IRAB, R7	:	1036
0044	8F	00		6E		00	2C 001EA	MOVW	#0, (SP), #0, #68, (R7)	:	
						67	001F1			:	

	67	4401	8F	B0	001F2	MOVW	#17409, (R7)	:	
1E	A7		01	90	001F7	MOVB	#1, 30(R7)	:	
3C	A7		56	D0	001FB	MOVL	R6, 60(R7)	:	
			57	DD	001FF	PUSHL	R7	:	1041
00000000G	00		01	FB	00201	CALLS	#1, SYSSCONNECT	:	
	06		50	E8	00208	BLBS	STATUS, 9\$	:	
08	A6	08	A7	7D	0020B	MOVQ	8(R7), 8(R6)	:	
				04	00210	RET		:	
	50		01	D0	00211 9\$:	MOVL	#1, R0	:	1042
			04	00214		RET		:	1043

; Routine Size: 533 bytes, Routine Base: \$CODE\$ + 070C

```
909 1044 1 GLOBAL ROUTINE MAIL$UPDATE_I_MSG =
910 1045 1 ++
911 1046 1 FUNCTIONAL DESCRIPTION:
912 1047 1
913 1048 1 Update the header record of the current message
914 1049 1
915 1050 1 INPUTS:
916 1051 1
917 1052 1 MAIL$SL_CURMSG points at current msg
918 1053 1 --
919 1054 2 BEGIN
920 1055 2
921 1056 2 BIND
922 1057 2 MMH = MAIL$SL_CURMSG[MMB_T_HDREC] + $BYTEOFFSET(MFR_T_DATA) : $BBLOCK;
923 1058 2
924 1059 2 MAIL$T_RD_RAB[RAB$SL_RFA0] = .MAIL$SL_CURMSG[MMB_L_RFA0];
925 1060 2 MAIL$T_RD_RAB[RAB$W_RFA4] = .MAIL$SL_CURMSG[MMB_W_RFA4];
926 1061 2 MAIL$T_RD_RAB[RAB$B_RAC] = RAB$C_RFA;
927 1062 2 MAIL$T_RD_RAB[RAB$SL_RBF] = MAIL$SL_CURMSG[MMB_T_HDREC];
928 1063 2 MAIL$T_RD_RAB[RAB$W_RSZ] = .MAIL$SL_CURMSG[MMB_Q_HDRECLN];
929 1064 2 MAIL$T_RD_RAB[RAB$SL_RBP] = 0;
930 1065 2 MAIL$T_RD_RAB[RAB$SL_CTX] = MAIL$ WRITEERR;
931 P 1066 2 IF_ERR($FIND(RAB=MAIL$T_RD_RAB,ERR=UTIL$REPORT_IO_ERROR));
932 1067 2 RETURN .STATUS;
933 1068 2
934 P 1069 2 IF_ERR($UPDATE(RAB=MAIL$T_RD_RAB,ERR=UTIL$REPORT_IO_ERROR));
935 1070 2 RETURN .STATUS;
936 1071 2
937 1072 2
938 1073 2 Update flags in module header
939 1074 2
940 1075 2 MAIL$SL_CURMSG[MMB_V_NEWMMSG] = .MMH[MMH_V_NEWMMSG];
941 1076 2 MAIL$SL_CURMSG[MMB_V_REPLIED] = .MMH[MMH_V_REPLIED];
942 1077 2 RETURN TRUE
943 1078 1 END;
```

				.EXTRN	SYSS\$UPDATE	
			003C 00000	.ENTRY	MAIL\$UPDATE_I_MSG, Save R2,R3,R4,R5	: 1044
	55	00000000'	00 9E 00002	MOVAB	MAIL\$SL_CURMSG, R5	:
	54	00000000G	00 9E 00009	MOVAB	UTIL\$REPORT_IO_ERROR, R4	:
	53	00000000G	00 9E 00010	MOVAB	MAIL\$T_RD_RAB, R3	:
	50		65 D0 00017	MOVL	MAIL\$SL_CURMSG, R0	: 1057
	52	77	A0 9E 0001A	MOVAB	119(R0), R2	:
10	A3		60 D0 0001E	MOVL	(R0), MAIL\$T_RD_RAB+16	: 1059
14	A3	04	A0 B0 00022	MOVW	4(R0), MAIL\$T_RD_RAB+20	: 1060
1E	A3		02 90 00027	MOVW	#2, MAIL\$T_RD_RAB+30	: 1061
28	A3	47	A0 9E 0002B	MOVAB	71(R0), MAIL\$T_RD_RAB+40	: 1062
22	A3	11	A0 B0 00030	MOVW	17(R0), MAIL\$T_RD_RAB+34	: 1063
		04	A3 D4 00035	CLRL	MAIL\$T_RD_RAB+4	: 1064
18	A3	007E10D0	8F D0 00038	MOVL	#8261840, MAIL\$T_RD_RAB+24	: 1065
			18 BB 00040	PUSHR	#^M<R3,R4>	: 1067
00000000G	00		02 FB 00042	CALLS	#2, SYSS\$FIND	:
	23		50 E9 00049	BLBC	STATUS, 1\$	:
			18 BB 0004C	PUSHR	#^M<R3,R4>	: 1070

		00000000G	00	02	FB	0004E	CALLS	#2, SYSSUPDATE	:	
			17	50	E9	00055	BLBC	STATUS, 1\$	:	
OA	A0		50	65	D0	00058	MOVL	MAIL\$ CURMSG, R0	:	1075
		01	00	62	F0	0005B	INSV	(R2), #0, #1, 10(R0)	:	
OA	A0	62	01	01	EF	00061	EXTZV	#1, #1, (R2), R1	:	1076
		01	01	51	F0	00066	INSV	R1, #1, #1, 10(R0)	:	
			50	01	D0	0006C	MOVL	#1, R0	:	1077
					04	0006F	RET		:	1078

; Routine Size: 112 bytes, Routine Base: \$CODE\$ + 0921


```

945 1079 1 GLOBAL ROUTINE MAIL$READ_INFO_REC(MAILRAB,INDEX,LOCKFLAG,RECDISC,RECRFA) =
946 1080 1 ++
947 1081 1 FUNCTIONAL DESCRIPTION:
948 1082 1
949 1083 1 Read the specified info record
950 1084 1
951 1085 1 INPUTS:
952 1086 1
953 1087 1 MAILRAB = address of rab connected
954 1088 1 INDEX = info record index
955 1089 1 LOCKFLAG = 1 to lock record, 0 to not
956 1090 1 RECDISC = addr of 2-lw vector to return record descriptor in
957 1091 1 RECRFA = (optional) addr of 2-lw vector to return RFA of record
958 1092 1
959 1093 1 --
960 1094 2 BEGIN
961 1095 2
962 1096 2 MAP
963 1097 2 MAILRAB : REF $BBLOCK,
964 1098 2 RECDISC : REF VECTOR[,LONG],
965 1099 2 RECRFA : REF VECTOR[,LONG];
966 1100 2
967 1101 2 BUILTIN
968 1102 2 NULLPARAMETER;
969 1103 2
970 1104 2 LOCAL
971 1105 2 STATUS,
972 1106 2 COUNTER,
973 1107 2 KEYBUFFER : VECTOR[8,BYTE];
974 1108 2
975 1109 2 LITERAL
976 1110 2 RETRY_RLK = 10, ! number of retries for a locked record
977 1111 2 SLEEP_RLK = 75; ! ms to sleep before retrying
978 1112 2
979 1113 2 BIND
980 1114 2
981 1115 2 Define the delta time to sleep before retrying to GET a locked record.
982 1116 2
983 1117 2 WAKEDELTA = UPLIT LONG(-10*1000*SLEEP_RLK,-1),
984 1118 2 LWBUF = KEYBUFFER : VECTOR[,LONG];
985 1119 2
986 1120 2 IF NOT NULLPARAMETER(5)
987 1121 3 THEN BEGIN
988 1122 3 RECRFA[0] = 0;
989 1123 3 RECRFA[1] = 0;
990 1124 2 END;
991 1125 2
992 1126 2 LWBUF[0] = INDEX;
993 1127 2 LWBUF[1] = 0;
994 1128 2
995 1129 2 MAILRAB[RAB$L_RDP] = RAB$M_LOC;
996 1130 2 MAILRAB[RAB$B_KRF] = 0;
997 1131 2 MAILRAB[RAB$B_KSZ] = 8;
998 1132 2 MAILRAB[RAB$B_RAC] = RAB$C_KEY;
999 1133 2 MAILRAB[RAB$L_KBF] = KEYBUFFER;
1000 1134 2 IF NOT .LOCKFLAG
1001 1135 2 THEN MAILRAB[RAB$V_NLK] = TRUE;
```

```
1002 1136 2
1003 1137 2 IF .LOCKFLAG
1004 1138 3 THEN BEGIN
1005 1139 3   COUNTER = RETRY_RLK;
1006 1140 4   WHILE ((STATUS = $GE((RAB=.MAILRAB)) EQL RMSS_RLK)
1007 1141 4     AND ((COUNTER = .COUNTER = 1) GEQ 0)
1008 1142 3     DO IF $SCHDWK(DAYTIM = WAKEDELTA) THEN $HIBER;
1009 1143 3   IF NOT .STATUS
1010 1144 3     THEN RETURN .STATUS;
1011 1145 3   END
1012 1146 3 ELSE IF NOT (STATUS = $GET(RAB=.MAILRAB))
1013 1147 2   THEN RETURN .STATUS;
1014 1148 2
1015 1149 2 RECDISC[0] = .MAILRAB[RAB$W_RSZ];
1016 1150 2 RECDISC[1] = .MAILRAB[RAB$L_RBF];
1017 1151 2
1018 1152 2 IF NOT NULLPARAMETER(5)
1019 1153 3 THEN BEGIN
1020 1154 3   RECRFA[0] = .MAILRAB[RAB$L_RFA0];
1021 1155 3   RECRFA[1] = .MAILRAB[RAB$W_RFA4];
1022 1156 2   END;
1023 1157 2 RETURN TRUE
1024 1158 1 END;
```

FFFFFFFF FFF48E50 00991 .BLKB 3
00994 P.AAK: .LONG -750000, -1

WAKEDELTA= P.AAK
.EXTRN SYSSCHDWK, SYSSHIBER

55	00000000G	00	9E	00002	.ENTRY	MAIL\$READ INFO_REC, Save R2,R3,R4,R5	: 1079
5E		08	C2	00009	MOVAB	SYSSGET, R5	
05		6C	91	0000C	SUBL2	#8, SP	
		0B	1F	0000F	CMPL	(AP), #5	: 1120
	14	AC	D5	00011	BLSSU	1\$	
		06	13	00014	TSTL	20(AP)	
50	14	AC	D0	00016	BEQL	1\$	
		60	7C	0001A	MOVL	RECRFA, R0	: 1122
6E	08	AC	D0	0001C	CLRQ	(R0)	
	04	AE	D4	00020	MOVL	INDEX, LWBUF	: 1126
52	04	AC	D0	00023	CLRL	LWBUF+4	: 1127
04	A2	8F	D0	00027	MOVL	MAILRAB, R2	: 1129
34	A2	08	B0	0002F	MOVL	#65536, 4(R2)	
1E	A2	01	90	00033	MOVW	#8, 52(R2)	: 1131
30	A2	6E	9E	00037	MOVB	#1, 30(R2)	: 1132
	08	AC	E8	0003B	MOVAB	KEYBUFFER, 48(R2)	: 1133
06	A2	10	88	0003F	BLBS	LOCKFLAG, 2\$	: 1134
32	0C	AC	E9	00043	BISB2	#16, 6(R2)	: 1135
54		0A	D0	00047	BLBC	LOCKFLAG, 4\$	: 1137
		52	DD	0004A	MOVL	#10, COUNTER	: 1139
65		01	FB	0004C	PUSHL	R2	: 1140
53		50	D0	0004F	CALLS	#1, SYSSGET	
000182AA	8F	53	D1	00052	MOVL	R0, STATUS	
		26	12	00059	CMPL	STATUS, #98986	
					BNEQ	5\$	

			54	D7	0005B	DECL	COUNTER	:	1141
			22	19	0005D	BLSS	5\$	:	
			7E	D4	0005F	CLRL	-(SP)	:	1142
		94	AF	9F	00061	PUSHAB	WAKEDELTA	:	
			7E	7C	00064	CLRQ	-(SP)	:	
00000000G	00		04	FB	00066	CALLS	#4, SYSS\$CHDWK	:	
	DA		50	E9	0006D	BLBC	R0, 3\$	:	
00000000G	00		00	FB	00070	CALLS	#0, SYSS\$HIBER	:	
			D1	11	00077	BRB	3\$	:	
			52	DD	00079	PUSHL	R2	:	1146
	65		01	FB	0007B	CALLS	#1, SYSS\$GET	:	
	53		50	D0	0007E	MOVL	R0, STATUS	:	
	04		53	E8	00081	BLBS	STATUS, 6\$	:	
	50		53	D0	00084	MOVL	STATUS, R0	:	1147
				04	00087	RET		:	
	50	10	AC	D0	00088	MOVL	RECDESC, R0	:	1149
	60	22	A2	3C	0008C	MOVZWL	34(R2), (R0)	:	
04	A0	28	A2	D0	00090	MOVL	40(R2), 4(R0)	:	1150
	05		6C	91	00095	CMPB	(AP), #5	:	1152
			12	1F	00098	BLSSU	7\$	:	
		14	AC	D5	0009A	TSTL	20(AP)	:	
			0D	13	0009D	BEQL	7\$	:	
	50	14	AC	D0	0009F	MOVL	RECRFA, R0	:	1154
	60	10	A2	D0	000A3	MOVL	16(R2), (R0)	:	
04	A0	14	A2	3C	000A7	MOVZWL	20(R2), 4(R0)	:	1155
	50		01	D0	000AC	MOVL	#1, R0	:	1157
			04	000AF	RET			:	1158

; Routine Size: 176 bytes, Routine Base: \$CODE\$ + 099C

```
: 1026      1159 1 GLOBAL ROUTINE MAIL$UPDATE_INFO_REC(MAILRAB,RECDISC,RECRFA) =
: 1027      1160 1  **
: 1028      1161 1  FUNCTIONAL DESCRIPTION:
: 1029      1162 1  :
: 1030      1163 1  :       Update an info record (or write it if non-existent)
: 1031      1164 1  :
: 1032      1165 1  INPUTS:
: 1033      1166 1  :
: 1034      1167 1  :       MAILRAB is addr of RAB
: 1035      1168 1  :       RECDISC is addr of descriptor of record
: 1036      1169 1  :       RECRFA (optional) is addr of 2-lw containing RFA of record
: 1037      1170 1  :
: 1038      1171 2 BEGIN
: 1039      1172 2
: 1040      1173 2 MAP
: 1041      1174 2     MAILRAB : REF $BBLOCK,
: 1042      1175 2     RECDISC : REF $BBLOCK,
: 1043      1176 2     RECRFA : REF VECTOR[,LONG];
: 1044      1177 2
: 1045      1178 2 BUILTIN
: 1046      1179 2     NULLPARAMETER;
: 1047      1180 2
: 1048      1181 2 LOCAL
: 1049      1182 2     STATUS;
: 1050      1183 2
: 1051      1184 2 BIND
: 1052      1185 2     RECORD_INDEX = .RECDISC[DSC$A_POINTER] : LONG,
: 1053      1186 2     INDEX_DATIM = .RECDISC[DSC$A_POINTER] : VECTOR[,BYTE];
: 1054      1187 2
: 1055      1188 2 MAILRAB[RAB$L_RDP] = 0;
: 1056      1189 2 MAILRAB[RAB$W_RSZ] = .RECDISC[DSC$W_LENGTH];
: 1057      1190 2 MAILRAB[RAB$L_RBF] = .RECDISC[DSC$A_POINTER];
: 1058      1191 2 MAILRAB[RAB$L_CTX] = MAIL$WRITEERR;
: 1059      1192 2 MAILRAB[RAB$L_KBF] = INDEX_DATIM;
: 1060      1193 2 MAILRAB[RAB$B_KSZ] = 8;
: 1061      1194 2 MAILRAB[RAB$B_KRF] = 0;
: 1062      1195 2
: 1063      1196 2 MAILRAB[RAB$L_RFA0] = 0;
: 1064      1197 2
: 1065      1198 2 IF NOT NULLPARAMETER(3)
: 1066      1199 2     AND .RECRFA[0] NEQ 0
: 1067      1200 3 THEN BEGIN
: 1068      1201 3     MAILRAB[RAB$L_RFA0] = .RECRFA[0];
: 1069      1202 3     MAILRAB[RAB$W_RFA4] = .RECRFA[1];
: 1070      1203 2     END;
: 1071      1204 2
: 1072      1205 2 IF .MAILRAB[RAB$L_RFA0] NEQ 0
: 1073      1206 3 THEN BEGIN
: 1074      1207 3     MAILRAB[RAB$B_RAC] = RAB$C_RFA;
: 1075      1208 3     IF_ERR($FIND(RAB=.MAILRAB,ERR=UTIL$REPORT_IO_ERROR);,
: 1076      1209 3         RETURN .STATUS);
: 1077      1210 3     IF_ERR($UPDATE(RAB=.MAILRAB,ERR=UTIL$REPORT_IO_ERROR);,
: 1078      1211 3         RETURN .STATUS);
: 1079      1212 3     END
: 1080      1213 3 ELSE BEGIN
: 1081      1214 3     MAILRAB[RAB$B_RAC] = RAB$C_KEY;
: 1082      1215 3     STATUS = $FIND(RAB=.MAILRAB);
```

.....

[illegible]

MAIL\$ISAM_SUBS ISAM MAIL MESSAGE SUBROUTINES
V04-000

C 7
16-Sep-1984 00:54:01
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VM\$MASTER:[MAIL.SRC]ISUBS.B32;1 Page 42
(13)

04 00090 6\$: RET

; 1227

; Routine Size: 145 bytes, Routine Base: \$CODE\$ + 0A4C

MAI
V04

52

: 1237

: 1228

				00000920	52	04	AC	D0	00002	MOVL	SIGARG, R2	:	1244
					8F	04	A2	D1	00006	CMPL	4(R2), #2336	:	1246
							04	12	0000E	BNEQ	1\$	:	
					50		01	D0	00010	MOVL	#1, R0	:	1247
								04	00013	RET		:	
					0C	04	A2	D1	00014	1\$: CMPL	4(R2), #12	:	1249
							06	12	00018	BNEQ	2\$	:	
					50	0918	8F	3C	0001A	MOVZWL	#2328, R0	:	1250
								04	0001F	RET		:	
06	A2		0C	00	0000007E		8F	F0	00020	2\$: INSV	#126, #0, #12, 6(R2)	:	1252
	04	04	A2	03			00	ED	0002A	CMPZV	#0, #3, 4(R2), #4	:	1253
							06	12	00030	BNEQ	3\$	:	
04	A2		03	00			02	F0	00032	INSV	#2, #0, #3, 4(R2)	:	1254
				62			02	C2	00038	3\$: SUBL2	#2, (R2)	:	1256
							7E	7C	0003B	CLRQ	-(SP)	:	1258
						B1	AF	9F	0003D	PUSHAB	PUT_OUT	:	
							52	DD	00040	PUSHL	R2	:	
				00000000G	00		04	FB	00042	CALLS	#4, SYSSPUTMSG	:	
					62		02	C0	00049	ADDL2	#2, (R2)	:	1259
							7E	7C	0004C	CLRQ	-(SP)	:	1261
				00000000G	00		02	FB	0004E	CALLS	#2, SYSSUNWIND	:	
							50	D4	00055	CLRL	R0	:	1262
							04	00057	RET			:	1263

; Routine Size: 88 bytes, Routine Base: \$CODE\$ + 0AEC


```
1133 1264 1 ROUTINE CALL_RECLAIM (STAT_FLAG) =
1134 1265 1 1**
1135 1266 1 FUNCTIONAL DESCRIPTION:
1136 1267 1
1137 1268 1 Call CONV$RECLAIM. If errors, will unwind to caller of this
1138 1269 1 routine.
1139 1270 1 --
1140 1271 2 BEGIN
1141 1272 2
1142 1273 2 OWN
1143 1274 2 RECLAIM_ROUTINE;
1144 1275 2
1145 1276 2 LOCAL
1146 1277 2 STATUS,
1147 1278 2 STATVEC : VECTOR[5, LONG];
1148 1279 2
1149 1280 2 BUILTIN
1150 1281 2 FP;
1151 1282 2
1152 1283 2 CH$FILL(0, 5*4, STATVEC);
1153 1284 2 STATVEC[0] = 4;
1154 1285 2 IF .RECLAIM_ROUTINE EQL 0
P 1286 2 THEN IF _ERR($LIB$FIND_IMAGE_SYMBOL(SD_CONV$HR, $DESCRIPTOR('CONV$RECLAIM'),
P 1287 2 RECLAIM_ROUTINE);,
1157 1288 2 RETURN .STATUS);
1158 1289 2
1159 1290 2 .FP = RECLAIM_HANDLER;
1160 1291 2 STATUS = (.RECLAIM_ROUTINE)(MAIL$Q_RD_DESC, STATVEC, %REF(1));
1161 1292 2 IF .STATUS
1162 1293 2 AND .STAT_FLAG
1163 1294 2 THEN SIGNAL(MAIL$ RECLSTAT, %X'10004',
1164 1295 2 .STATVEC[1], .STATVEC[2], .STATVEC[3], .STATVEC[4]);
1165 1296 2 .FP = 0;
1166 1297 2 RETURN .STATUS
1167 1298 2
1168 1299 1 END;
```

```
4D 49 41 4C 43 45 52 24 56 4E 4F 43 00B44 P.AAM: .ASCII \CONV$RECLAIM\
0000000C 00B50 P.AAL: .LONG 12
00000000' 00B54 .ADDRESS P.AAM
```

```
.PSECT $OWNS, NOEXE, 2
```

```
00090 RECLAIM_ROUTINE:
.BLK 4
```

```
.PSECT $CODE$, NOWRT, 2
```

```
00FC 00000 CALL_RECLAIM:
```

```
57 00000000' 00 9E 00002 .WORD Save R2, R3, R4, R5, R6, R7
56 EC AF 9E 00009 MOVAB RECLAIM_ROUTINE, R7
5E 18 C2 0000D SUBL2 P.AAL, R6
#24, SP
```

```
: 1264
```

14	00	6E	00	2C	00010	MOVCS	#0, (SP), #0, #20, STATVEC	1283
			04	AE	00015			
	04	AE	04	D0	00017	MOVL	#4, STATVEC	1284
			67	D5	0001B	TSTL	RECLAIM_ROUTINE	1285
			12	12	0001D	BNEQ	1\$	
		00C0	8F	BB	0001F	PUSHR	#^M<R6,R7>	1288
		F4C8	C6	9F	00023	PUSHAB	SD_CONVSHR	
00000000G	00		03	FB	00027	CALLS	#3, LIB\$FIND_IMAGE_SYMBOL	
	47		50	E9	0002E	BLBC	STATUS, 3\$	
	6D	FF5F	CF	9E	00031	MOVAB	RECLAIM_HANDLER, (FP)	1290
	50		67	D0	00036	MOVL	RECLAIM_ROUTINE, R0	1291
	6E		01	D0	00039	MOVL	#1, (SPT)	
			5E	DD	0003C	PUSHL	SP	
		08	AE	9F	0003E	PUSHAB	STATVEC	
		00000000G	00	9F	00041	PUSHAB	MAIL\$Q_RD_DESC	
	60		03	FB	00047	CALLS	#3, (R0)	
	52		50	D0	0004A	MOVL	R0, STATUS	
	23		52	E9	0004D	BLBC	STATUS, 2\$	1292
	1F	04	AC	E9	00050	BLBC	STAT_FLAG, 2\$	1293
		14	AE	DD	00054	PUSHL	STATVEC+16	1295
		14	AE	DD	00057	PUSHL	STATVEC+12	
		14	AE	DD	0005A	PUSHL	STATVEC+8	
		14	AE	DD	0005D	PUSHL	STATVEC+4	
		00010004	8F	DD	00060	PUSHL	#65540	1294
		007EBE9B	8F	DD	00066	PUSHL	#8306331	
00000000G	00		06	FB	0006C	CALLS	#6, LIB\$SIGNAL	
			6D	D4	00073	CLRL	(FP)	1296
	50		52	D0	00075	MOVL	STATUS, R0	1297
			04	00078	3\$:	RET		1299

; Routine Size: 121 bytes, Routine Base: \$CODE\$ + 0B58


```
1170 1300 1 GLOBAL ROUTINE MAIL$RECLAIM(STAT_FLAG) =
1171 1301 1 ++
1172 1302 1 FUNCTIONAL DESCRIPTION:
1173 1303 1
1174 1304 1 Close the current ISAM file, call CONV$RECLAIM, and then re-open
1175 1305 1 the file
1176 1306 1
1177 1307 1 INPUTS:
1178 1308 1
1179 1309 1 STAT_FLAG = 1 to print stats, 0 to not
1180 1310 1
1181 1311 1 --
1182 1312 2 BEGIN
1183 1313 2
1184 1314 2 LOCAL
1185 1315 2 RECDISC : VECTOR[2, LONG],
1186 1316 2 RECPTR : REF $BLOCK,
1187 1317 2 RECRFA : VECTOR[2, LONG],
1188 1318 2 STATUS;
1189 1319 2
1190 1320 2 MAIL$DISABLE_CTRLY();
1191 1321 2 $CLOSE(FAB=MAIL$RD_RAB[RAB$FAB]);
1192 1322 2 MAIL$GL_FLAGS[MAIF_V_FILOPN] = 0;
1193 1323 2
1194 1324 2 STATUS = CALL RECLAIM(.STAT_FLAG);
1195 1325 2 MAIL$OPEN_FILE(MAIL$Q_RD_DESC);
1196 1326 2
1197 1327 2 Zero the count of deleted messages in the fileinfo record
1198 1328 2
1199 1329 2 IF .STATUS
1200 1330 2 THEN IF MAIL$READ_INFO_REC(MAIL$RD_RAB, MFR_C_FILEINFO, 1, RECDISC, RECRFA)
1201 1331 3 THEN BEGIN
1202 1332 3 RECPTR = RECDISC[1];
1203 1333 3 (RECPTR[MFR_T_DATA]) < 0, 32, 0 > = 0;
1204 1334 3 MAIL$DELMSGCNT = 0;
1205 1335 3 MAIL$UPDATE_INFO_REC(MAIL$RD_RAB, RECDISC, RECRFA);
1206 1336 2 END;
1207 1337 2
1208 1338 2 MAIL$ENABLE_CTRLY();
1209 1339 2 RETURN TRUE
1210 1340 1 END;
```

```
001C 00000
54 82 AF 9E 00002
53 00000000G 00 9E 00006
5E 10 C2 0000D
00000000G 00 00 FB 00010
3C A3 DD 00017
00000000G 00 01 FB 0001A
00000000G 00 08 8A 00021
04 AC DD 00028
64 01 FB 0002B
52 50 D0 0002E
```

```
.ENTRY MAIL$RECLAIM, Save R2,R3,R4
MOVAB CALL RECLAIM, R4
MOVAB MAIL$RD_RAB, R3
SUBL2 #16, SP
CALLS #0, MAIL$DISABLE_CTRLY
PUSHL MAIL$RD_RAB+60
CALLS #1, SY$SCLOSE
BICB2 #8, MAIL$GL_FLAGS+1
PUSHL STAT_FLAG
CALLS #1, CALL RECLAIM
MOVL R0, STATUS
```

```
1300
1320
1321
1322
1324
```

00000000G	00	00000000G	00	9F 00031	PUSHAB	MAIL\$Q RD DESC	:	1325
	2C		01	FB 00037	CALLS	#1, MAIL\$OPEN_FILE	:	
			52	E9 0003E	BLBC	STATUS, 1\$	:	1329
			5E	DD 00041	PUSHL	SP	:	1330
		0C	AE	9F 00043	PUSHAB	RECDESC	:	
			01	DD 00046	PUSHL	#1	:	
			03	DD 00048	PUSHL	#3	:	
			53	DD 0004A	PUSHL	R3	:	
FE44	C4		05	FB 0004C	CALLS	#5, MAIL\$READ_INFO_REC	:	
	19		50	E9 00051	BLBC	R0, 1\$	:	
	50		AE	D0 00054	MOVL	RECDESC+4, RECPTR	:	1332
		0C	A0	D4 00058	CLRL	48(RECPTR)	:	1333
		30	00	D4 0005B	CLRL	MAIL\$L_DELMMSGCNT	:	1334
		00000000'	5E	DD 00061	PUSHL	SP	:	1335
			AE	9F 00063	PUSHAB	RECDESC	:	
		0C	53	DD 00066	PUSHL	R3	:	
FEF4	C4		03	FB 00068	CALLS	#3, MAIL\$UPDATE_INFO_REC	:	
00000000G	00		00	FB 0006D 1\$:	CALLS	#0, MAIL\$ENABLE_CTRLV	:	1338
	50		01	D0 00074	MOVL	#1, R0	:	1339
			04	00077	RET		:	1340

; Routine Size: 120 bytes, Routine Base: \$CODE\$ + 0BD1

```
1212 1341 1 ROUTINE CALL_CONV_COMPRESS(FLAGS,INDESC,TMPDESC) =
1213 1342 1 **
1214 1343 1 FUNCTIONAL DESCRIPTION:
1215 1344 1
1216 1345 1 Call convert to copy the input file to SYS$LOGIN:CONVRTMAI.TMP
1217 1346 1 This file will then be renamed
1218 1347 1
1219 1348 1 Inputs:
1220 1349 1
1221 1350 1 INDESC = Address of descriptor of input file
1222 1351 1 TMPDESC = address of descriptor of temporary output file
1223 1352 1 --
1224 1353 2 BEGIN
1225 1354 2 MAP
1226 1355 2 FLAGS : BITVECTOR;
1227 1356 2
1228 1357 2 OWN
1229 1358 2 CONV_PASS_FILES,
1230 1359 2 CONV_PASS_OPTIONS,
1231 1360 2 CONV_CONVERT;
1232 1361 2
1233 1362 2 LOCAL
1234 1363 2 STATUS,
1235 1364 2 OPTION_ARRAY : VECTOR[7, LONG],
1236 1365 2 STATVEC : VECTOR[5, LONG];
1237 1366 2
1238 1367 2 BUILTIN
1239 1368 2 FP;
1240 1369 2
1241 1370 2 CH$FILL(0, 5*4, STATVEC);
1242 1371 2 STATVEC[0] = 4;
1243 1372 2 CH$FILL(0, 7*4, OPTION_ARRAY);
1244 1373 2
1245 1374 2 If this is first time, then get addresses of CONVERT routines.
1246 1375 2
1247 1376 2 IF .CONV_PASS_FILES EQL 0
1248 1377 3 THEN BEGIN
1249 P 1378 3 IF_ERR(LIB$FIND_IMAGE_SYMBOL(SD_CONVSHR, $DESCRIPTOR('CONV$PASS_FILES'),
1250 P 1379 3 CONV_PASS_FILES);,
1251 1380 3 RETURN .STATUS);
1252 P 1381 3 IF_ERR(LIB$FIND_IMAGE_SYMBOL(SD_CONVSHR, $DESCRIPTOR('CONV$PASS_OPTIONS'),
1253 P 1382 3 CONV_PASS_OPTIONS);,
1254 1383 3 RETURN .STATUS);
1255 P 1384 3 IF_ERR(LIB$FIND_IMAGE_SYMBOL(SD_CONVSHR, $DESCRIPTOR('CONV$CONVERT'),
1256 P 1385 3 CONV_CONVERT);,
1257 1386 3 RETURN .STATUS);
1258 1387 3 END;
1259 1388 2
1260 1389 2 .FP = RECLAIM_HANDLER;
1261 1390 3 IF NOT (STATUS = (.CONV_PASS_FILES)(.INDESC,.TMPDESC,SD_NL,SD_NL,%REF(1)))
1262 1391 2 THEN RETURN .STATUS;
1263 1392 2 OPTION_ARRAY[0] = 6; !# of options
1264 1393 2 OPTION_ARRAY[3] = 1; !Fast load
1265 1394 2 OPTION_ARRAY[6] = 0; !No sort
1266 1395 3 IF NOT (STATUS = (.CONV_PASS_OPTIONS)(OPTION_ARRAY,%REF(1)))
1267 1396 2 THEN RETURN .STATUS;
1268 1397 2 STATUS = (.CONV_CONVERT)(STATVEC,%REF(1));
```

```
: 1269      1398 2 :FP = 0;
: 1270      1399 2 IF .STATUS
: 1271      1400 2 THEN SIGNAL(MAIL$_COPIEDR,3,.INDESC,.TMPDESC,.STATVEC[2]);
: 1272      1401 2 RETURN .STATUS
: 1273      1402 2
: 1274      1403 1 END;
```

```
53 45 4C 49 46 5F 53 53 41 50 24 56 4E 4F 43 00C49 P.AAO: .ASCII \CONVPASS_FILES\
                                0000000F 00C58 P.AAN: .LONG 15
                                00000000' 00C5C P.AAQ: .ADDRESS P.AAO
4F 49 54 50 4F 5F 53 53 41 50 24 56 4E 4F 43 00C60 P.AAQ: .ASCII \CONVPASS_OPTIONS\
                                53 4E 00C6F
                                00C71 .BLKB 3
                                00000011 00C74 P.AAP: .LONG 17
                                00000000' 00C78 P.AAP: .ADDRESS P.AAP
                                54 52 45 56 4E 4F 43 24 56 4E 4F 43 00C7C P.AAS: .ASCII \CONV$CONVERT\
                                0000000C 00C88 P.AAR: .LONG 12
                                00000000' 00C8C P.AAR: .ADDRESS P.AAS
```

.PSECT \$OWNS,NOEXE,2

00094 CONV_PASS_FILES:

.BLKB 4

00098 CONV_PASS_OPTIONS:

.BLKB 4

0009C CONV_CONVERT:

.BLKB 4

.PSECT \$CODE\$,NOWRT,2

01FC 00000 CALL_CONV COMPRESS:

```
58 00000000G 00 9E 00002 .WORD Save R2,R3,R4,R5,R6,R7,R8 : 1341
57 00000000' 00 9E 00009 MOVAB LIB$FIND_IMAGE_SYMBOL, R8
56 F374 CF 9E 00010 MOVAB CONV_PASS_FILES, R7
5E 34 C2 00015 SUBL2 #52, SP
14 00 6E 00 2C 00018 MOVCS #0, (SP), #0, #20, STATVEC : 1370
04 AE 0001D
6E 04 D0 0001F MOVL #4, STATVEC : 1371
1C 00 04 00 2C 00023 MOVCS #0, (SP), #0, #28, OPTION_ARRAY : 1372
18 AE 00028
67 D5 0002A TSTL CONV_PASS_FILES : 1376
2C 12 0002C BNEQ 2$
57 DD 0002E PUSHL R7 : 1380
0C40 C6 9F 00030 PUSHAB P.AAN
56 DD 00034 PUSHL R6
68 03 FB 00036 CALLS #3, LIB$FIND_IMAGE_SYMBOL
1B 50 E9 00039 BLBC STATUS, 1$
04 A7 9F 0003C PUSHAB CONV_PASS_OPTIONS : 1383
0C5C C6 9F 0003F PUSHAB P.AAP
56 DD 00043 PUSHL R6
68 03 FB 00045 CALLS #3, LIB$FIND_IMAGE_SYMBOL
0C 50 E9 00048 BLBC STATUS, 1$
```

	08	A7	9F	0004B	PUSHAB	CONV CONVERT	1386
	0C70	C6	9F	0004E	PUSHAB	P.AAR	
		56	DD	00052	PUSHL	R6	
68		03	FB	00054	CALLS	#3, LIB\$FIND_IMAGE_SYMBOL	
70		50	E9	00057	BLBC	STATUS, 4\$	
6D	FDFE	CF	9E	0005A	MOVAB	RECLAIM_HANDLER, (FP)	1389
50		67	D0	0005F	MOVL	CONV_PASS_FILES, R0	1390
6E		01	D0	00062	MOVL	#1, (SP)	
		5E	DD	00065	PUSHL	SP	
	1C	A6	9F	00067	PUSHAB	SD_NL	
	1C	A6	9F	0006A	PUSHAB	SD_NL	
7E	08	AC	7D	0006D	MOVQ	INDESC, -(SP)	
60		05	FB	00071	CALLS	#5, (R0)	
52		50	D0	00074	MOVL	R0, STATUS	
4D		52	E9	00077	BLBC	STATUS, 3\$	
18	AE	06	D0	0007A	MOVL	#6, OPTION_ARRAY	1392
24	AE	01	D0	0007E	MOVL	#1, OPTION_ARRAY+12	1393
		AE	D4	00082	CLRL	OPTION_ARRAY+24	1394
50	30	A7	D0	00085	MOVL	CONV_PASS_OPTIONS, R0	1395
6E	04	01	D0	00089	MOVL	#1, (SP)	
		5E	DD	0008C	PUSHL	SP	
	1C	AE	9F	0008E	PUSHAB	OPTION_ARRAY	
60		02	FB	00091	CALLS	#2, (R0)	
52		50	D0	00094	MOVL	R0, STATUS	
2D		52	E9	00097	BLBC	STATUS, 3\$	
50	08	A7	D0	0009A	MOVL	CONV_CONVERT, R0	1397
6E		01	D0	0009E	MOVL	#1, (SP)	
		5E	DD	000A1	PUSHL	SP	
	08	AE	9F	000A3	PUSHAB	STATVEC	
60		02	FB	000A6	CALLS	#2, (R0)	
52		50	D0	000A9	MOVL	R0, STATUS	
		6D	D4	000AC	CLRL	(FP)	1398
16		52	E9	000AE	BLBC	STATUS, 3\$	1399
	0C	AE	DD	000B1	PUSHL	STATVEC+8	1400
7E	08	AC	7D	000B4	MOVQ	INDESC, -(SP)	
		03	DD	000B8	PUSHL	#3	
	007E1069	8F	DD	000BA	PUSHL	#8261737	
00000000G	00	05	FB	000C0	CALLS	#5, LIB\$SIGNAL	
	50	52	D0	000C7	MOVL	STATUS, R0	1401
		04	000CA	4\$:	RET		1403

; Routine Size: 203 bytes, Routine Base: \$CODE\$ + 0C90

```
: 1276      1404 1 GLOBAL ROUTINE MAIL$COMPRESS(FLAGS,OUTFILE,FDLFILE) =
: 1277      1405 1 +-
: 1278      1406 1 FUNCTIONAL DESCRIPTION:
: 1279      1407 1
: 1280      1408 1     Compress the currently open file by use of CONVERT
: 1281      1409 1
: 1282      1410 1 Inputs:
: 1283      1411 1
: 1284      1412 1     Flags = flags.  None currently defined
: 1285      1413 1     outfile = address of descriptor of output file.  If not specified
: 1286      1414 1     output file will be same as input file.
: 1287      1415 1     fdlfile = address of descriptor for fdl file.  If not specified,
: 1288      1416 1     MAIL's default fdl description will be used. (NYI)
: 1289      1417 1
: 1290      1418 1 --
: 1291      1419 2 BEGIN
: 1292      1420 2 MAP
: 1293      1421 2     FLAGS : BITVECTOR,
: 1294      1422 2     OUTFILE : REF $BBLOCK,
: 1295      1423 2     FDLFILE : REF $BBLOCK;
: 1296      1424 2
: 1297      1425 2 LOCAL
: 1298      1426 2     STATUS,
: 1299      1427 2     FILENAMEBUF : $BBLOCK[NAM$C_MAXRSS],
: 1300      1428 2     FILENAMEDESC : VECTOR[2,LONG],
: 1301      1429 2     DESC : VECTOR[2,LONG],
: 1302      1430 2     OPEN_DESC : VECTOR[2,LONG],
: 1303      1431 2     NAMRSS : $BBLOCK[NAM$C_MAXRSS],
: 1304      1432 2     NAMESS : $BBLOCK[NAM$C_MAXRSS],
: 1305      1433 2     OPENRSS : $BBLOCK[NAM$C_MAXRSS],
: 1306      1434 2     OPENESS : $BBLOCK[NAM$C_MAXRSS],
: 1307      1435 2     PROTBUF : $BBLOCK[NAM$C_MAXRSS],
: 1308      1436 2     OPEN_FAB : $FAB_DECL,
: 1309      1437 2     OPEN_NAM : $NAM_DECL,
: 1310      1438 2     NEW_NAME : DYN_DESC_DECL,
: 1311      1439 2     PROTFAB : $FAB_DECL,
: 1312      1440 2     PROTXAB : $XABPRO_DECL,
: 1313      1441 2     PROTNAM : $NAM_DECL,
: 1314      1442 2     TFAB : $FAB_DECL,
: 1315      1443 2     TRAB : $RAB_DECL,
: 1316      1444 2     TNAM : $NAM_DECL;
: 1317      1445 2
: 1318      1446 2
: 1319      1447 2 STR$COPY_DX(DYN_DESC,MAIL$Q_RD_DESC);
: 1320      1448 2 MAIL$CLOSE_FILE();
: 1321      1449 2 FILENAMEDESC[0] = NAM$C_MAXRSS;
: 1322      1450 2 FILENAMEDESC[1] = FILENAMEBUF;
: 1323      1451 2 SYSS$FAO($DESCRIPTOR('MAIL !4XL COMPRESS.TMP'),FILENAMEDESC,
: 1324      1452 2     FILENAMEDESC,MAIL$L_MYPID);
: 1325      1453 2
: 1326      1454 2     Always create using MAIL's internal description.  /fdl not implemented.
: 1327      1455 2
: 1328      1456 2 IF .FDLFILE[DESC$W_LENGTH] EQL 0
: 1329      1457 2 THEN BEGIN
: 1330      1458 2     $NAM_INIT(NAM=TNAM,
: 1331      1459 2     RSS=NAM$C_MAXRSS,
: 1332      1460 2     RSA=NAMRSS,
```



```

: 1333 P 1461 2      ESS=NAME$C MAXRSS,
: 1334 1462 2      ESA=NAME$S);
: 1335 1463 2      !
: 1336 1464 2      ! Use input file as default to get created in same device/dir
: 1337 1465 2      STATUS = MAIL$CREATE_ISAM(TFAB,TRAB,TNAM,
: 1338 1466 2      FILENAME$DESC,0,MCR,M_NIL,
: 1339 1467 3      (IF .OUTFILE[DSC$W_LENGTH] EQL 0
: 1340 1468 3      THEN DYN_DESC
: 1341 1469 2      ELSE .OUTFILE));
: 1342 1470 2      IF NOT .STATUS
: 1343 1471 3      THEN BEGIN
: 1344 1472 4          UTIL$REPORT_IO_ERROR((IF NOT .TFAB[FAB$S_STS]
: 1345 1473 4          THEN TFAB
: 1346 1474 3          ELSE TRAB));
: 1347 1475 3          TFAB[FAB$V_DLT] = 1;
: 1348 1476 3          $CLOSE(FAB=TFAB);
: 1349 1477 3          RETURN .STATUS
: 1350 1478 2      END;
: 1351 1479 2      $CLOSE(FAB=TFAB);
: 1352 1480 2      !
: 1353 1481 2      CH$MOVE(8,UTIL$GETFILENAME(TFAB),FILENAME$DESC);
: 1354 1482 2      CH$MOVE(FILENAME$DESC[0],FILENAME$DESC[1],FILENAME$BUF);
: 1355 1483 2      FILENAME$DESC[1] = FILENAME$BUF;
: 1356 1484 2      SIGNAL(MAIL$ CREATED,1,FILENAME$DESC);
: 1357 1485 2      STATUS = CALC_CONV_COMPRESS(.FLAGS,DYN_DESC,FILENAME$DESC);
: 1358 1486 2      IF NOT .STATUS
: 1359 1487 2      THEN RETURN .STATUS;
: 1360 1488 2      !
: 1361 1489 2      ! Now rename the input file to .OLD, and the temp file to the input file
: 1362 1490 2      !
: 1363 1491 2      NEW_NAME[DSC$B_DTYPE] = DSC$K_DTYPE_T;
: 1364 1492 2      NEW_NAME[DSC$B_CLASS] = DSC$K_CLASS_D;
: 1365 1493 2      IF .OUTFILE[DSC$W_LENGTH] EQL 0
: 1366 P 1494 2      THEN IF_ERR(LIB$RENAME FILE(DYN_DESC,$DESCRIPTOR('.OLD'),0,0,0,
: 1367 P 1495 2      MAIL$RENAME_SUCCESS,MAIL$RENAME_ERROR,
: 1368 P 1496 2      0,0,0,NEW_NAME);,
: 1369 1497 2      RETURN .STATUS);
: 1370 1498 2      !
: 1371 1499 2      ! Set the protection of the .OLD file so the the owner can delete it
: 1372 1500 2      !
: 1373 P 1501 2      $FAB_INIT(FAB=PROTFAB,
: 1374 P 1502 2      FNS=.NEW_NAME[DSC$W_LENGTH],
: 1375 P 1503 2      FNA=.NEW_NAME[DSC$A_POINTER],
: 1376 P 1504 2      FAC=(GET-PUT),
: 1377 P 1505 2      CTX=MAIL$ FILNOTDEL,
: 1378 P 1506 2      NAM=PROTNAM,
: 1379 1507 2      XAB=PROTXAB);
: 1380 1508 2      !
: 1381 P 1509 2      $NAM_INIT(NAM=PROTNAM,
: 1382 P 1510 2      ESS=NAME$C MAXRSS,
: 1383 P 1511 2      ESA=PROTBOF,
: 1384 P 1512 2      RSS=NAME$C MAXRSS,
: 1385 1513 2      RSA=PROTBOF);
: 1386 1514 2      !
: 1387 P 1515 2      $XABPRO_INIT(XAB=PROTXAB,
: 1388 1516 2      NXT=0);
: 1389 1517 2
```

```
1390 1518 2 $OPEN(FAB=PROTFAB);
1391 1519 2 PROTXAB[XAB$W_PRO] = DELETE_PRO;
1392 1520 2 $CLOSE(FAB=PROTFAB);
1393 1521 2
1394 1522 2
1395 1523 2 ! Do a parse to ensure that the output file has the correct file type
1396 1524 2
1397 1525 2 DESC[0] = .DYN_DESC[DSC$W_LENGTH];
1398 1526 2 DESC[1] = .DYN_DESC[DSC$A_POINTER];
1399 1527 2 IF .OUTFILE[DSC$W_LENGTH] NEQ 0
1400 1528 3 THEN BEGIN
1401 1529 3     DESC[0] = .OUTFILE[DSC$W_LENGTH];
1402 1530 3     DESC[1] = .OUTFILE[DSC$A_POINTER];
1403 1531 2 END;
1404 P 1532 2 $FAB_INIT(FAB=TFAB,
1405 P 1533 2     FNS=.DESC[0],
1406 P 1534 2     FNA=.DESC[1],
1407 P 1535 2     NAM=TNAM,
1408 P 1536 2     DNS=4,
1409 1537 2     DNA=UPLIT('.MAI'));
1410 P 1538 2 $NAM_INIT(NAM=TNAM,
1411 P 1539 2     RSS=NAM$C_MAXRSS,
1412 P 1540 2     RSA=NAMRSS,
1413 P 1541 2     ESS=NAM$C_MAXRSS,
1414 1542 2     ESA=NAMESS);
1415 1543 2 $PARSE(FAB=TFAB);
1416 1544 2 $SEARCH(FAB=TFAB);
1417 1545 2 CH$MOVE(8,UTIL$GETFILENAME(TFAB),DESC);
1418 1546 2 DESC[0] = .DESC[0] - .TNAM[NAM$B_VER]; !No specific version
1419 P 1547 2 IF_ERR(LIB$RENAME_FILE(FILENAMEDESC,DESC,0,0,0,
1420 P 1548 2     MAIL$RENAME_SUCCESS,MAIL$RENAME_ERROR,0,DYN_DESC);,
1421 1549 2     RETURN .STATUS);
1422 1550 2
1423 1551 2 ! Open the newly created file
1424 1552 2
1425 P 1553 2 $FAB_INIT(FAB=OPEN_FAB,
1426 P 1554 2     FNS=.DYN_DESC[DSC$W_LENGTH],
1427 P 1555 2     FNA=.DYN_DESC[DSC$A_POINTER],
1428 P 1556 2     NAM=OPEN_NAM,
1429 P 1557 2     DNS=4,
1430 1558 2     DNA=UPLIT('.MAI'));
1431 P 1559 2 $NAM_INIT(NAM=OPEN_NAM,
1432 P 1560 2     RSS=NAM$C_MAXRSS,
1433 P 1561 2     RSA=OPENRSS,
1434 P 1562 2     ESS=NAM$C_MAXRSS,
1435 1563 2     ESA=OPENESS);
1436 1564 2 $PARSE(FAB=OPEN_FAB);
1437 1565 2 $SEARCH(FAB=OPEN_FAB);
1438 1566 2 OPEN_DESC[0] = .DYN_DESC[DSC$W_LENGTH] - .OPEN_NAM[NAM$B_VER]; ! Strip version
1439 1567 2 OPEN_DESC[1] = .DYN_DESC[DSC$A_POINTER];
1440 1568 2 MAIL$OPEN_FILE(OPEN_DESC);
1441 1569 2
1442 1570 2 RETURN TRUE
1443 1571 1 END;
```


				OFFC	00000	.ENTRY	MAIL\$COMPRESS, Save R2,R3,R4,R5,R6,R7,R8,- R9,R10,R11	1404		
		5B	00000000G	00	9E	00002	MOVAB	UTIL\$GETFILENAME, R11		
		5A	00000000G	00	9E	00009	MOVAB	SYSS\$CLOSE, R10		
		59	D1	AF	9E	00010	MOVAB	P.AAT, R9		
		58	00000000'	00	9E	00014	MOVAB	DYN_DESC, R8		
		5E	F734	CE	9E	0001B	MOVAB	-2252(SP), SP		
	01FC	CE	020E0000	8F	D0	00020	MOVL	#34471936, NEW_NAME	1438	
			0200	CE	D4	00029	CLRL	NEW_NAME+4		
			00000000G	00	9F	0002D	PUSHAB	MAIL\$Q_RD_DESC	1447	
				58	DD	00033	PUSHL	R8		
	00000000G	00		02	FB	00035	CALLS	#2, STR\$COPY DX		
	00000000G	00		00	FB	0003C	CALLS	#0, MAIL\$CLOSE_FILE	1448	
	FEF8	CD	FF	8F	9A	00043	MOVZBL	#255, FILENAMEDESC	1449	
	FEFC	CD	FF00	CD	9E	00049	MOVAB	FILENAMEBUF, FILENAMEDESC+4	1450	
			00000000G	00	DD	00050	PUSHL	MAIL\$L_MYPID	1452	
			FEF8	CD	9F	00056	PUSHAB	FILENAMEDESC	1451	
			FEF8	CD	9F	0005A	PUSHAB	FILENAMEDESC		
				59	DD	0005E	PUSHL	R9		
0060	8F		00000000G	00	04	FB	00060	CALLS	#4, SYSS\$FAO	
		00		6E	00	2C	00067	MOVCS	#0, (SP), #0, #96, \$RMS_PTR	1462
				6E			0006E			
		6E	6002	8F	B0	0006F	MOVW	#24578, \$RMS_PTR		
	02	AE		01	8E	00074	MNEGB	#1, \$RMS_PTR+2		
	04	AE	FDE8	CD	9E	00078	MOVAB	NAMRSS, \$RMS_PTR+4		
	0A	AE		01	8E	0007E	MNEGB	#1, \$RMS_PTR+10		
	OC	AE	FCE8	CD	9E	00082	MOVAB	NAMESS, \$RMS_PTR+12		
		57	08	AC	D0	00088	MOVL	OUTFILE, R7	1467	
				67	B5	0008C	TSTW	(R7)		
				07	12	0008E	BNEQ	1\$		
		50		68	9E	00090	MOVAB	DYN_DESC, R0		
				50	DD	00093	PUSHL	R0		
				02	11	00095	BRB	2\$		
				57	DD	00097	PUSHL	R7	1469	
				08	DD	00099	PUSHL	#8	1465	
				7E	D4	0009B	CLRL	-(SP)		
			FEF8	CD	9F	0009D	PUSHAB	FILENAMEDESC		
			10	AE	9F	000A1	PUSHAB	TNAM		
			74	AE	9F	000A4	PUSHAB	TRAB		
			00BC	CE	9F	000A7	PUSHAB	TFAB		
	F8CC	CF		07	FB	000AB	CALLS	#7, MAIL\$CREATE_ISAM		
		56		50	D0	000B0	MOVL	R0, STATUS		
		28		56	E8	000B3	BLBS	STATUS, 5\$	1470	
		07	00AC	CE	E8	000B6	BLBS	TFAB+8, 3\$	1472	

			50	00A4	CE	9E	0008B	MOVAB	TFAB, R0		
					04	11	000C0	BRB	4\$		
			50	60	AE	9E	000C2	3\$: MOVAB	TRAB, R0		
					50	DD	000C6	4\$: PUSHL	R0		
		00000000G	00		01	FB	000C8	CALLS	#1, UTIL\$REPORT_IO_ERROR		
		00A9	CE	80	8F	88	000CF	BISB2	#128, TFAB+5	1475	
				00A4	CE	9F	000D5	PUSHAB	TFAB	1476	
			6A		01	FB	000D9	CALLS	#1, SYS\$CLOSE		
					4C	11	000DC	BRB	6\$	1477	
				00A4	CE	9F	000DE	5\$: PUSHAB	TFAB	1479	
			6A		01	FB	000E2	CALLS	#1, SYS\$CLOSE		
				00A4	CE	9F	000E5	PUSHAB	TFAB	1481	
			6B		01	FB	000E9	CALLS	#1, UTIL\$GETFILENAME		
FEF8	CD		60		08	28	000EC	MOVC3	#8, (R0), FILENAMEDESC		
FF00	CD	FEFC	DD	FEF8	CD	28	000F2	MOVC3	FILENAMEDESC, @FILENAMEDESC+4, FILENAMEBUF	1482	
		FEFC	CD	FF00	CD	9E	000FC	MOVAB	FILENAMEBUF, FILENAMEDESC+4	1483	
				FEF8	CD	9F	00103	PUSHAB	FILENAMEDESC	1484	
					01	DD	00107	PUSHL	#1		
		00000000G	00	007E1071	8F	DD	00109	PUSHL	#8261745		
				FEF8	CD	9F	00116	CALLS	#3, LIB\$SIGNAL		
					58	DD	0011A	PUSHAB	FILENAMEDESC	1485	
				04	AC	DD	0011C	PUSHL	R8		
	FDDC	CF			03	FB	0011F	PUSHL	FLAGS		
		56			50	DD	00124	CALLS	#3, CALL_CONV_COMPRESS		
		04			56	E8	00127	MOVL	R0, STATUS		
		50			56	DD	0012A	BLBS	STATUS, 7\$	1486	
						04	0012D	MOVL	STATUS, R0	1487	
		01FE	CE	020E	8F	B0	0012E	RET			
					67	B5	00135	7\$: MOVW	#526, NEW_NAME+2	1491	
					28	12	00137	TSTW	(R7)	1493	
				01FC	CE	9F	00139	BNEQ	8\$		
					7E	7C	0013D	PUSHAB	NEW_NAME	1497	
					7E	D4	0013F	CLRQ	-(SP)		
		00000000G			00	9F	00141	CLRL	-(SP)		
		00000000G			00	9F	00147	PUSHAB	MAIL\$RENAME_ERROR		
					7E	7C	0014D	PUSHAB	MAIL\$RENAME_SUCCESS		
					7E	D4	0014F	CLRQ	-(SP)		
				0C	A9	9F	00151	CLRL	-(SP)		
					58	DD	00154	PUSHAB	P.AAV		
		00000000G	00		08	FB	00156	PUSHL	R8		
			01		50	E8	0015D	CALLS	#11, LIB\$RENAME_FILE		
						04	00160	BLBS	STATUS, 8\$		
0050	8F		00	6E	00	2C	00161	RET			
				01AC	CE		00168	8\$: MOVC5	#0, (SP), #0, #80, \$RMS_PTR	1507	
				5003	8F	B0	0016B	MOVW	#20483, \$RMS_PTR		
		01AC	CE		03	90	00172	MOVW	#3, \$RMS_PTR+22		
		01C2	CE	007E11F2	8F	DD	00177	MOVL	#8262130, \$RMS_PTR+24		
		01C4	CE		02	90	00180	MOVW	#2, \$RMS_PTR+3T		
		01CB	CE		CE	9E	00185	MOVAB	PROTXAB, \$RMS_PTR+36		
		01D0	CE	0154	CE	9E	0018C	MOVAB	PROTNAM, \$RMS_PTR+40		
		01D4	CE	00F4	CE	DD	00193	MOVL	NEW_NAME+4, \$RMS_PTR+44		
		01D8	CE	0200	CE	90	0019A	MOVW	NEW_NAME, \$RMS_PTR+52		
0060	8F		00	6E	00	2C	001A1	MOVC5	#0, (SP), #0, #96, \$RMS_PTR	1513	
					CE		001A8				
		00F4	CE	00F4	8F	B0	001AB	MOVW	#24578, \$RMS_PTR		
		00F6	CE	6002	01	8E	001B2	MNEGB	#1, \$RMS_PTR+2		

0058	8F	00	00F8 CE 02B4 CE 9E 001B7	MOVAB	PROTBUF, \$RMS_PTR+4	
			00FE CE 01 8E 001BE	MNEGB	#1, \$RMS_PTR+T0	
			0100 CE 02B4 CE 9E 001C3	MOVAB	PROTBUF, \$RMS_PTR+12	
			6E 00 2C 001CA	MOVCS	#0, (SP), #0, #88, \$RMS_PTR	1516
			0154 CE 0154 CE 8F B0 001D1	MOVW	#22547, \$RMS_PTR	
			5813 CE 8F B0 001D4	PUSHAB	PROTFAB	1518
			01AC CE 9F 001DB	CALLS	#1, SYSS\$OPEN	
		00000000G	01 FB 001DF	MOVW	#-188, PROTXAB+8	1519
		015C CE FF44 8F B0 001E6		PUSHAB	PROTFAB	1520
			01AC CE 9F 001ED	CALLS	#1, SYSS\$CLOSE	
		6A 68 3C 001F4		MOVZWL	DYN_DESC, DESC	1525
		FEF0 CD 04 AB D0 001F9		MOVL	DYN_DESC+4, DESC+4	1526
		FEF4 CD 67 B5 001FF		TSTW	(R7)	1527
			0B 13 00201	BEQL	9\$	
		FEF0 CD 67 3C 00203		MOVZWL	(R7), DESC	1529
0050	8F	00	FEF4 CD 04 A7 D0 00208	MOVL	4(R7), DESC+4	1530
			6E 00 2C 0020E 9\$:	MOVCS	#0, (SP), #0, #80, \$RMS_PTR	1537
			00A4 CE 00 00215			
			5003 8F B0 00218	MOVW	#20483, \$RMS_PTR	
			00A4 CE 02 90 0021F	MOVB	#2, \$RMS_PTR+22	
			00BA CE 02 90 00224	MOVB	#2, \$RMS_PTR+31	
			00C3 CE 6E 9E 00229	MOVAB	TNAM, \$RMS_PTR+40	
			00CC CE FEF4 CD D0 0022E	MOVL	DESC+4, \$RMS_PTR+44	
			00D0 CE 14 A9 9E 00235	MOVAB	P.AAX, \$RMS_PTR+48	
			00D4 CE FEF0 CD 90 0023B	MOVB	DESC, \$RMS_PTR+52	
			00D8 CE 04 90 00242	MOVB	#4, \$RMS_PTR+53	
0060	8F	00	00D9 CE 00 2C 00247	MOVCS	#0, (SP), #0, #96, \$RMS_PTR	1542
			6E 00 0024E			
			02 6E 6002 8F B0 0024F	MOVW	#24578, \$RMS_PTR	
			04 AE 01 8E 00254	MNEGB	#1, \$RMS_PTR+2	
			0A AE FDE8 CD 9E 00258	MOVAB	NAMRSS, \$RMS_PTR+4	
			0C AE 01 8E 0025E	MNEGB	#1, \$RMS_PTR+10	
			00000000G	MOVAB	NAMESS, \$RMS_PTR+12	
			00000000G	PUSHAB	TFAB	1543
			00A4 CE 01 FB 0026C	CALLS	#1, SYSS\$PARSE	
			00A4 CE 9F 00273	PUSHAB	TFAB	1544
			00A4 CE 01 FB 00277	CALLS	#1, SYSS\$SEARCH	
			00A4 CE 9F 0027E	PUSHAB	TFAB	1545
			6B 01 FB 00282	CALLS	#1, UTIL\$GETFILENAME	
	FEF0	CD	60 08 28 00285	MOVCS	#8, (R0), DESC	
			50 3D AE 9A 0028B	MOVZBL	TNAM+61, R0	1546
			FEF0 CD 50 C2 0028F	SUBL2	R0, DESC	
			00000000G	PUSHL	R8	1549
			00000000G	CLRL	-(SP)	
			00 00 9F 00298	PUSHAB	MAIL\$RENAME_ERROR	
			00 00 9F 0029E	PUSHAB	MAIL\$RENAME_SUCCESS	
			7E 7C 002A4	CLRQ	-(SP)	
			7E D4 002A6	CLRL	-(SP)	
			FEF0 CD 9F 002A8	PUSHAB	DESC	
			FEF8 CD 9F 002AC	PUSHAB	FILENAMEDESC	
		00000000G	09 FB 002B0	CALLS	#9, LIB\$RENAME_FILE	
		01 50 E8 002B7		BLBS	STATUS, 10\$	
			04 002BA	RET		
0050	8F	00	6E 00 2C 002BB 10\$:	MOVCS	#0, (SP), #0, #80, \$RMS_PTR	1558
			0264 CE 00 002C2			
			5003 8F B0 002C5	MOVW	#20483, \$RMS_PTR	

0060	8F	00	027A	CE	02	90	002CC	MOVB	#2, SRMS_PTR+22	:		
			0283	CE	02	90	002D1	MOVB	#2, SRMS_PTR+31	:		
			028C	CE	0204	CE	9E	002D6	MOVAB	OPEN_NAM, SRMS_PTR+40	:	
			0290	CE	04	A8	D0	002DD	MOVL	DYN_DESC+4, SRMS_PTR+44	:	
			0294	CE	18	A9	9E	002E3	MOVAB	P.AAY, SRMS_PTR+28	:	
			0298	CE		68	90	002E9	MOVB	DYN_DESC, SRMS_PTR+52	:	
			0299	CE		04	90	002EE	MOVB	#4, SRMS_PTR+53	:	
				6E		00	2C	002F3	MOVCS	#0, (SP); #0, #96, SRMS_PTR	:	1563
					0204	CE		002FA		:		
			0204	CE	6002	8F	B0	002FD	MOVW	#24578, SRMS_PTR	:	
			0206	CE		01	8E	00304	MNEGB	#1, SRMS_PTR+2	:	
			0208	CE	FBE8	CD	9E	00309	MOVAB	OPENRSS, SRMS_PTR+4	:	
			020E	CE		01	8E	00310	MNEGB	#1, SRMS_PTR+10	:	
			0210	CE	03B4	CE	9E	00315	MOVAB	OPENESS, SRMS_PTR+12	:	
					0264	CE	9F	0031C	PUSHAB	OPEN_FAB	:	1564
		00000000G	00			01	FB	00320	CALLS	#1, SYSSPARSE	:	
					0264	CE	9F	00327	PUSHAB	OPEN_FAB	:	1565
		00000000G	00			01	FB	0032B	CALLS	#1, SYSSSEARCH	:	
			50			68	3C	00332	MOVZWL	DYN_DESC, R0	:	1566
			51		0241	CE	9A	00335	MOVZBL	OPEN_NAM+61, R1	:	
			50			51	C3	0033A	SUBL3	R1, R0, OPEN_DESC	:	
	FEE8	CD			04	A8	D0	00340	MOVL	DYN_DESC+4, OPEN_DESC+4	:	1567
		FEEC	CD		FEE8	CD	9F	00346	PUSHAB	OPEN_DESC	:	1568
						01	FB	0034A	CALLS	#1, MAIL\$OPEN_FILE	:	
		00000000G	00			01	D0	00351	MOVL	#1, R0	:	1570
			50			04	00354	RET		:	1571	

; Routine Size: 853 bytes, Routine Base: \$CODE\$ + 0090

```
1445 1572 1 ROUTINE DECODE_MISCDATA(MSGDESC) =
1446 1573 1 **
1447 1574 1 FUNCTIONAL DESCRIPTION:
1448 1575 1
1449 1576 1 Decode the miscellaneous data and fill in fields in MMB
1450 1577 1
1451 1578 1 INPUTS:
1452 1579 1
1453 1580 1 MSGDESC is addr of MMB for msg
1454 1581 1 --
1455 1582 2 BEGIN
1456 1583 2
1457 1584 2 MAP
1458 1585 2 MSGDESC : REF $BBLOCK;
1459 1586 2
1460 1587 2 BIND
1461 1588 2 HDREC = MSGDESC[MMB_T_HDREC] : $BBLOCK,
1462 1589 2 MMH = HDREC[MMH_T_DATA] : $BBLOCK;
1463 1590 2
1464 1591 2 LOCAL
1465 1592 2 RECEND,
1466 1593 2 PTR : REF VECTOR[WORD];
1467 1594 2
1468 1595 2 CH$FILL(0,4*DSC$C_S_BLN,MSGDESC[MMB_Q_FROMDESC]); !Zero all descriptors
1469 1596 2 MSGDESC[MMB_L_NRECS] = 0;
1470 1597 2 PTR = MMH[MMH_T_MISCDAT];
1471 1598 2
1472 1599 2 RECEND = HDREC + .MSGDESC[MMB_W_HDRECLN];
1473 1600 2 WHILE .PTR LSSU .RECEND
1474 1601 3 DO BEGIN
1475 1602 3 CASE .PTR[0] FROM MMH_C_FROM TO MMH_C_NREC
1476 1603 3 OF SET
1477 1604 3 [MMH_C_FROM] :
1478 1605 4 BEGIN
1479 1606 4 (MSGDESC[MMB_Q_FROMDESC]) = .PTR[1];
1480 1607 4 MSGDESC[MMB_Q_FROMDESC]+4 = PTR[2];
1481 1608 3 END;
1482 1609 3 [MMH_C_TO] :
1483 1610 4 BEGIN
1484 1611 4 (MSGDESC[MMB_Q_TODESC]) = .PTR[1];
1485 1612 4 MSGDESC[MMB_Q_TODESC]+4 = PTR[2];
1486 1613 3 END;
1487 1614 3 [MMH_C_SUBJ] :
1488 1615 4 BEGIN
1489 1616 4 (MSGDESC[MMB_Q_SUBJDESC]) = .PTR[1];
1490 1617 4 MSGDESC[MMB_Q_SUBJDESC]+4 = PTR[2];
1491 1618 3 END;
1492 1619 3 [MMH_C_NREC] :
1493 1620 4 BEGIN
1494 1621 4 BIND
1495 1622 4 NREC = PTR[2] : LONG;
1496 1623 4
1497 1624 4 MSGDESC[MMB_L_NRECS] = .NREC;
1498 1625 3 END;
1499 1626 3 [INRANGE,
1500 1627 3 OUTRANGE] :
1501 1628 3 1;
```

```
: 1502      1629 3      TES;  
: 1503      1630 3      PTR = PTR[2] + .PTR[1];  
: 1504      1631 2      END;  
: 1505      1632 2  
: 1506      1633 2 RETURN TRUE  
: 1507      1634 1 END;
```

!Skip this entry

00FC 00000 DECODE_MISCDATA:

		56	04	AC	D0	00002	.WORD	Save R2,R3,R4,R5,R6,R7	: 1572
		57	47	A6	9E	00006	MOVL	MSGDESC, R6	: 1588
20	00	6E		00	2C	0000A	MOVAB	71(R6), R7	
			13	A6		0000F	MOVCS	#0, (SP), #0, #32, 19(R6)	: 1595
			33	A6	D4	00011	CLRL	51(R6)	: 1596
		50	40	A7	9E	00014	MOVAB	64(R7), PTR	: 1597
		52	11	A6	3C	00018	MOVZWL	17(R6), RECENT	: 1599
		52		57	C0	0001C	ADDL2	R7, RECENT	
		52		50	D1	0001F	CMPL	PTR, RECENT	: 1600
				46	1E	00022	BGEQU	8\$	
0037	05	00		60	AF	00024	CASEW	(PTR), #0, #5	: 1602
	0026	001A		000E		00028	.WORD	3\$-2\$,-	
		0032		0037		00030		4\$-2\$,-	
								5\$-2\$,-	
								7\$-2\$,-	
								7\$-2\$,-	
								6\$-2\$	
			29	11	00034		BRB	7\$	
13	A6	02	A0	3C	00036	3\$:	MOVZWL	2(PTR), 19(R6)	: 1606
17	A6	04	A0	9E	00038		MOVAB	4(R0), 23(R6)	: 1607
			1D	11	00040		BRB	7\$	: 1602
1B	A6	02	A0	3C	00042	4\$:	MOVZWL	2(PTR), 27(R6)	: 1611
1F	A6	04	A0	9E	00047		MOVAB	4(R0), 31(R6)	: 1612
			11	11	0004C		BRB	7\$	: 1602
23	A6	02	A0	3C	0004E	5\$:	MOVZWL	2(PTR), 35(R6)	: 1616
27	A6	04	A0	9E	00053		MOVAB	4(R0), 39(R6)	: 1617
			05	11	00058		BRB	7\$	: 1602
33	A6	04	A0	D0	0005A	6\$:	MOVL	4(PTR), 51(R6)	: 1624
	51	02	A0	3C	0005F	7\$:	MOVZWL	2(PTR), R1	: 1630
	50	04	A140	9E	00063		MOVAB	4(R1)[PTR], PTR	
			B5	11	00068		BRB	1\$	: 1600
	50		01	D0	0006A	8\$:	MOVL	#1, R0	: 1633
			04	0006D			RET		: 1634

: Routine Size: 110 bytes, Routine Base: \$CODE\$ + 10E5

: 1508 1635 1


```
: 1510      1636 1 GLOBAL ROUTINE MAIL$DESELECT_MSGS =
: 1511      1637 1 |++
: 1512      1638 1 | FUNCTIONAL DESCRIPTION:
: 1513      1639 1 |
: 1514      1640 1 |         Deallocate all msgs in msg table
: 1515      1641 1 | --
: 1516      1642 2 BEGIN
: 1517      1643 2
: 1518      1644 2 IF .MAIL$NMSG GTRU 0
: 1519      1645 2 THEN INCRU I FROM 0 TO .MAIL$NMSG-1
: 1520      1646 3 DO BEGIN
: 1521      1647 3     MAIL$CURMSG = .MAIL$MSGTBL[I];
: 1522      1648 3     IF .MAIL$CURMSG NEQ 0
: 1523      1649 3     THEN DEALLOC_MSG();
: 1524      1650 2     END;
: 1525      1651 2
: 1526      1652 2 STR$FREE1 DX(MAIL$Q_CURFOLDER);
: 1527      1653 2 MAIL$NMSG = 0;
: 1528      1654 2 MAIL$MSG_NUM = 0;
: 1529      1655 2 MAIL$CURMSG = 0;
: 1530      1656 2 MAIL$MSG_STS = MAIL$NOTREADIN;
: 1531      1657 2
: 1532      1658 2 RETURN TRUE
: 1533      1659 1 END;
```

			001C 00000	.ENTRY MAIL\$DESELECT_MSGS, Save R2,R3,R4	: 1636
54	00000000'	00	9E 00002	MOVAB MAIL\$NMSG, R4	: 1644
50		64	D0 00009	MOVL MAIL\$NMSG, R0	: 1645
		1F	13 0000C	BEQL 4\$	: 1647
53	FF	A0	9E 0000E	MOVAB -1(R0), R3	: 1648
		52	D4 00012	CLRL I	: 1649
		12	11 00014	BRB 3\$	: 1652
50	08	A4	D0 00016 1\$:	MOVL MAIL\$MSGTBL, R0	: 1653
04	A4	6042	D0 0001A	MOVL (R0)[I], MAIL\$CURMSG	: 1654
		05	13 0001F	BEQL 2\$	: 1655
F425	CF	00	FB 00021	CALLS #0, DEALLOC_MSG	: 1656
		52	D6 00026 2\$:	INCL I	: 1658
53		52	D1 00028 3\$:	CML I, R3	: 1659
		E9	1B 0002B	BLEQU 1\$	
	EC	A4	9F 0002D 4\$:	PUSHAB MAIL\$Q_CURFOLDER	
00000000G	00	01	FB 00030	CALLS #1, STR\$FREE1 DX	
	00000000G	00	D4 00037	CLRL MAIL\$MSG_NUM	
		64	7C 0003D	CLRL MAIL\$NMSG	
00000000G	00 007E8022	8F	D0 0003F	MOVL #8290338, MAIL\$MSG_STS	
	50	01	D0 0004A	MOVL #1, R0	
		04	0004D	RET	

; Routine Size: 78 bytes, Routine Base: \$CODE\$ + 1153

```
1535 1660 1 GLOBAL ROUTINE MAIL$REWIND_I_MSG =
1536 1661 1 ++
1537 1662 1 FUNCTIONAL DESCRIPTION:
1538 1663 1
1539 1664 1 Rewind the current message in preparation for reading the text of it
1540 1665 1
1541 1666 1 INPUTS:
1542 1667 1
1543 1668 1 MAIL$L_CURMSG points to current msg descriptor
1544 1669 1 --
1545 1670 2 BEGIN
1546 1671 2
1547 1672 2 IF .MAIL$L_CURMSG EQL 0
1548 1673 2 THEN RETURN MAIL$_NOTREADIN;
1549 1674 2
1550 1675 2 IF .MAIL$L_CURMSG[MMB_L_TXTLST] EQL 0
1551 1676 2 THEN BEGIN
1552 1677 3 READ SAVE MSG();
1553 1678 3 MAIL$L_CURMSG[MMB_L_NBLKS] = (.MAIL$L_CURMSG[MMB_L_NBLKS]+511)/512;
1554 1679 2 END;
1555 1680 2
1556 1681 2 MSGTRAN = MMB C FROM;
1557 1682 2 MAIL$L_MSG_STS = TRUE;
1558 1683 2 RETURN TRUE
1559 1684 1 END;
```

				0004 00000	.ENTRY MAIL\$REWIND_I_MSG, Save R2	1660
		52 00000000'	00	9E 00002	MOVAB MAIL\$L_CURMSG, R2	1672
		50	62	D0 00009	MOVL MAIL\$L_CURMSG, R0	1673
			08	12 0000C	BNEQ 1\$	1675
		50 007E8022	8F	D0 0000E	MOVL #8290338, R0	1677
				04 00015	RET	1678
		06	A0	D5 00016 1\$:	TSTL 6(R0)	1681
			1A	12 00019	BNEQ 2\$	1682
	F2EA	CF	00	FB 0001B	CALLS #0, READ_SAVE_MSG	1683
		50	62	D0 00020	MOVL MAIL\$L_CURMSG, R0	1684
	51 3F	A0 000001FF	8F	C1 00023	ADDL3 #511, 63(R0), R1	
3F	A0	51 00000200	8F	C7 0002C	DIVL3 #512, R1, 63(R0)	
	E4	A2	01	90 00035 2\$:	MOVB #1, MSGTRAN	
	00000000G	00	01	D0 00039	MOVL #1, MAIL\$L_MSG_STS	
		50	01	D0 00040	MOVL #1, R0	
			04	00043	RET	

; Routine Size: 68 bytes, Routine Base: \$CODE\$ + 11A1


```
1561 1685 1 GLOBAL ROUTINE MAIL$GET_I_MSG(OUTDESC,MORE_FLAG) =
1562 1686 1 ++
1563 1687 1 FUNCTIONAL DESCRIPTION:
1564 1688 1
1565 1689 1 Get next unpacked text record of message
1566 1690 1
1567 1691 1 OUTPUTS:
1568 1692 1
1569 1693 1 OUTDESC is the address of a two-longword vector filled in with
1570 1694 1 a descriptor of the text. String must be copied if it is needed
1571 1695 1 later.
1572 1696 1
1573 1697 1 MORE_FLAG = true if more text, else false
1574 1698 1 --
1575 1699 2 BEGIN
1576 1700 2 MAP
1577 1701 2 OUTDESC : REF VECTOR[,LONG];
1578 1702 2
1579 1703 2 OWN
1580 1704 2 CURTXT : REF $BLOCK,
1581 1705 2 TXTPTR : REF VECTOR[,WORD],
1582 1706 2 TXTEPTR,
1583 1707 2 DESC : $BLOCK[DSC$C_S_BLN]
1584 1708 2 PRESET ([DSC$B_DTYPE] = DSC$K_DTYPE_T,
1585 1709 2 [DSC$B_CLASS] = DSC$K_CLASS_D);
1586 1710 2
1587 1711 2 IF NOT .MAIL$GL_FLAGS[MAIF_V_ISAM]
1588 1712 3 THEN BEGIN
1589 1713 3 IF .MAIL$MSG_STS EQL 0 !If already at eof, return it
1590 1714 3 THEN RETURN RM$EOF
1591 1715 3 ELSE BEGIN
1592 1716 4 OUTDESC[0] = .MAIL$W_MSG_RSIZ;
1593 1717 4 OUTDESC[1] = .MAIL$MSG_RADR;
1594 1718 4 MAIL$GET_MSG(); !Get next record
1595 1719 4 RETURN TRUE
1596 1720 3 END;
1597 1721 2 END;
1598 1722 2 .MORE_FLAG = 1; !Assume more text
1599 1723 2 MAIL$MSG_STS = 1;
1600 1724 2 CASE .MSGTRAN FROM MMB_C_NONE TO MMB_C_FROMXDT
1601 1725 2 OF SET
1602 1726 2 [MMB_C_NONE] :
1603 1727 3 BEGIN
1604 1728 3 IF .TXTPTR GEQU .TXTEPTR
1605 1729 4 THEN BEGIN
1606 1730 4 IF .CURTXT EQL 0
1607 1731 4 OR (.CURTXT = .CURTXT[TEXT_L_NEXT]) EQL 0
1608 1732 5 THEN BEGIN
1609 1733 5 OUTDESC[0] = 0;
1610 1734 5 .MORE_FLAG = 0;
1611 1735 5 MAIL$MSG_STS = 0;
1612 1736 5 RETURN FALSE;
1613 1737 5 END
1614 1738 5 ELSE BEGIN
1615 1739 5 TXTPTR = CURTXT[TEXT_T_TEXT];
1616 1740 5 TXTEPTR = .TXTPTR + .CURTXT[TEXT_L_TEXTLEN];
1617 1741 4 END;
```

```
: 1618      1742      3      END;
: 1619      1743      3      IF .MAIL$L_CURMSG[MMB_V_EXTNSTD]
: 1620      1744      4      THEN BEGIN
: 1621      1745      4          OUTDESC[0] = .CURTXT[TEXT_L_TEXTLEN];
: 1622      1746      4          OUTDESC[1] = TXTPTR[0];
: 1623      1747      4          TXTPTR = .OUTDESC[0] + .OUTDESC[1];
: 1624      1748      4      END
: 1625      1749      4      ELSE BEGIN
: 1626      1750      4          OUTDESC[0] = .TXTPTR[0];
: 1627      1751      4          OUTDESC[1] = TXTPTR[1];
: 1628      1752      4          TXTPTR = TXTPTR[1] + .TXTPTR[0];
: 1629      1753      4          IF .MAIL$L_CURMSG[MMB_V_EXTMSG]
: 1630      1754      4              THEN TXTPTR = (.TXTPTR + 1) AND NOT 1;
: 1631      1755      3      END;
: 1632      1756      3      :
: 1633      1757      3      If current record strays out of message buffer boundry
: 1634      1758      3      or if the record is too big we must abort the message
: 1635      1759      3      :
: 1636      1760      3      IF (.OUTDESC[1] + .OUTDESC[0] GTRU .TXTENDPTR) OR
: 1637      1761      3          (.OUTDESC[0] GTRU 2048) THEN
: 1638      1762      4      BEGIN
: 1639      1763      4          .MORE_FLAG = 0;
: 1640      1764      4          MAIL$C_MSG_STS = 0;
: 1641      1765      4          SIGNAL(MAIL$ RECTOBIG,1,.OUTDESC[0]);
: 1642      1766      4          RETURN FALSE;
: 1643      1767      3      END;
: 1644      1768      3      IF .OUTDESC[0] GEQU 2
: 1645      1769      4          AND ((.OUTDESC[1] + .OUTDESC[0] - 2) < 0,16)
: 1646      1770      4          EQLU (CR OR LF^8)
: 1647      1771      3          THEN OUTDESC[0] = .OUTDESC[0] - 2;
: 1648      1772      3      IF .TXTPTR GEQU .TXTENDPTR
: 1649      1773      3          AND .CURTXT[TEXT_L_NEXT] EQL 0
: 1650      1774      3          THEN .MORE_FLAG = 0;
: 1651      1775      2      END;
: 1652      1776      2      [MMB_C_FROM,
: 1653      1777      2      MMB_C_FROMXDT] :
: 1654      1778      2      BEGIN
: 1655      1779      3          BIND
: 1656      1780      3              MSGREC = MAIL$L_CURMSG[MMB_T_HDREC] : $BBLOCK;
: 1657      1781      3
: 1658      1782      3      IF .MSGTRAN EQL MMB_C_FROM
: 1659      1783      4      THEN BEGIN
: 1660      1784      4          DATEDESC[0] = 18;
: 1661      1785      4          DATEDESC[1] = DATEBUF;
: 1662      1786      4          $FAO($DESCRIPTOR('!18XD'),DATEDESC,DATEDESC,MSGREC[MFR_Q_DATIM]);
: 1663      1787      4          END
: 1664      1788      4      ELSE BEGIN
: 1665      1789      4          DATEDESC[0] = 0;
: 1666      1790      4          DATEDESC[1] = DATEBUF;
: 1667      1791      4          END;
: 1668      1792      3      STR$CONCAT(DESC,MAIL$GETMSG(MAIL$_ISUB10),
: 1669      1793      3          MAIL$L_CURMSG[MMB_Q_FROMDESC],DATEDESC);
: 1670      1794      3      MSGTRAN = MMB_C_TO;
: 1671      1795      3      OUTDESC[0] = .DESC[DSC$W_LENGTH];
: 1672      1796      3      OUTDESC[1] = .DESC[DSC$A_POINTER];
: 1673      1797      3      END;
: 1674      1798      2
```

```
: 1675      1799  2  [MMB C TO] :  
: 1676      1800  3      BEGIN  
: 1677      1801  3      MSGTRAN = MMB C SUBJ;  
: 1678      1802  3      STR$CONCAT(DESC,MAIL$GETMSG(MAIL$ ISUB20),  
: 1679      1803  3      MAIL$L_CURMSG[MMB_Q_TODESC]);  
: 1680      1804  3      OUTDESC[0] = .DESC[DSC$W_LENGTH];  
: 1681      1805  3      OUTDESC[1] = .DESC[DSC$A_POINTER];  
: 1682      1806  3      END;  
: 1683      1807  2  
: 1684      1808  2  [MMB C SUBJ] :  
: 1685      1809  3      BEGIN  
: 1686      1810  3      MSGTRAN = MMB C BLANK;  
: 1687      1811  3      STR$CONCAT(DESC,MAIL$GETMSG(MAIL$ ISUB30),  
: 1688      1812  3      MAIL$L_CURMSG[MMB_Q_SUBJDESC]);  
: 1689      1813  3      OUTDESC[0] = .DESC[DSC$W_LENGTH];  
: 1690      1814  3      OUTDESC[1] = .DESC[DSC$A_POINTER];  
: 1691      1815  2      END;  
: 1692      1816  2  
: 1693      1817  2  [MMB C BLANK] :  
: 1694      1818  3      BEGIN  
: 1695      1819  3      MSGTRAN = MMB C TEXT;  
: 1696      1820  3      OUTDESC[0] = 0;  
: 1697      1821  3      OUTDESC[1] = .DESC[DSC$A_POINTER];  
: 1698      1822  2      END;  
: 1699      1823  2  
: 1700      1824  2  [MMB C TEXT] :  
: 1701      1825  3      BEGIN  
: 1702      1826  3      CURTXT = .MAIL$L_CURMSG[MMB_L_TXTLST];  
: 1703      1827  3      :  
: 1704      1828  3      Compute the message text boundry  
: 1705      1829  3      :  
: 1706      1830  3      IF .CURTXT NEQ 0  
: 1707      1831  4      THEN BEGIN  
: 1708      1832  4          TXTPTR = CURTXT[TEXT_T_TEXT];  
: 1709      1833  4          TXTENDPTR = .TXTPTR + .CURTXT[TEXT_L_TEXTLEN];  
: 1710      1834  4      END  
: 1711      1835  3      ELSE (TXTPTR = 1;TXTENDPTR = 0);  
: 1712      1836  3      MSGTRAN = MMB C NONE;  
: 1713      1837  3      RETURN MAIL$GET_I_MSG(.OUTDESC,.MORE_FLAG);  
: 1714      1838  2      END;  
: 1715      1839  2      TES;  
: 1716      1840  2  
: 1717      1841  2      RETURN TRUE  
: 1718      1842  2  
: 1719      1843  1      END;
```

```
44 25 38 31 21 20 011E5 P.ABA: .ASCII \ !18XD\ :  
011EB .BLKB 1 :  
00000006 011EC P.AAZ: .LONG 6 :  
00000000 011F0 .ADDRESS P.ABA :  
: :  
: .PSECT $OWNS,NOEXE,2 :  
000A0 CURTXT: .BLKB 4 :  
000A4 TXTPTR: .BLKB 4 :
```

000AB TXTENDPTR:

02	00#	000AC	DESC:	.BLKB	4
	0E	000AE		.BYTE	0[2]
		000B0		.BYTE	14, 2
				.BLKB	4

.PSECT \$CODE\$,NOWRT,2

07FC 00000

.ENTRY MAIL\$GET_I_MSG, Save R2,R3,R4,R5,R6,R7,R8,- : 1685

5A	00000000G	00	9E	00002	MOVAB	STR\$CONCAT, R10	
59	00000000G	00	9E	00009	MOVAB	MAIL\$GETMSG, R9	
58	E5	AF	9E	00010	MOVAB	P.AAZ, R8	
57	00000000G	00	9E	00014	MOVAB	MAIL\$MSG_STS, R7	
56	00000000'	00	9E	0001B	MOVAB	MSGTRAN, R6	
55	00000000'	00	9E	00022	MOVAB	TXTPTR, R5	
29	00000000G	00	E8	00029	BLBS	MAIL\$GL_FLAGS, 2\$	1711
		67	D5	00030	TSTL	MAIL\$MSG_STS	1713
		08	12	00032	BNEQ	1\$	
50	0001827A	8F	D0	00034	MOVL	#98938, R0	1715
			04	0003B	RET		
50	04	AC	D0	0003C	1\$: MOVL	OUTDESC, R0	1716
60	00000000G	00	3C	00040	MOVZWL	MAIL\$W_MSG_RSIZ, (R0)	
04	00000000G	00	D0	00047	MOVL	MAIL\$MSG_RADR, 4(R0)	1717
00000000G	00	00	FB	0004F	CALLS	#0, MAIL\$GET_MSG	1718
		018A	31	00056	BRW	27\$	1719
54	08	AC	D0	00059	2\$: MOVL	MORE_FLAG, R4	1722
64		01	D0	0005D	MOVL	#1, 7(R4)	
67		01	D0	00060	MOVL	#1, MAIL\$MSG_STS	1723
52	04	AC	D0	00063	MOVL	OUTDESC, R2	1745
51		66	9A	00067	MOVZBL	MSGTRAN, R1	1724
00		51	8F	0006A	CASEB	R1, #0, #6	
0122	06				3\$: .WORD	4\$-3\$, -	
0112	00C2	000E		0006E		15\$-3\$, -	
00C2	014D	0141		00076		18\$-3\$, -	
						19\$-3\$, -	
						22\$-3\$, -	
						24\$-3\$, -	
						15\$-3\$	
04	A5	65	D1	0007C	4\$: CMPL	TXTPTR, TXTENDPTR	1728
		23	1F	00080	BLSSU	7\$	
50	FC	A5	D0	00082	MOVL	CURTXT, R0	1730
		06	13	00086	BEQL	5\$	
FC	A5	60	D0	00088	MOVL	(R0), CURTXT	1731
		09	12	0008C	BNEQ	6\$	
		04	B0	0008E	5\$: CLRL	@OUTDESC	1733
		64	D4	00091	CLRL	(R4)	1734
		67	D4	00093	CLRL	MAIL\$MSG_STS	1735
		73	11	00095	BRB	11\$	1736
50	FC	A5	D0	00097	6\$: MOVL	CURTXT, R0	1739
65	0C	A0	9E	0009B	MOVAB	12(R0), TXTPTR	
65	04	A0	C1	0009F	ADDL3	4(R0), TXTPTR, TXTENDPTR	1740
50		65	D0	000A5	7\$: MOVL	TXTPTR, R0	1746
51	1C	A6	D0	000AB	MOVL	MAIL\$CURMSG, R1	1743
13	0A	A1	06	E1	BBC	#6, 10(R1), 8\$	

		53	FC	A5	D0	000B1	MOVL	CURTXT, R3	1745
		62	04	A3	D0	000B5	MOVL	4(R3), (R2)	
65	04	A2		50	D0	000B9	MOVL	R0, 4(R2)	1746
		62	04	A2	C1	000BD	ADDL3	4(R2), (R2), TXTPTR	1747
				1D	11	000C2	BRB	9\$	1743
		53		60	3C	000C4	MOVZWL	(R0), R3	1750
		62		53	D0	000C7	MOVL	R3, (R2)	
	04	A2	02	A0	9E	000CA	MOVAB	2(R0), 4(R2)	1751
		65	02	A340	9E	000CF	MOVAB	2(R3)[R0], TXTPTR	1752
08	0A	A1		03	E1	000D4	BBC	#3, 10(R1), 9\$	1753
50		65		01	C1	000D9	ADDL3	#1, TXTPTR, R0	1754
65		50		01	CB	000DD	BICL3	#1, R0, TXTPTR	
53	04	A2		62	C1	000E1	ADDL3	(R2), 4(R2), R3	1760
	04	A5		53	C1	000E6	CMPL	R3, TXTENDPTR	
				09	1A	000EA	BGTRU	10\$	
00000800		8F		62	D1	000EC	CMPL	(R2), #2048	1761
				18	1B	000F3	BLEQU	12\$	
				64	D4	000F5	CLRL	(R4)	1763
				67	D4	000F7	CLRL	MAIL\$ _MSG_STS	1764
				62	DD	000F9	PUSHL	(R2)	1765
				01	DD	000FB	PUSHL	#1	
				8F	DD	000FD	PUSHL	#8290618	
00000000G	00		007E813A	03	FB	00103	CALLS	#3, LIB\$ SIGNAL	
				00DA	31	0010A	BRW	28\$	1766
				62	D1	0010D	CMPL	(R2), #2	1768
				0B	1F	00110	BLSSU	13\$	
0A0D	8F		FE	A3	B1	00112	CMPL	-2(R3), #2573	1770
				03	12	00118	BNEQ	13\$	
				02	C2	0011A	SUBL2	#2, (R2)	1771
	04	A5		65	D1	0011D	CMPL	TXTPTR, TXTENDPTR	1772
				0A	1F	00121	BLSSU	14\$	
		50	FC	A5	D0	00123	MOVL	CURTXT, R0	1773
				60	D5	00127	TSIL	(R0)	
				02	12	00129	BNEQ	14\$	
				64	D4	0012B	CLRL	(R4)	1774
				00B3	31	0012D	BRW	27\$	1724
50	1C	A6	00000047	8F	C1	00130	ADDL3	#71, MAIL\$ _CURMSG, R0	1781
		01		51	91	00139	CMPL	R1, #1	1783
				1C	12	0013C	BNEQ	16\$	
	E4	A5		12	D0	0013E	MOVL	#18, DATEDESC	1785
	E8	A5		A5	9E	00142	MOVAB	DATEBUF, DATEDESC+4	1786
				50	DD	00147	PUSHL	R0	1787
				E4	A5	9F	PUSHAB	DATEDESC	
				E4	A5	9F	PUSHAB	DATEDESC	
				58	DD	0014F	PUSHL	R8	
00000000G	00			04	FB	00151	CALLS	#4, SYS\$FAO	
				08	11	00158	BRB	17\$	1783
				E4	A5	D4	CLRL	DATEDESC	1790
	E8	A5		C4	A5	9E	MOVAB	DATEBUF, DATEDESC+4	1791
				E4	A5	9F	PUSHAB	DATEDESC	1794
7E	1C	A6		13	C1	00165	ADDL3	#19, MAIL\$ _CURMSG, -(SP)	
			007EDDE8	8F	DD	0016A	PUSHL	#8314344	1793
		69		01	FB	00170	CALLS	#1, MAIL\$ GETMSG	
				50	DD	00173	PUSHL	R0	
				08	A5	9F	PUSHAB	DESC	
		6A		04	FB	00178	CALLS	#4, STR\$CONCAT	1794
		66		02	90	0017B	MOVB	#2, MSGTRAN	1795

		66	29	11	0017E	BRB	21\$	1796		
7E	1C	A6	03	90	00180	18\$:	MOVB	#3, MSGTRAN	1801	
			1B	C1	00183		ADDL3	#27, MAIL\$\$_CURMSG, -(SP)	1803	
		007EDDF0	8F	DD	00188		PUSHL	#8314352	1802	
			0E	11	0018E		BRB	20\$		
		66	04	90	00190	19\$:	MOVB	#4, MSGTRAN	1810	
7E	1C	A6	23	C1	00193		ADDL3	#35, MAIL\$\$_CURMSG, -(SP)	1812	
		007EDDF8	8F	DD	00198		PUSHL	#8314360	1811	
		69	01	FB	0019E	20\$:	CALLS	#1, MAIL\$GETMSG		
			50	DD	001A1		PUSHL	R0		
			08	A5	9F	001A3	PUSHAB	DESC		
		6A	03	FB	001A6		CALLS	#3, STR\$CONCAT	1812	
		62	08	A5	3C	001A9	21\$:	MOVZWL	DESC, (R2)	1813
			05	11	001AD		BRB	23\$	1814	
		66	05	90	001AF	22\$:	MOVB	#5, MSGTRAN	1819	
			62	D4	001B2		CLRL	(R2)	1820	
	04	A2	0C	A5	D0	001B4	23\$:	MOVL	DESC+4, 4(R2)	1821
			28	11	001B9		BRB	27\$	1724	
		50	06	A6	D0	001BB	24\$:	MOVL	MAIL\$\$_CURMSG, R0	1826
	FC	A5	06	A0	D0	001BF		MOVL	6(R0), CURTXT	
		50	FC	A5	D0	001C4		MOVL	CURTXT, R0	1830
			0C	13	001C8		BEQL	25\$		
		65	0C	A0	9E	001CA		MOVAB	12(R0), TXTPTR	1832
04	A5	65	04	A0	C1	001CE		ADDL3	4(R0), TXTPTR, TXTENDPTR	1833
			03	11	001D4		BRB	26\$	1830	
		65	01	7D	001D6	25\$:	MOVQ	#1, TXTPTR	1835	
			66	94	001D9	26\$:	CLRB	MSGTRAN	1836	
			14	BB	001DB		PUSHR	#^M<R2,R4>	1837	
	FE1E	CF	02	FB	001DD		CALLS	#2, MAIL\$GET_I_MSG		
				04	001E2		RET			
		50	01	D0	001E3	27\$:	MOVL	#1, R0	1841	
				04	001E6		RET			
			50	D4	001E7	28\$:	CLRL	R0	1843	
				04	001E9		RET			

; Routine Size: 490 bytes, Routine Base: \$CODE\$ + 11F4


```
1721 1844 1 GLOBAL ROUTINE MAIL$FIND_!_MSG(MSGNUM,NEXT_FLAG,DELMMSG_FLAG) =
1722 1845 1 ++
1723 1846 1 FUNCTIONAL DESCRIPTION:
1724 1847 1
1725 1848 1 Find the specified message. If MSGNUM is 0, the next message is found
1726 1849 1 If NEXT_FLAG = 1 and specified msg is deleted, then find next undeleted
1727 1850 1 message. If NEXT_FLAG = -1 and specified msg is deleted, then find
1728 1851 1 previous undeleted msg
1729 1852 1
1730 1853 1 --
1731 1854 2 BEGIN
1732 1855 2
1733 1856 2 BUILTIN
1734 1857 2 NULLPARAMETER;
1735 1858 2
1736 1859 2 LOCAL
1737 1860 2 DELF,
1738 1861 2 NEXTF,
1739 1862 2 SRCH_MSGNUM;
1740 1863 2
1741 1864 3 NEXTF = (IF NULLPARAMETER(2)
1742 1865 3 THEN 1
1743 1866 3 ELSE .NEXT_FLAG);
1744 1867 3 DELF = (IF NULLPARAMETER(3)
1745 1868 3 THEN 0
1746 1869 3 ELSE .DELMMSG_FLAG);
1747 1870 3 SRCH_MSGNUM = (IF .MSGNUM EQL 0
1748 1871 3 THEN .MAIL$MSG_NUM + 1
1749 1872 3 ELSE MINU(.MSGNUM,.MAIL$NMSG));
1750 1873 2
1751 1874 2 IF .SRCH_MSGNUM GTRU .MAIL$NMSG
1752 1875 2 OR .SRCH_MSGNUM EQL 0
1753 1876 2 THEN BEGIN
1754 1877 3 MAIL$CURMSG = 0;
1755 1878 3 IF .MAIL$NMSG EQL 0
1756 1879 4 THEN RETURN (MAIL$MSG_STS = MAIL$NOMSGS)
1757 1880 4 ELSE (MAIL$MSG_NUM = .MAIL$NMSG+1;
1758 1881 3 RETURN (MAIL$MSG_STS = MAIL$NOMOREMSG));
1759 1882 2 END;
1760 1883 2
1761 1884 2 MAIL$CURMSG = .MAIL$MSGTBL[.SRCH_MSGNUM-1];
1762 1885 2
1763 1886 2 !
1764 1887 2 Special case LAST or 9999 when last message is deleted
1765 1888 2
1766 1889 2 IF .MSGNUM GEQU .MAIL$NMSG
1767 1890 2 AND .MAIL$CURMSG EQL 0
1768 1891 2 AND .NEXTF GTRU 0
1769 1892 2 THEN NEXTF = -1;
1770 1893 2
1771 1894 2 IF .MAIL$CURMSG EQL 0
1772 1895 2 AND NOT .DELF
1773 1896 2 THEN BEGIN
1774 1897 3 IF .MSGNUM NEQ 0
1775 1898 3 AND NOT .NEXTF
1776 1899 3 THEN RETURN (MAIL$MSG_STS = MAIL$DELMMSG);
1777 1900 3 IF .NEXTF GTR 0
```

```
1778 1901 4 THEN BEGIN
1779 1902 4 INCRU I FROM .SRCH_MSGNUM+1 TO .MAIL$ _NMSG
1780 1903 5 DO BEGIN
1781 1904 5 MAIL$ _MSG_NUM = .I;
1782 1905 5 IF .MAIL$ _MSGTBL[.I-1] NEQ 0
1783 1906 6 THEN BEGIN
1784 1907 6 MAIL$ _CURMSG = .MAIL$ _MSGTBL[.I-1];
1785 1908 7 RETURN (MAIL$ _MSG_STS = TRUE)
1786 1909 5 END;
1787 1910 4 END;
1788 1911 4 END
1789 1912 3 ELSE IF .NEXTF LSS 0
1790 1913 4 THEN BEGIN
1791 1914 4 DECRU I FROM .SRCH_MSGNUM-1 TO 1
1792 1915 5 DO BEGIN
1793 1916 5 MAIL$ _MSG_NUM = .I;
1794 1917 5 IF .MAIL$ _MSGTBL[.I-1] NEQ 0
1795 1918 6 THEN BEGIN
1796 1919 6 MAIL$ _CURMSG = .MAIL$ _MSGTBL[.I-1];
1797 1920 7 RETURN (MAIL$ _MSG_STS = TRUE)
1798 1921 5 END;
1799 1922 4 END;
1800 1923 3 END;
1801 1924 4 RETURN (MAIL$ _MSG_STS = MAIL$ _NOMOREMSG)
1802 1925 2 END;
1803 1926 2
1804 1927 2 MAIL$ _MSG_NUM = .SRCH_MSGNUM;
1805 1928 2 MAIL$ _MSG_STS = TRUE;
1806 1929 2 RETURN TRUE
1807 1930 1 END;
```

58	00000000G	00	9E	00002	.ENTRY	MAIL\$FIND I MSG, Save R2,R3,R4,R5,R6,R7,R8	1844
57	00000000G	00	9E	00009	MOVAB	MAIL\$ _MSG_NUM, R8	
56	00000000'	00	9E	00010	MOVAB	MAIL\$ _MSG_STS, R7	
02		6C	91	00017	MOVAB	MAIL\$ _CURMSG, R6	
		05	1F	0001A	CMPB	(AP), #2	1864
	08	AC	D5	0001C	BLSSU	1\$	
		05	12	0001F	TSTL	8(AP)	
50		01	D0	00021	BNEQ	2\$	
		04	11	00024	MOVL	#1, NEXTF	
50	08	AC	D0	00026	BRB	3\$	
03		6C	91	0002A	MOVL	NEXT_FLAG, NEXTF	1866
		05	1F	0002D	CMPB	(AP), #3	1867
	0C	AC	D5	0002F	BLSSU	4\$	
		04	12	00032	TSTL	12(AP)	
		55	D4	00034	BNEQ	5\$	
55	0C	AC	D0	00038	CLRL	DELF	
53	04	AC	D0	0003C	BRB	6\$	
		06	12	00040	MOVL	DELMMSG_FLAG, DELF	1869
51	68	01	C1	00042	MOVL	MSGNUM, R3	1870
		10	11	00046	BNEQ	7\$	
					ADDL3	#1, MAIL\$ _MSG_NUM, SRCH_MSGNUM	1871
					BRB	9\$	

FC	52	53	D0	00048	7\$:	MOVL	R3, R2	1872	
	A6	52	D1	0004B		CMPL	R2, MAIL\$ _NMSG		
		04	1B	0004F		BLEQU	8\$		
	52	FC	A6	D0	00051	MOVL	MAIL\$ _NMSG, R2		
	51		52	D0	00055	8\$:	MOVL	R2, SRCH_MSGNUM	
	54	FC	A6	D0	00058	9\$:	MOVL	MAIL\$ _NMSG, R4	1874
	54		51	D1	0005C		CMPL	SRCH_MSGNUM, R4	
			04	1A	0005F		BGT&U	10\$	
			51	D5	00061		TSTL	SRCH_MSGNUM	1875
			2C	12	00063		BNEQ	13\$	
			66	D4	00065	10\$:	CLRL	MAIL\$ _CURMSG	1877
			54	D5	00067		TSTL	R4	1878
			10	12	00069		BNEQ	11\$	
	67	007E804A	8F	D0	0006B		MOVL	#8290378, MAIL\$ _MSG_STS	1879
	52	007E804A	8F	D0	00072		MOVL	#8290378, R2	1880
			12	11	00079		BRB	12\$	
	68	01	A4	9E	0007B	11\$:	MOVAB	1(R4), MAIL\$ _MSG_NUM	
	67	007E8042	8F	D0	0007F		MOVL	#8290370, MAIL\$ _MSG_STS	1881
	52	007E8042	8F	D0	00086		MOVL	#8290370, R2	
	50		52	D0	0008D	12\$:	MOVL	R2, R0	1880
				04	00090		RET		
	52	04	A6	D0	00091	13\$:	MOVL	MAIL\$ _MSGTBL, R2	1884
	66	FC	A241	D0	00095		MOVL	-4(R2)[SRCH_MSGNUM], MAIL\$ _CURMSG	
	54		53	D1	0009A		CMPL	R3, R4	1889
			0B	1F	0009D		BLSSU	14\$	
			66	D5	0009F		TSTL	MAIL\$ _CURMSG	1890
			07	12	000A1		BNEQ	14\$	
			50	D5	000A3		TSTL	NEXTF	1891
			03	13	000A5		BEQL	14\$	
	50		01	CE	000A7		MNEGL	#1, NEXTF	1892
			66	D5	000AA	14\$:	TSTL	MAIL\$ _CURMSG	1894
			5E	12	000AC		BNEQ	23\$	
	5B		55	E8	000AE		BLBS	DELF, 23\$	1895
			53	D5	000B1		TSTL	R3	1897
			12	13	000B3		BEQL	15\$	
	0F		50	E8	000B5		BLBS	NEXTF, 15\$	1898
	67	007E80B2	8F	D0	000B8		MOVL	#8290482, MAIL\$ _MSG_STS	1899
	50	007E80B2	8F	D0	000BF		MOVL	#8290482, R0	
				04	000C6		RET		
			50	D5	000C7	15\$:	TSTL	NEXTF	1900
			18	15	000C9		BLEQ	18\$	
	50		51	D0	000CB		MOVL	SRCH_MSGNUM, I	1902
			0A	11	000CE		BRB	17\$	
	68		50	D0	000D0	16\$:	MOVL	I, MAIL\$ _MSG_NUM	1904
	53	FC	A240	D0	000D3		MOVL	-4(R2)[I], R3	1905
			1A	12	000D8		BNEQ	20\$	
			50	D6	000DA	17\$:	INCL	I	1902
	54		50	D1	000DC		CMPL	I, R4	
			EF	1B	000DF		BLEQU	16\$	
			1A	11	000E1		BRB	22\$	1900
			18	18	000E3	18\$:	BGEQ	22\$	1912
	50		51	D0	000E5		MOVL	SRCH_MSGNUM, I	1914
			0F	11	000E8		BRB	21\$	
	68		50	D0	000EA	19\$:	MOVL	I, MAIL\$ _MSG_NUM	1916
	53	FC	A240	D0	000ED		MOVL	-4(R2)[I], R3	1917
			05	13	000F2		BEQL	21\$	
	66		53	D0	000F4	20\$:	MOVL	R3, MAIL\$ _CURMSG	1919

MAIL\$ISAM_SUBS ISAM MAIL MESSAGE SUBROUTINES
V04-000

G 9
16-Sep-1984 00:54:01
14-Sep-1984 12:42:20

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

Page 72
(23)

	16	11	000F7		BRB	24\$	:	1920
	50	D7	000F9	21\$:	DECL	I	:	1914
	ED	12	000FB		BNEQ	19\$	:	
67 007E8042	8F	D0	000FD	22\$:	MOVL	#8290370, MAIL\$ _MSG_STS	:	1924
50 007E8042	8F	D0	00104		MOVL	#8290370, R0	:	
		04	0010B		RET		:	
68	51	D0	0010C	23\$:	MOVL	SRCH MSGNUM, MAIL\$ _MSG_NUM	:	1927
67	01	D0	0010F	24\$:	MOVL	#1, MAIL\$ _MSG_STS	:	1928
50	01	D0	00112		MOVL	#1, R0	:	1929
	04	00115			RET		:	1930

; Routine Size: 278 bytes, Routine Base: \$CODE\$ + 13DE

```
1809 1931 1 GLOBAL ROUTINE MAIL$INFO_I_MSG(SENDER,SUBJ,DATE,FLAGS,  
1810 1932 1 FULL_SENDER,SENDER_PATH,TO_DESC) =  
1811 1933 1 ++  
1812 1934 1 FUNCTIONAL DESCRIPTION:  
1813 1935 1  
1814 1936 1 Return info about message for directory listing  
1815 1937 1  
1816 1938 1 OUTPUTS:  
1817 1939 1  
1818 1940 1 SENDER = sender node-name and username  
1819 1941 1 SUBJ = subject string  
1820 1942 1 DATE = date-string  
1821 1943 1 FLAGS = string of flags present (ascii description)  
1822 1944 1 FULL_SENDER = complete sender including personal name  
1823 1945 1 SENDER_PATH = complete sender excluding personal name  
1824 1946 1 TO_DESC = 'to' string  
1825 1947 1  
1826 1948 1 --  
1827 1949 2 BEGIN  
1828 1950 2  
1829 1951 2 BUILTIN  
1830 1952 2 NULLPARAMETER,  
1831 1953 2 ACTUALCOUNT;  
1832 1954 2  
1833 1955 2 BIND  
1834 1956 2 HDREC = MAIL$L_CURMSG[MMB_T_HDREC] : $BBLOCK;  
1835 1957 2  
1836 1958 2 LOCAL  
1837 1959 2 T_SENDER : VECTOR[2, LONG],  
1838 1960 2 PTR,  
1839 1961 2 TRY_SIZE,  
1840 1962 2 TRY_PTR;  
1841 1963 2  
1842 1964 2 IF .MAIL$L_CURMSG EQL 0  
1843 1965 2 THEN RETURN MAIL$_NOTREADIN;  
1844 1966 2  
1845 1967 2 IF NOT NULLPARAMETER(1)  
1846 1968 3 THEN BEGIN  
1847 1969 3 CH$MOVE(DSC$C_S_BLN,MAIL$L_CURMSG[MMB_Q_FROMDESC],T_SENDER);  
1848 1970 3  
1849 1971 3 | Trim sender name to only his nodename (if present) and username  
1850 1972 3 |  
1851 1973 3 TRY_SIZE = .T_SENDER[0];  
1852 1974 3 TRY_PTR = .T_SENDER[1];  
1853 1975 3 WHILE NOT CH$FAIL(PTR = CH$FIND_SUB(.TRY_SIZE,.TRY_PTR,  
1854 1976 3 %CHARCOUNT('::'),UPLIT('::')))  
1855 1977 4 DO BEGIN  
1856 1978 4 T_SENDER[0] = .TRY_SIZE;  
1857 1979 4 T_SENDER[1] = .TRY_PTR;  
1858 1980 4 TRY_SIZE = .TRY_SIZE - (.PTR + %CHARCOUNT('::') - .TRY_PTR);  
1859 1981 4 TRY_PTR = .PTR + %CHARCOUNT('::');  
1860 1982 3 END;  
1861 1983 3 IF NOT CH$FAIL(PTR = CH$FIND_CH(.T_SENDER[0],.T_SENDER[1],%ASCII ' '))  
1862 1984 3 THEN T_SENDER[0] = .PTR = .T_SENDER[1];  
1863 1985 3 STR$COPY_DX(.SENDER,T_SENDER);  
1864 1986 2 END;  
1865 1987 2
```

```
1866 1988 2 IF NOT NULLPARAMETER(2)
1867 1989 3 THEN STR$COPY_DX(.SUBJ,MAIL$L_CURMSG[MMB_Q_SUBJDESC]);
1868 1990 3
1869 1991 2 :
1870 1992 2 :
1871 1993 2 :
1872 1994 2 :
1873 1995 3 IF NOT NULLPARAMETER(3)
1874 1996 3 THEN BEGIN
1875 1997 3 DATEDESC[0] = 20;
1876 1998 3 DATEDESC[1] = DATEBUF;
1877 1999 3 $FAO($DESCRIPTOR('!20%D'),DATEDESC,ATEDESC,HDREC[MFR_Q_DATIM]);
1878 2000 3 STR$COPY_DX(.DATE,DATEDESC);
1879 2001 2 :
1880 2002 2 :
1881 2003 2 :
1882 2004 2 :
1883 2005 3 IF NOT NULLPARAMETER(4)
1884 2006 3 THEN BEGIN
1885 2007 3 LOCAL
1886 2008 3 NEWMSG,
1887 2009 3 REPLIED;
1888 2010 4 IF (NEWMSG = .MAIL$L_CURMSG[MMB_V_NEWMSG])
1889 2011 4 OR (REPLIED = .MAIL$L_CURMSG[MMB_V_REPLIED])
1890 2012 4 OR .MAIL$L_CURMSG[MMB_L_NRECS] NEQ 0
1891 2013 4 OR .MAIL$L_CURMSG[MMB_V_EXTMSG]
1892 2014 4 THEN BEGIN
1893 2015 4 STR$COPY_DX(DYN_DESC,$DESCRIPTOR(' '));
1894 2016 4 IF .NEWMSG
1895 2017 4 THEN STR$APPEND(DYN_DESC,MAIL$GETMSG(MAIL$INFO10));
1896 2018 4 IF .REPLIED
1897 2019 4 THEN STR$APPEND(DYN_DESC,MAIL$GETMSG(MAIL$INFO20));
1898 2020 4 IF .MAIL$L_CURMSG[MMB_L_NRECS] NEQ 0
1899 2021 4 THEN BEGIN
1900 2022 4 IF .NEWMSG OR .REPLIED
1901 2023 4 THEN STR$APPEND(DYN_DESC,$DESCRIPTOR(', '));
1902 2024 4 STR$APPEND(DYN_DESC,
1903 2025 4 MAIL$FAO_GETMSG(MAIL$INFO30,.MAIL$L_CURMSG[MMB_L_NRECS]));
1904 2026 4 END;
1905 2027 4 IF .MAIL$L_CURMSG[MMB_V_EXTMSG]
1906 2028 4 THEN BEGIN
1907 2029 4 IF .NEWMSG OR .REPLIED
1908 2030 4 OR .MAIL$L_CURMSG[MMB_L_NRECS] GTRU 0
1909 2031 4 THEN STR$APPEND(DYN_DESC,$DESCRIPTOR(', '));
1910 2032 4 STR$APPEND(DYN_DESC,
1911 2033 4 MAIL$FAO_GETMSG(MAIL$INFO50,
1912 2034 4 .(MAIL$L_CURMSG[MMB_Q_DATID])<32,32,0>,
1913 2035 4 .(MAIL$L_CURMSG[MMB_Q_DATID])<0,32,0>));
1914 2036 4 END;
1915 2037 4 STR$COPY_DX(.FLAGS,DYN_DESC);
1916 2038 4 END
1917 2039 3 ELSE STR$FREE1_DX(.FLAGS);
1918 2040 2 :
1919 2041 2 :
1920 2042 2 :
1921 2043 2 :
1922 2044 2 IF NOT NULLPARAMETER(5)
THEN STR$COPY_DX(.FULL_SENDER,MAIL$L_CURMSG[MMB_Q_FROMDESC]);
```

```
1923 2045 2 |
1924 2046 2 | Get full sender name without stuff in quotes
1925 2047 2 |
1926 2048 2 | IF NOT NULLPARAMETER(6)
1927 2049 2 | THEN BEGIN
1928 2050 3 |     LOCAL
1929 2051 3 |         T_CHAR : BYTE,
1930 2052 3 |         QQUOTE_FLAG;
1931 2053 3 |
1932 2054 3 |     CH$MOVE(DSC$C_S_BLN,MAIL$L_CURMSG[MMB_Q_FROMDESC],T_SENDER);
1933 2055 3 |     QQUOTE_FLAG = 0;
1934 2056 3 |     INCRU-I FROM 0 TO .T_SENDER[0] - 1
1935 2057 4 |     DO BEGIN
1936 2058 4 |         IF (T_CHAR = CH$RCHAR(.T_SENDER[I]+.1)) EQL %ASCII '''
1937 2059 4 |             THEN QQUOTE_FLAG = .QQUOTE_FLAG XOR 1
1938 2060 4 |             ELSE IF .T_CHAR EQL %ASCII ' '
1939 2061 4 |                 AND NOT .QQUOTE_FLAG
1940 2062 5 |                 THEN BEGIN
1941 2063 5 |                     T_SENDER[0] = .I;
1942 2064 5 |                     EXITLOOP;
1943 2065 4 |                 END;
1944 2066 3 |             END;
1945 2067 3 |     STR$COPY_DX(.SENDER_PATH,T_SENDER);
1946 2068 2 |     END;
1947 2069 2 |
1948 2070 2 | IF NOT NULLPARAMETER(7)
1949 2071 2 | THEN STR$COPY_DX(.TO_DESC,MAIL$L_CURMSG[MMB_Q_TODESC]);
1950 2072 2 |
1951 2073 2 | RETURN TRUE
1952 2074 1 | END;
```

```
44 00 00 3A 3A 014F4 P.ABB: .ASCII \::\<0><0>
25 30 32 21 014F8 P.ABD: .ASCII \!20%D\
014FD .BLKB 3
00000005 01500 P.ABC: .LONG 5
00000000' 01504 .ADDRESS P.ABD
09 01508 P.ABF: .ASCII <9>
01509 .BLKB 3
00000001 0150C P.ABE: .LONG 1
00000000' 01510 .ADDRESS P.ABF
2C 01514 P.ABH: .ASCII \,\
01515 .BLKB 3
00000001 01518 P.ABG: .LONG 1
00000000' 0151C .ADDRESS P.ABH
2C 01520 P.ABJ: .ASCII \,\
01521 .BLKB 3
00000001 01524 P.ABI: .LONG 1
00000000' 01528 .ADDRESS P.ABJ
```

```
OFFC 00000
5B C3 AF 9E 00002
5A 00000000G 00 9E 00006
```

```
.ENTRY MAIL$INFO_1_MSG, Save R2,R3,R4,R5,R6,R7,R8,-; 1931
R9,R10,R11
MOVAB P.ABB, R11
MOVAB STR$APPEND, R10
```

		59	00000000G	00	9E	0000D	MOVAB	STR\$COPY_DX, R9			
		58	000000000'	00	9E	00014	MOVAB	MAIL\$L_CURMSG, R8			
		57	000000000'	00	9E	0001B	MOVAB	DYN_DESC, R7			
		5E		08	C2	00022	SUBL2	#8, SP			
		50		68	D0	00025	MOVL	MAIL\$L_CURMSG, R0		1956	
		56	47	A0	9E	00028	MOVAB	71(R0), R6			
				50	D5	0002C	TSTL	R0		1964	
				08	12	0002E	BNEQ	1\$			
		50	007E8022	8F	D0	00030	MOVL	#8290338, R0		1965	
					04	00037	RET				
				6C	95	00038	1\$: TSTB	(AP)		1967	
				51	13	0003A	BEQL	7\$			
			04	AC	D5	0003C	TSTL	4(AP)			
				4C	13	0003F	BEQL	7\$			
	6E	13	A0	08	28	00041	MOV C3	#8, 19(R0), T_SENDER		1969	
			55	6E	D0	00046	MOVL	T_SENDER, TRY_SIZE		1973	
			54	04	AE	D0	00049	MOVL	T_SENDER+4, TRY_PTR	1974	
64		55	6B	02	39	0004D	2\$: MATCHC	#2, P.ABB, TRY_SIZE, (TRY_PTR)		1976	
				03	13	00052	BEQL	3\$			
			53	02	D0	00054	MOVL	#2, R3			
			53	02	C2	00057	3\$: SUBL2	#2, R3			
				16	13	0005A	BEQL	4\$			
			6E	55	D0	0005C	MOVL	TRY_SIZE, T_SENDER		1978	
			AE	54	D0	0005F	MOVL	TRY_PTR, T_SENDER+4		1979	
	50		54	53	C3	00063	SUBL3	PTR, TRY_PTR, R0		1980	
			55	FE	A045	9E	00067	MOVAB	-2(R0)[TRY_SIZE], TRY_SIZE		
			54	02	A3	9E	0006C	MOVAB	2(R3), TRY_PTR		1981
					DB	11	00070	BRB	2\$	1975	
	04	BE	6E		20	3A	00072	4\$: LOCC	#32, T_SENDER, @T_SENDER+4	1983	
					02	12	00077	BNEQ	5\$		
					51	D4	00079	CLRL	R1		
			53		51	D0	0007B	5\$: MOVL	R1, PTR		
					05	13	0007E	BEQL	6\$		
		6E	53	04	AE	C3	00080	SUBL3	T_SENDER+4, PTR, T_SENDER	1984	
					5E	DD	00085	6\$: PUSHL	SP	1985	
				04	AC	DD	00087	PUSHL	SENDER		
			69		02	FB	0008A	CALLS	#2, STR\$COPY_DX		
			02		6C	91	0008D	7\$: CMPB	(AP), #2	1988	
					0F	1F	00090	BLSSU	8\$		
				08	AC	D5	00092	TSTL	3(AP)		
					0A	13	00095	BEQL	3\$		
		7E	68		23	C1	00097	ADDL3	#35, MAIL\$L_CURMSG, -(SP)	1989	
				08	AC	DD	0009B	PUSHL	SUBJ		
			69		02	FB	0009E	CALLS	#2, STR\$COPY_DX		
			03		6C	91	000A1	8\$: CMPB	(AP), #3	1993	
					29	1F	000A4	BLSSU	9\$		
				0C	AC	D5	000A6	TSTL	12(AP)		
					24	13	000A9	BEQL	9\$		
		28	A7		14	D0	000AB	MOVL	#20, DATEDESC	1995	
		2C	A7	08	A7	9E	000AF	MOVAB	DATEBUF, DATEDESC+4	1996	
					56	DD	000B4	PUSHL	R6	1997	
				28	A7	9F	000B6	PUSHAB	DATEDESC		
				28	A7	9F	000B9	PUSHAB	DATEDESC		
				0C	AB	9F	000BC	PUSHAB	P.ABC		
		00000000G	00		04	FB	000BF	CALLS	#4, SYS\$FAO		
				28	A7	9F	000C6	PUSHAB	DATEDESC	1998	
				0C	AC	DD	000C9	PUSHL	DATE		

			69	02	FB	000CC	CALLS	#2, STR\$COPY_DX		
			04	6C	91	000CF	9\$: CMPB	(AP), #4		2003
				03	1E	000D2	BGEQU	11\$		
				00D9	31	000D4	10\$: BRW	22\$		
				10	AC	D5	000D7	11\$: TSTL	16(AP)	
					F8	13	000DA	BEQL	10\$	
			50	68	D0	000DC	MOVL	MAIL\$CURMSG, R0		2009
53	0A	A0	01	00	EF	000DF	EXTZV	#1, 10(R0), NEWMSG		
			16	53	E8	000E5	BLBS	NEWMSG, 12\$		
52	0A	A0	01	01	EF	000E8	EXTZV	#1, #1, 10(R0), REPLIED		2010
			0D	52	E8	000EE	BLBS	REPLIED, 12\$		
				33	A0	D5	000F1	TSTL	51(R0)	2011
					08	12	000F4	BNEQ	12\$	
	03	0A	A0	03	E0	000F6	BBS	#3, 10(R0), 12\$		2012
				00A8	31	000FB	BRW	21\$		
				18	AB	9F	000FE	12\$: PUSHAB	P.ABE	2014
					57	DD	00101	PUSHL	R7	
			69	02	FB	00103	CALLS	#2, STR\$COPY_DX		
			14	53	E9	00106	BLBC	NEWMSG, 13\$		2015
				007EDEA0	8F	DD	00109	PUSHL	#8314528	2016
		00000000G	00	01	FB	0010F	CALLS	#1, MAIL\$GETMSG		
				50	DD	00116	PUSHL	R0		
				57	DD	00118	PUSHL	R7		
			6A	02	FB	0011A	CALLS	#2, STR\$APPEND		
			14	52	E9	0011D	13\$: BLBC	REPLIED, 14\$		2017
				007EDEA8	8F	DD	00120	PUSHL	#8314536	2018
		00000000G	00	01	FB	00126	CALLS	#1, MAIL\$GETMSG		
				50	DD	0012D	PUSHL	R0		
				57	DD	0012F	PUSHL	R7		
			6A	02	FB	00131	CALLS	#2, STR\$APPEND		
			50	68	D0	00134	14\$: MOVL	MAIL\$CURMSG, R0		2019
				33	A0	D5	00137	TSTL	51(R0)	
					28	13	0013A	BEQL	17\$	
			03	53	E8	0013C	BLBS	NEWMSG, 15\$		2021
			08	52	E9	0013F	BLBC	REPLIED, 16\$		
				24	AB	9F	00142	15\$: PUSHAB	P.ABG	2022
					57	DD	00145	PUSHL	R7	
			6A	02	FB	00147	CALLS	#2, STR\$APPEND		
			50	68	D0	0014A	16\$: MOVL	MAIL\$CURMSG, R0		2024
				33	A0	DD	0014D	PUSHL	51(R0)	
				007EDEB0	8F	DD	00150	PUSHL	#8314544	
		00000000G	00	02	FB	00156	CALLS	#2, MAIL\$FAO_GETMSG		
				50	DD	0015D	PUSHL	R0		
				57	DD	0015F	PUSHL	R7		2023
			6A	02	FB	00161	CALLS	#2, STR\$APPEND		
			50	68	D0	00164	17\$: MOVL	MAIL\$CURMSG, R0		2026
			A0	03	E1	00167	BBC	#3, 10(R0), 20\$		
			08	53	E8	0016C	BLBS	NEWMSG, 18\$		2028
			05	52	E8	0016F	BLBS	REPLIED, 18\$		
				33	A0	D5	00172	TSTL	51(R0)	2029
					08	13	00175	BEQL	19\$	
				30	AB	9F	00177	18\$: PUSHAB	P.ABI	2030
					57	DD	0017A	PUSHL	R7	
			6A	02	FB	0017C	CALLS	#2, STR\$APPEND		
			50	68	D0	0017F	19\$: MOVL	MAIL\$CURMSG, R0		2034
				37	A0	DD	00182	PUSHL	55(R0)	
				38	A0	DD	00185	PUSHL	59(R0)	2033

00000000G	00	007EDED8	8F DD 00188	PUSHL	#8314584	2032
			03 FB 0018E	CALLS	#3, MAIL\$FAO_GETMSG	
			50 DD 00195	PUSHL	R0	
			57 DD 00197	PUSHL	R7	2031
	6A		02 FB 00199	CALLS	#2, STR\$APPEND	
			57 DD 0019C	PUSHL	R7	2036
		10	AC DD 0019E	PUSHL	FLAGS	
	69		02 FB 001A1	CALLS	#2, STR\$COPY_DX	
			0A 11 001A4	BRB	22\$	2009
		10	AC DD 001A6	PUSHL	FLAGS	2038
00000000G	00		01 FB 001A9	CALLS	#1, STR\$FREE1_DX	
	05		6C 91 001B0	CMPB	(AP), #5	2043
			0F 1F 001B3	BLSSU	23\$	
		14	AC D5 001B5	TSTL	20(AP)	
			0A 13 001B8	BEQL	23\$	
7E	68		13 C1 001BA	ADDL3	#19, MAIL\$L_CURMSG, -(SP)	2044
		14	AC DD 001BE	PUSHL	FULL_SENDER	
	69		02 FB 001C1	CALLS	#2, STR\$COPY_DX	
	06		6C 91 001C4	CMPB	(AP), #6	2048
			45 1F 001C7	BLSSU	29\$	
		18	AC D5 001C9	TSTL	24(AP)	
			40 13 001CC	BEQL	29\$	
	50		68 D0 001CE	MOVL	MAIL\$L_CURMSG, R0	2054
6E	13	A0	08 28 001D1	MOV3	#8, 197(R0), T_SENDER	
			52 D4 001D6	CLRL	QUOTE_FLAG	2055
53	6E		01 C3 001D8	BL3	#1, T_SENDER, R3	2056
			50 D4 001DC	CLRL	I	
			21 11 001DE	BRB	27\$	
	51	04 BE40	9A 001E0	MOVZBL	@T_SENDER+4[I], R1	2058
	54		51 90 001E5	MOVB	R1, T_CHAR	
	22		51 91 001E8	CMPB	R1, #34	
			05 12 001EB	BNEQ	25\$	
	52		01 8C 001ED	XORB2	#1, QUOTE_FLAG	2059
			0D 11 001F0	BRB	26\$	
	20		54 91 001F2	CMPB	T_CHAR, #32	2060
			08 12 001F5	BNEQ	26\$	
	05		52 E8 001F7	BLBS	QUOTE_FLAG, 26\$	2061
	6E		50 D0 001FA	MOVL	I, T_SENDER	2063
			07 11 001FD	BRB	28\$	2062
			50 D6 001FF	INCL	I	2056
	53		50 D1 00201	CMPL	I, R3	
			DA 1B 00204	BLEQU	24\$	
			5E DD 00206	PUSHL	SP	2067
		18	AC DD 00208	PUSHL	SENDER_PATH	
	69		02 FB 0020B	CALLS	#2, STR\$COPY_DX	
	07		6C 91 0020E	CMPB	(AP), #7	2070
			0F 1F 00211	BLSSU	30\$	
		1C	AC D5 00213	TSTL	28(AP)	
			0A 13 00216	BEQL	30\$	
7E	68		1B C1 00218	ADDL3	#27, MAIL\$L_CURMSG, -(SP)	2071
		1C	AC DD 0021C	PUSHL	T0_DESC	
	69		02 FB 0021F	CALLS	#2, STR\$COPY_DX	
	50		01 D0 00222	MOVL	#1, R0	2073
			04 00225	RET		2074

; Routine Size: 550 bytes, Routine Base: \$CODE\$ + 152C

MAIL\$ISAM_SUBS ISAM MAIL MESSAGE SUBROUTINES
V04-000

N 9
16-Sep-1984 00:54:01
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]ISUBS.B32;1 Page 79
(24)

MAI
V04

: F

```
1954 2075 1 ROUTINE SELECT_MSGS (FOLDER_DESC,SELECT_DATA,OUT_DESC) =
1955 2076 1 ++
1956 2077 1 FUNCTIONAL DESCRIPTION:
1957 2078 1
1958 2079 1 Perform the actual message selection
1959 2080 1
1960 2081 1 --
1961 2082 2 BEGIN
1962 2083 2 MAP
1963 2084 2 FOLDER_DESC : REF $BBLOCK,
1964 2085 2 SELECT_DATA : REF VECTOR[,LONG],
1965 2086 2 OUT_DESC : REF VECTOR[,LONG];
1966 2087 2
1967 2088 2 BIND
1968 2089 2 QUAL_FLAGS = SELECT_DATA[0] : $BBLOCK;
1969 2090 2
1970 2091 2 LOCAL
1971 2092 2 KEYBUFFER : $BBLOCK[MFR_S_FILEKEY],
1972 2093 2 RECDISC : VECTOR[2,LONG],
1973 2094 2 RECPTR : REF VECTOR[,BYTE],
1974 2095 2 MSGDESC : REF $BBLOCK,
1975 2096 2 PTR : REF $BBLOCK,
1976 2097 2 MMH : REF $BBLOCK,
1977 2098 2 DATIM : REF VECTOR[,LONG],
1978 2099 2 STATUS;
1979 2100 2
1980 2101 2 Read on the folder name.
1981 2102 2
1982 2103 2 MAIL$RD_RAB[RAB$R_OP] = RAB$M_LIM; !Read on folder name
1983 2104 2 MAIL$RD_RAB[RAB$B_KRF] = 1;
1984 2105 2 MAIL$RD_RAB[RAB$B_KBF] = KEYBUFFER;
1985 2106 2 MAIL$RD_RAB[RAB$B_KSZ] = MFR_S_FILEKEY;
1986 2107 2 CH$COPY(.FOLDER_DESC[DSC$W_LENGTH],.FOLDER_DESC[DSC$A_POINTER],0,
1987 2108 2 MFR_S_FILEKEY,KEYBUFFER);
1988 2109 2 MAIL$RD_RAB[RAB$B_RAC] = RAB$C_KEY;
1989 2110 2 MAIL$RD_RAB[RAB$V_LOC] = TRUE;
1990 2111 2 MAIL$RD_RAB[RAB$V_NLK] = TRUE;
1991 2112 2 MAIL$RD_RAB[RAB$B_CTX] = MAIL$READERR;
1992 2113 2 IF_ERR$FIND(RAB=MAIL$RD_RAB);
1993 2114 2 IF .QUAL_FLAGS[QUAL_NEW]
1994 2115 2 THEN BEGIN
1995 2116 2 IF .QUAL_FLAGS[QUAL IMPLIED_NEW]
1996 2117 2 THEN BEGIN
1997 2118 2 QUAL_FLAGS[QUAL_NEW] = QUAL_FLAGS[QUAL IMPLIED_NEW] = FALSE,
1998 2119 2 OUT_DESC[0] = MAIL$Q_MSG_FNM;
1999 2120 2 RETURN MAIL$SELECT_MSGS(MAIL$Q_MSG_FNM,.SELECT_DATA)
2000 2121 2 END;
2001 2122 2 MAIL$INCR_NEWMAIL(MAIL$Q_USERNAME,0); !Reset new mail count
2002 2123 2 RETURN SIGNAL(MAIL$_NONE@MAIL);
2003 2124 2 END;
2004 2125 2 IF .STATUS NEQ RM$S_RNF
2005 2126 2 THEN RETURN UTI[$REPORT IO ERROR(MAIL$RD_RAB)
2006 2127 2 ELSE RETURN SIGNAL(MAIL$_NOTEXIST,1,.FOLDER_DESC));
2007 2128 2
2008 2129 2 MAIL$RD_RAB[RAB$B_RAC] = RAB$C_SEQ;
2009 2130 2 WHILE TRUE
2010 2131 2 DO BEGIN
```

```
2011 2132 3 STATUS = $GET(RAB=MAIL$RD_RAB);
2012 2133 3 IF .STATUS EQL RMSS OK LIM
2013 2134 3 OR .STATUS EQL RMSS RNF
2014 2135 3 OR .STATUS EQL RMSS EOF
2015 2136 3 OR .MAIL$GL_FLAGS[MAIF_V_CTRL CFL]
2016 2137 3 THEN EXIT LOOP;
2017 2138 3 IF NOT .STATUS
2018 2139 3 AND .STATUS NEQ RMSS RLK
2019 2140 3 THEN RETURN UTIL$REPORT_TO_ERROR(MAIL$RD_RAB)
2020 2141 4 ELSE BEGIN
2021 2142 4 PTR = .MAIL$RD_RAB[RAB$L_RBF];
2022 2143 4 DATIM = PTR[MFR_Q_DATIM];
2023 2144 4 MMH = PTR[MFR_T_DATA];
2024 2145 5 IF (.DATIM[1] GTRU .SELECT_DATA[4]
2025 2146 7 OR ((.DATIM[1] EQLU .SELECT_DATA[4])
2026 2147 5 AND (.DATIM[0] GTRU .SELECT_DATA[3])))
2027 2148 5 AND (.DATIM[1] LSSU .SELECT_DATA[2]
2028 2149 7 OR ((.DATIM[1] EQLU .SELECT_DATA[2])
2029 2150 5 AND (.DATIM[0] LSSU .SELECT_DATA[1])))
2030 2151 5 AND (IF .QUAL_FLAGS[QUAL_NEW]
2031 2152 5 THEN .MMH[MMH_V_NEWMSG]
2032 2153 5 ELSE TRUE)
2033 2154 5 THEN BEGIN
2034 2155 5 IF_ERR(GET_MEM(MMB_C_LENGTH+.MAIL$RD_RAB[RAB$W_RSZ],MSGDESC);,
2035 2156 5 RETURN .STATUS);
2036 2157 5 MSGDESC[MMB_L_RFA0] = .MAIL$RD_RAB[RAB$L_RFA0];
2037 2158 5 MSGDESC[MMB_W_RFA4] = .MAIL$RD_RAB[RAB$W_RFA4];
2038 2159 5 MSGDESC[MMB_L_TXTLST] = 0;
2039 2160 5 MSGDESC[MMB_W_HDRECLN] = .MAIL$RD_RAB[RAB$W_RSZ];
2040 2161 5 CH$MOVE(.MSGDESC[MMB_W_HDRECLN],
2041 2162 5 .MAIL$RD_RAB[RAB$L_RBF],MSGDESC[MMB_T_HDREC]);
2042 2163 5 MSGDESC[MMB_W_FLAGS] = .MMH[MMH_W_FLAGS];
2043 2164 5 CH$MOVE(8,MMH[MMH_Q_DATID],MSGDESC[MMB_Q_DATID]);
2044 2165 5 DECODE_MISCDATA(.MSGDESC);
2045 2166 5 IF_ERR(ADD_MESSAGE(.MSGDESC);,
2046 2167 5 RETURN .STATUS);
2047 2168 4 END;
2048 2169 3 END;
2049 2170 2 END;
2050 2171 2
2051 2172 2 If selecting new messages and none found, clear user's new msg count
2052 2173 2
2053 2174 2 IF .MAIL$L_NMSG EQL 0
2054 2175 2 AND .QUAL_FLAGS[QUAL_NEW]
2055 2176 3 THEN BEGIN
2056 2177 3 IF .QUAL_FLAGS[QUAL IMPLIED_NEW]
2057 2178 4 THEN BEGIN
2058 2179 4 QUAL_FLAGS[QUAL_NEW] = QUAL_FLAGS[QUAL IMPLIED_NEW] = FALSE;
2059 2180 4 OUT_DESC[0] = MAIL$Q_MSG_FNM;
2060 2181 4 RETURN MAIL$SELECT_MSGS(MAIL$Q_MSG_FNM,.SELECT_DATA)
2061 2182 3 END;
2062 2183 3 IF NOT .MAIL$GL_FLAGS[MAIF_V_CTRL CFL]
2063 2184 3 THEN MAIL$INCR_NEWMAIL(MAIL$Q_USERNAME,0);
2064 2185 3 RETURN SIGNAL(MAIL$_NONEWMAIL)
2065 2186 2 END;
2066 2187 2
2067 2188 2 RETURN TRUE
```

				OFFC 00000	SELECT_MSGS:		
		SE	34	C2	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11
		5A	08	AC	D0	SUBL2	#52, SP
	00000000G	00	4000	8F	3C	MOVL	SELECT_DATA, R10
	00000000G	00	0C	AE	9E	MOVZWL	#16384, MAILST_RD_RAB+4
	00000000G	00	0127	8F	B0	MOVAB	KEYBUFFER, MAILST_RD_RAB+48
		56	04	AC	D0	MOVW	#295, MAILST_RD_RAB+52
27	00	04	B6	66	2C	MOVL	FOLDER_DESC, R6
			0C	AE	0002D	MOVCS	(R6), #4(R6), #0, #39, KEYBUFFER
	00000000G	00		01	90	MOVB	#1, MAILST_RD_RAB+30
	00000000G	00		11	88	BISB2	#17, MAILST_RD_RAB+6
	00000000G	00	007E10B0	8F	D0	MOVL	#8261808, MAILST_RD_RAB+24
			00000000G	00	9F	PUSHAB	MAILST_RD_RAB
	00000000G	00		01	FB	CALLS	#1, SYSS\$FIND
		52		50	D0	MOVL	R0, STATUS
		28		52	E8	BLBS	STATUS, 3\$
09		6A		04	E1	BBC	#4, (R10), 2\$
		03		6A	E9	BLBC	(R10), 1\$
			0117	31	00062	BRW	15\$
			0137	31	00065	BRW	17\$
	000182B2	8F		52	D1	CMPL	STATUS, #98994
				59	12	BNEQ	7\$
				56	DD	PUSHL	R6
				01	DD	PUSHL	#1
			007E80CA	8F	DD	PUSHL	#8290506
	00000000G	00		03	FB	CALLS	#3, LIB\$SIGNAL
				04	00082	RET	
			00000000G	00	94	CLRB	MAILST_RD_RAB+30
			00000000G	00	9F	PUSHAB	MAILST_RD_RAB
	00000000G	00		01	FB	CALLS	#1, SYSS\$GET
		5B		50	D0	MOVL	R0, STATUS
	00018051	8F		5B	D1	CMPL	STATUS, #98385
				19	13	BEQL	5\$
	000182B2	8F		5B	D1	CMPL	STATUS, #98994
				10	13	BEQL	5\$
	0001827A	8F		5B	D1	CMPL	STATUS, #98938
				07	13	BEQL	5\$
		03	00000000G	00	E9	BLBC	MAIL\$GL_FLAGS+1, 6\$
				00AF	31	BRW	14\$
		17		5B	E8	BLBS	STATUS, 8\$
	000182AA	8F		5B	D1	CMPL	STATUS, #98986
				0E	13	BEQL	8\$
			00000000G	00	9F	PUSHAB	MAILST_RD_RAB
	00000000G	00		01	FB	CALLS	#1, UTIL\$REPORT_IO_ERROR
				04	000D7	RET	
		57	00000000G	00	D0	MOVL	MAILST_RD_RAB+40, PTR
		59		57	D0	MOVL	PTR, DATIM
		56	30	A7	9E	MOVAB	48(R7), MMH
	10	AA	04	A9	D1	CMPL	4(DATIM), 16(R10)
				08	1A	BGTRU	9\$

	0C	AA		9A	12	000ED	BNEQ	4\$	:	2146	
				69	D1	000EF	CMPL	(DATIM), 12(R10)	:	2147	
	08	AA	04	94	1B	000F3	BLEQU	4\$	:		
				A9	D1	000F5	9\$:	CMPL	4(DATIM), 8(R10)	:	2148
				08	1F	000FA	BLSSU	10\$	:		
	04	AA		8B	12	000FC	BNEQ	4\$	:	2149	
				69	D1	000FE	CMPL	(DATIM), 4(R10)	:	2150	
06		6A		85	1E	00102	BGEQU	4\$	:		
		03		04	E1	00104	10\$:	BBC	#4, (R10), 12\$	:	2151
				66	E8	00108	BLBS	(MMH), 12\$	:	2152	
				FF7B	31	0010B	11\$:	BRW	1\$	:	
				5E	DD	0010E	12\$:	PUSHL	SP	:	2156
		7E	00000000G	00	3C	00110	MOVZWL	MAIL\$T_RD_RAB+34, -(SP)	:		
		6E	00000047	8F	C0	00117	ADDL2	#71, (SP)	:		
	E7E3	CF		02	FB	0011E	CALLS	#2, GET_MEM	:		
		01		50	E8	00123	BLBS	STATUS, -13\$	:		
					04	00126	RET		:		
		58		6E	D0	00127	13\$:	MOVL	MSGDESC, R8	:	2157
		68	00000000G	00	D0	0012A	MOVL	MAIL\$T_RD_RAB+16, (R8)	:		
	04	A8	00000000G	00	B0	00131	MOVW	MAIL\$T_RD_RAB+20, 4(R8)	:	2158	
			06	A8	D4	00139	CLRL	6(R8)	:	2159	
	11	A8	00000000G	00	B0	0013C	MOVW	MAIL\$T_RD_RAB+34, 17(R8)	:	2160	
		50	00000000G	00	D0	00144	MOVL	MAIL\$T_RD_RAB+40, R0	:	2162	
47	A8	60	11	A8	28	0014B	MOV3	17(R8), (R0), 71(R8)	:		
		0A	A8	66	B0	00151	MOVW	(MMH), 10(R8)	:	2163	
37	A8	08	A6	08	28	00155	MOV3	#8, 8(MMH), 55(R8)	:	2164	
				58	DD	0015B	PUSHL	R8	:	2165	
	F831	CF		01	FB	0015D	CALLS	#1, DECODE_MISCDATA	:		
				58	DD	00162	PUSHL	R8	:	2167	
	E7BF	CF		01	FB	00164	CALLS	#1, ADD_MESSAGE	:		
		9F		50	E8	00169	BLBS	STATUS, -11\$	:		
					04	0016C	RET		:		
			00000000'	00	D5	0016D	14\$:	TSTL	MAIL\$L_NMSG	:	2174
				47	12	00173	BNEQ	19\$	:		
43		6A		04	E1	00175	BBC	#4, (R10), 19\$	:	2175	
		1C		6A	E9	00179	BLBC	(R10), 16\$	:	2177	
		6A	7F	8F	8A	0017C	15\$:	BICB2	#127, (R10)	:	2179
	0C	BC	00000000G	00	9E	00180	MOVAB	MAIL\$Q_MSG_FNM, @OUT_DESC	:	2180	
				5A	DD	00188	PUSHL	R10	:	2181	
			00000000G	00	9F	0018A	PUSHAB	MAIL\$Q_MSG_FNM	:		
	00000000V	00		02	FB	00190	CALLS	#2, MAIL\$SELECT_MSGS	:		
					04	00197	RET		:		
		0F	00000000G	00	E8	00198	16\$:	BLBS	MAIL\$GL_FLAGS+1, 18\$	:	2183
				7E	D4	0019F	17\$:	CLRL	-(SP)	:	2184
			00000000G	00	9F	001A1	PUSHAB	MAIL\$Q_USERNAME	:		
	00000000G	00		02	FB	001A7	CALLS	#2, MAIL\$INCR_NEWMAIL	:		
			007EDF40	8F	DD	001AE	18\$:	PUSHL	#8314688	:	2185
	00000000G	00		01	FB	001B4	CALLS	#1, LIB\$SIGNAL	:		
					04	001BB	RET		:		
		50		01	D0	001BC	19\$:	MOVL	#1, R0	:	2188
					04	001BF	RET		:	2189	

; Routine Size: 448 bytes, Routine Base: \$CODE\$ + 1752

```
2070 2190 1 GLOBAL ROUTINE MAIL$SELECT_MSGS(FOLDER_DESC,SELECT_DATA) =
2071 2191 1 **
2072 2192 1 FUNCTIONAL DESCRIPTION:
2073 2193 1
2074 2194 1 Select a new set of current msgs
2075 2195 1
2076 2196 1 INPUTS:
2077 2197 1
2078 2198 1 FOLDER_DESC is addr of descriptor of folder
2079 2199 1 SELECT_DATA is addr of select data vector
2080 2200 1 --
2081 2201 2 BEGIN
2082 2202 2
2083 2203 2 MAP
2084 2204 2 FOLDER_DESC : REF $BBLOCK,
2085 2205 2 SELECT_DATA : REF VECTOR[ ,LONG];
2086 2206 2
2087 2207 2 BIND
2088 2208 2 QUAL_FLAGS = SELECT_DATA[0] : $BBLOCK;
2089 2209 2
2090 2210 2 LOCAL
2091 2211 2 DESC : REF $BBLOCK,
2092 2212 2 FOLDERSEL,
2093 2213 2 STATUS;
2094 2214 2
2095 2215 2
2096 2216 2 ! If explicit /NEW, ensure in default message file
2097 2217 2
2098 2218 2 IF .QUAL_FLAGS[QUAL_NEW]
2099 2219 2 AND NOT .MAIL$GL_FLAGS[MAIF_V_DFMSG]
2100 2220 2 THEN BEGIN
2101 2221 2 LOCAL
2102 2222 2 TDESC : VECTOR[2,LONG];
2103 2223 2
2104 2224 2 TDESC[0] = 0;
2105 2225 2 TDESC[1] = TDESC[0];
2106 2226 2 MAIL$OPEN_FILE(TDESC);
2107 2227 2 IF NOT .MAIL$GL_FLAGS[MAIF_V_DFMSG]
2108 2228 2 THEN RETURN FALSE;
2109 2229 2
2110 2230 2 END;
2111 2231 2 MAIL$DESELECT_MSGS(); !Delete any current messages
2112 2232 2
2113 2233 2 FOLDERSEL = .FOLDER_DESC[DSC$W_LENGTH] NEQ 0;
2114 2234 2 IF NOT .FOLDERSEL
2115 2235 2 THEN DESC = MAIL$Q MSG_FNM
2116 2236 2 ELSE DESC = .FOLDER_DESC;
2117 2237 2 IF .QUAL_FLAGS[QUAL_NEW]
2118 2238 2 OR NOT (.QUAL_FLAGS[QUAL_BEFORE] OR .QUAL_FLAGS[QUAL_SINCE]
2119 2239 2 OR .FOLDERSEL)
2120 2240 2 AND .MAIL$GL_FLAGS[MAIF_V_DFMSG]
2121 2241 2 THEN BEGIN
2122 2242 2 IF NOT .QUAL_FLAGS[QUAL_NEW]
2123 2243 2 THEN QUAL_FLAGS[QUAL IMPLIED_NEW] = TRUE;
2124 2244 2 QUAL_FLAGS[QUAL_NEW] = TRUE;
2125 2245 2 QUAL_FLAGS[QUAL_SINCE] = TRUE;
2126 2246 2 DESC = MAIL$SD_NEWMAIL;
```



```
2127      2247 2      END;  
2128      2248 2 STATUS = SELECT MSGS(.DESC,.SELECT_DATA,DESC);  
2129      2249 2 MAIL$GL_FLAGS[MAIF_V_CTRL[FL]] = 0;  
2130      2250 2 IF .STATUS  
2131      2251 3     AND (.MAIL$L_NMSG GTRU 0)  
2132      2252 2     THEN STR$COPY_DX(MAIL$Q_CURFOLDER,.DESC);  
2133      2253 2  
2134      2254 2 RETURN .STATUS  
2135      2255 1 END;
```

				001C 00000	.ENTRY MAIL\$SELECT MSGS, Save R2,R3,R4	2190
		54	F83B	CF 9E 00002	MOVAB MAIL\$DESELECT_MSGS, R4	
		53	00000000G	00 9E 00007	MOVAB MAIL\$GL_FLAGS, R3	
		5E		0C C2 0000E	SUBL2 #12, SP	
		52	08	AC D0 00011	MOVL SELECT_DATA, R2	2208
1C		62		04 E1 00015	BBC #4, (R2), 1\$	2218
17	01	A3		04 E0 00019	BBS #4, MAIL\$GL_FLAGS+1, 1\$	2219
			04	AE D4 0001E	CLRL TDESC	2224
	08	AE	04	AE 9E 00021	MOVAB TDESC, TDESC+4	2225
			04	AE 9F 00026	PUSHAB TDESC	2226
	00000000G	00		01 FB 00029	CALLS #1, MAIL\$OPEN_FILE	
71	01	A3		04 E1 00030	BBC #4, MAIL\$GL_FLAGS+1, 8\$	2227
		64		00 FB 00035	CALLS #0, MAIL\$DESELECT_MSGS	2231
				50 D4 00038	CLRL R0	2233
			04	BC B5 0003A	TSTW @FOLDER_DESC	
				02 13 0003D	BEQL 2\$	
				50 D6 0003F	INCL R0	
	09			50 E8 00041	BLBS FOLDERSEL, 3\$	2234
	6E	00000000G		00 9E 00044	MOVAB MAIL\$Q_MSG_FNM, DESC	2235
				04 11 0004B	BRB 4\$	
	6E	04		AC D0 0004D	MOVL FOLDER_DESC, DESC	2236
18		62		04 E0 00051	BBS #4, (R2), 5\$	2237
		1D		62 E8 00055	BLBS (R2), 6\$	2238
19		62		01 E0 00058	BBS #1, (R2), 6\$	
		16		50 E8 0005C	BLBS FOLDERSEL, 6\$	2239
11	01	A3		04 E1 0005F	BBC #4, MAIL\$GL_FLAGS+1, 6\$	2240
05		62		04 E0 00064	BBS #4, (R2), 5\$	2242
07		00		01 F0 00068	INSV #1, #0, #7, (R2)	2243
		62		12 88 0006D	BISB2 #18, (R2)	2245
		6E	E682	CF 9E 00070	MOVAB MAIL\$SD_NEWMAIL, DESC	2246
			4004	8F BB 00075	PUSHR #*M<R2,5P>	2248
			08	AE DD 00079	PUSHL DESC	
	05FF	C4		03 FB 0007C	CALLS #3, SELECT_MSGS	
		52		50 D0 00081	MOVL R0, STATUS	
	01	A3		01 8A 00084	BICB2 #1, MAIL\$GL_FLAGS+1	2249
		17		52 E9 00088	BLBC STATUS, 7\$	2250
			00000000'	00 D5 0008B	TSTL MAIL\$L_NMSG	2251
				0F 13 00091	BEQL 7\$	
			00000000'	6E DD 00093	PUSHL DESC	2252
				00 9F 00095	PUSHAB MAIL\$Q_CURFOLDER	
	00000000G	00		02 FB 0009B	CALLS #2, STR\$COPY_DX	
		50		52 D0 000A2	MOVL STATUS, R0	2254
				04 000A5	RET	

MAIL\$ISAM_SUBS ISAM MAIL MESSAGE SUBROUTINES
V04-000

H 10
16-Sep-1984 00:54:01
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]ISUBS.B32;1 Page 86
(26)

MAI
V04

50 D4 000A6 8\$: CLRL R0
04 000A8 RET

: 2255
:

; Routine Size: 169 bytes, Routine Base: \$CODE\$ + 1912


```

2137 2256 1 GLOBAL ROUTINE MAIL$REMOVE_I_MSG =
2138 2257 1  **
2139 2258 1  FUNCTIONAL DESCRIPTION:
2140 2259 1
2141 2260 1      Deallocate memory for msg and zero entry in msg tbl
2142 2261 1
2143 2262 1  INPUTS:
2144 2263 1
2145 2264 1      MAIL$CURMSG points to descriptor
2146 2265 1      MAIL$MSG_NUM is msg number
2147 2266 1  --
2148 2267 2 BEGIN
2149 2268 2
2150 2269 2 DEALLOC MSG();
2151 2270 2 MAIL$MSGTBL[.MAIL$MSG_NUM-1] = 0;
2152 2271 2
2153 2272 2 MAIL$MSG_STS = MAIL$NOTREADIN;
2154 2273 2
2155 2274 2 RETURN TRUE
2156 2275 2
2157 2276 1 END;

```

```

          EBDC  CF          0000 00000
                   51 00000000' 00 FB 00002
                   50 00000000G 00 D0 00007
                                FC A140 D4 00015
00000000G 00 007E8022 8F D0 00019
                   50          01 D0 00024
                                04 00027

```

```

.ENTRY MAIL$REMOVE_I_MSG, Save nothing      : 2256
CALLS  #0, DEALLOC MSG                      : 2269
MOVL   MAIL$MSGTBL, R1                      : 2270
MOVL   MAIL$MSG_NUM, R0
CLRL   -4(R1)[R0]
MOVL   #8290338, MAIL$MSG_STS               : 2272
MOVL   #1, R0                               : 2274
RET                                          : 2276

```

; Routine Size: 40 bytes, Routine Base: \$CODE\$ + 198B

```
2159 2277 1 GLOBAL ROUTINE MAIL$DELETE_I_MSG =
2160 2278 1 ++
2161 2279 1 FUNCTIONAL DESCRIPTION:
2162 2280 1
2163 2281 1 Delete msg by filing into wastebasket folder
2164 2282 1
2165 2283 1 INPUTS:
2166 2284 1
2167 2285 1 MAIL$CURMSG points to msg
2168 2286 1 MAIL$MSG_NUM is msg number
2169 2287 1 --
2170 2288 2 BEGIN
2171 2289 2
2172 2290 2 LOCAL
2173 2291 2 MMH : REF $BBLOCK,
2174 2292 2 HDREC : REF $BBLOCK;
2175 2293 2
2176 2294 2 IF .MAIL$CURMSG EQL 0
2177 2295 2 THEN RETURN MAIL$_NOTREADIN;
2178 2296 2
2179 2297 2 Don't allow a DELETE command in the wastebasket folder
2180 2298 2
2181 2299 2 IF CH$EQL(.MAIL$CURFOLDER[DSC$W_LENGTH],.MAIL$CURFOLDER[DSC$A_POINTER],
2182 2300 2 .MAIL$WASTEDESC[DSC$W_LENGTH],
2183 2301 2 .MAIL$WASTEDESC[DSC$A_POINTER],0)
2184 2302 2 THEN RETURN MAIL$_DELWASTE;
2185 2303 2
2186 2304 2 HDREC = MAIL$CURMSG[MMB_T_HDREC];
2187 2305 2 IF .MAIL$CURMSG[MMB_V_NEWMSG]
2188 2306 3 THEN BEGIN
2189 2307 3 MMH = HDREC[MFR_T_DATA];
2190 2308 3 MMH[MMH_V_NEWMSG] = 0;
2191 2309 3 MAIL$CURMSG[MMB_V_NEWMSG] = 0;
2192 2310 3 MAIL$DECR_NEWMAIL(MAIL$CURFOLDER[DSC$A_POINTER]);
2193 2311 2 END;
2194 2312 2 HDREC[MFR_B_FILEKEYLEN] = .MAIL$WASTEDESC[DSC$W_LENGTH];
2195 2313 2 CH$COPY(.MAIL$WASTEDESC[DSC$W_LENGTH],.MAIL$WASTEDESC[DSC$A_POINTER],0,
2196 2314 2 MFR_S_FILEKEY,HDREC[MFR_T_FILEKEY]);
2197 2315 2
2198 P 2316 2 IF _ERR(MAIL$UPDATE_I_MSG()),
2199 2317 2 RETURN .STATOS;
2200 2318 2
2201 2319 2 MAIL$REMOVE_I_MSG();
2202 2320 2 RETURN TRUE
2203 2321 2
2204 2322 1 END;
```

56 00000000'	00	007C 00000	.ENTRY MAIL\$DELETE_I_MSG, Save R2,R3,R4,R5,R6	: 2277
54 0C	A6	9E 00002	MOVAB MAIL\$WASTEDESC+4, R6	: 2294
	08	D0 00009	MOVL MAIL\$CURMSG, R4	: 2295
50 007E8022	8F	12 0000D	BNEQ 1\$	: 2295
	D0	0000F	MOVL #8290338, R0	: 2295
	04	00016	RET	: 2295

FC	A6	00	51	F8	A6	D0	00017	1\$:	MOVL	MAIL\$Q_CURFOLDER+4, R1	2299
			50		66	D0	0001B		MOVL	MAIL\$Q_WASTEDESC+4, R0	2301
			61	F4	A6	2D	0001E		CMPCS	MAIL\$Q_CURFOLDER, (R1), #0, -	2299
					60		00025			MAIL\$Q_WASTEDESC, (R0)	
					08	12	00026		BNEQ	2\$	
			50	007E8CBA	8F	D0	00028		MOVL	#8290490, R0	2302
						04	0002F		RET		
			52	47	A4	9E	00030	2\$:	MOVAB	71(R4), HDREC	2304
			18	0A	A4	E9	00034		BLBC	10(R4), 3\$	2305
			50	30	A2	9E	00038		MOVAB	48(R2), MMH	2307
			60		01	8A	0003C		BICB2	#1, (MMH)	2308
	0A		A4		01	8A	0003F		BICB2	#1, 10(R4)	2309
					00	9F	00043		PUSHAB	MAIL\$Q_USERNAME	2310
	00000000G		00		01	FB	00049		CALLS	#1, MAIL\$DECR_NEWMAIL	
			50	FC	A6	3C	00050	3\$:	MOVZWL	MAIL\$Q_WASTEDESC, R0	2312
		08	A2		50	90	00054		MOVB	R0, 8(HDREC)	
			51		66	D0	00058		MOVL	MAIL\$Q_WASTEDESC+4, R1	2313
27		00	61		50	2C	0005B		MOVCS	R0, (RT), #0, #39, 9(HDREC)	2314
				09	A2		00060				
	EED7		CF		00	FB	00062		CALLS	#0, MAIL\$UPDATE_I_MSG	2317
			08		50	E9	00067		BLBC	STATUS, 4\$	
	FF69		CF		00	FB	0006A		CALLS	#0, MAIL\$REMOVE_I_MSG	2319
			50		01	D0	0006F		MOVL	#1, R0	2320
					04	00072	4\$:		RET		2322

; Routine Size: 115 bytes, Routine Base: \$CODE\$ + 19E3

```
2206 2323 1 GLOBAL ROUTINE MAIL$FILE_1_MSG (NEW_FOLDER,CONFIRM_FLAG) =
2207 2324 1 ++
2208 2325 1 FUNCTIONAL DESCRIPTION:
2209 2326 1
2210 2327 1 File a msg into a different folder
2211 2328 1
2212 2329 1 INPUTS:
2213 2330 1
2214 2331 1 NEW_FOLDER addr of descriptor of new folder name
2215 2332 1 CONFIRM_FLAG = true to ask if new folder is about to be created,
2216 2333 1 false to just signal the fact
2217 2334 1
2218 2335 1 IMPLICIT INPUTS:
2219 2336 1
2220 2337 1 MAIL$CURMSG points to msg
2221 2338 1
2222 2339 1 ROUTINE VALUE:
2223 2340 1
2224 2341 1 Returns RMSS_RNF if not confirmed to create new folder
2225 2342 1 --
2226 2343 2 BEGIN
2227 2344 2
2228 2345 2 MAP
2229 2346 2 NEW_FOLDER : REF $BLOCK;
2230 2347 2
2231 2348 2 LOCAL
2232 2349 2 NEWF,
2233 2350 2 MMH : REF $BLOCK,
2234 2351 2 HDREC : REF $BLOCK;
2235 2352 2
2236 2353 2 IF .MAIL$CURMSG EQL 0
2237 2354 2 THEN RETURN MAIL$_NOTREADIN;
2238 2355 2
2239 2356 2 HDREC = MAIL$CURMSG[MMB_T_HDREC];
2240 2357 2 HDREC[MFR_B_FILEKEYLEN] = .NEW_FOLDER[DSC$W_LENGTH];
2241 2358 2 CH$COPY(.NEW_FOLDER[DSC$W_LENGTH],.NEW_FOLDER[DSC$A_POINTER],0,
2242 2359 2 MFR_S_FILEKEY,HDREC[MFR_T_FILEKEY]);
2243 2360 2
2244 2361 2 If folder doesn't exist, issue informational that folder created
2245 2362 2
2246 2363 2 MAIL$RD_RAB[RAB$L_RDP] = RAB$M_NLK;
2247 2364 2 MAIL$RD_RAB[RAB$L_KBF] = HDREC[MFR_T_FILEKEY];
2248 2365 2 MAIL$RD_RAB[RAB$B_KSZ] = MFR_S_FILEKEY;
2249 2366 2 MAIL$RD_RAB[RAB$B_KRF] = 1;
2250 2367 2 MAIL$RD_RAB[RAB$B_RAC] = RAB$C_KEY;
2251 2368 3 IF NOT (NEWF = $FIND(RAB=MAIL$RD_RAB))
2252 2369 3 THEN BEGIN
2253 2370 3 IF .CONFIRM_FLAG
2254 2371 3 THEN IF NOT MAIL$CONFIRM(YN_DESC,
2255 2372 3 MAIL$FAO_GETMSG(MAIL$_CRENEWFOLD,.NEW_FOLDER))
2256 2373 3 THEN RETURN RMSS_RNF;
2257 2374 2 END;
2258 2375 2
2259 2376 2 IF .MAIL$CURMSG[MMB_V_NEWMSG]
2260 2377 2 THEN BEGIN
2261 2378 3 MMH = HDREC[MFR_T_DATA];
2262 2379 3 MMH[MMH_V_NEWMSG] = 0;
```

```
: 2263      2380 3      MAIL$CURMSG[MMB V NEWMSG] = 0;  
: 2264      2381 3      MAIL$DECR_NEWMAIL(MAIL$Q_USERNAME);  
: 2265      2382 2      END;  
: 2266      2383 2  
: 2267      2384 2      IF_ERR(MAIL$UPDATE I MSG());  
: 2268      2385 2      RETURN .STATUS);  
: 2269      2386 2  
: 2270      2387 2      IF NOT .NEWF  
: 2271      2388 2      THEN SIGNAL(MAIL$_NEWFOLDER,1,.NEW_FOLDER);  
: 2272      2389 2  
: 2273      2390 2      MAIL$REMOVE_I_MSG();  
: 2274      2391 2      RETURN TRUE  
: 2275      2392 2  
: 2276      2393 1      END;
```

				03FC 00000		.ENTRY		
						MAIL\$FILE_I_MSG, Save R2,R3,R4,R5,R6,R7,R8,-;	2323	
						R9		
						MOVAB MAIL\$CURMSG, R9		
						MOVAB MAIL\$RD_RAB+4, R8		
						MOVL MAIL\$CURMSG, R6		2353
						BNEQ 1\$		
						MOVL #8290338, R0		2354
						RET		
						MOVAB 71(R6), HDREC		2356
						MOVL NEW_FOLDER, R7		2357
						MOVB (R7), 8(HDREC)		
						MOVCS (R7), @4(R7), #0, #39, 9(HDREC)		2359
						MOVL #1048576, MAIL\$RD_RAB+4		2363
						MOVAB 9(HDREC), MAIL\$RD_RAB+48		2364
						MOVW #295, MAIL\$RD_RAB+52		2365
						MOVB #1, MAIL\$RD_RAB+30		2367
						PUSHAB MAIL\$RD_RAB		2368
						CALLS #1, SY\$FIND		
						MOVL R0, NEWF		
						BLBS NEWF, 2\$		
						BLBC CONFIRM_FLAG, 2\$		2370
						PUSHL R7		2372
						PUSHL #8314680		
						CALLS #2, MAIL\$FAO_GETMSG		
						PUSHL R0		
						PUSHAB YN_DESC		2371
						CALLS #2, MAIL\$CONFIRM		
						BLBS R0, 2\$		
						MOVL #98994, R0		2373
						RET		
						MOVL MAIL\$CURMSG, R0		2376
						BLBC 10(R0), 3\$		
						ADDL2 #48, MMH		2378
						BICB2 #1, (MMH)		2379
						BICB2 #1, 10(R0)		2380
						PUSHAB MAIL\$Q_USERNAME		2381
						CALLS #1, MAIL\$DECR_NEWMAIL		

N 10
16-Sep-1984 00:54:01 VAX-11 Bliss-32 V4.0-742 Page 92
14-Sep-1984 12:42:20 DISK\$VMSMASTER:[MAIL.SRC]ISUBS.B32;1 (29)

Address	Op Code	Op Name	Op Description	Op Address
EE24	CF	00	FB 000A2 3\$:	CALLS #0, MAIL\$UPDATE_I_MSG ; 2385
	1C	50	E9 000A7	BLBC STATUS, 5\$;
	11	52	E8 000AA	BLBS NEWF, 4\$; 2387
		57	DD 000AD	PUSHL R7 ; 2388
		01	DD 000AF	PUSHL #1 ;
	007EBEAB	8F	DD 000B1	PUSHL #8306347 ;
00000000G	00	03	FB 000B7	CALLS #3, LIB\$SIGNAL ;
FEA2	CF	00	FB 000BE 4\$:	CALLS #0, MAIL\$REMOVE_I_MSG ; 2390
	50	01	DO 000C3	MOVL #1, R0 ; 2391
		04	000C6 5\$:	RET ; 2393

; Routine Size: 199 bytes, Routine Base: \$CODE\$ + 1A56

MAI
V04

.....


```
2278 2394 1 GLOBAL ROUTINE MAIL$INFO_FOLDER(NAME_DESC,SEARCH_MODE) =
2279 2395 1 ++
2280 2396 1 FUNCTIONAL DESCRIPTION:
2281 2397 1
2282 2398 1 Return next folder name
2283 2399 1
2284 2400 1 INPUTS:
2285 2401 1
2286 2402 1 NAME_DESC = addr of descriptor of starting point
2287 2403 1 SEARCH_MODE = 0 to search KGT, 1 to search KGE
2288 2404 1 --
2289 2405 2 BEGIN
2290 2406 2
2291 2407 2 MAP
2292 2408 2 NAME_DESC : REF VECTOR[2, LONG];
2293 2409 2
2294 2410 2 BUILTIN
2295 2411 2 NULLPARAMETER;
2296 2412 2
2297 2413 2 LOCAL
2298 2414 2 STATUS,
2299 2415 2 FOLDERBUF : $BBLOCK[MFR_S_FILEKEY];
2300 2416 2
2301 2417 2 IF .NAME_DESC[0] GTRU 0
2302 2418 2 THEN CH$COPY(.NAME_DESC[0],.NAME_DESC[1],0,MFR_S_FILEKEY,FOLDERBUF)
2303 2419 2 ELSE CH$FILL(0,MFR_S_FILEKEY,FOLDERBUF);
2304 2420 2
2305 2421 2 MAIL$T_RD_RAB[RAB$L_KBF] = FOLDERBUF;
2306 2422 2 MAIL$T_RD_RAB[RAB$B_KSZ] = MFR_S_FILEKEY;
2307 2423 2 MAIL$T_RD_RAB[RAB$B_KRF] = 1;
2308 2424 2 MAIL$T_RD_RAB[RAB$B_RAC] = RAB$C_KEY;
2309 2425 2 MAIL$T_RD_RAB[RAB$L_ROP] = RAB$M_LOC OR RAB$M_NLK;
2310 2426 3 MAIL$T_RD_RAB[RAB$V_KGE] = (IF NOT NULLPARAMETER(2)
2311 2427 3 THEN .SEARCH_MODE
2312 2428 3 ELSE FALSE);
2313 2429 2 MAIL$T_RD_RAB[RAB$V_KGT] = NOT .MAIL$T_RD_RAB[RAB$V_KGE];
2314 2430 2 MAIL$T_RD_RAB[RAB$L_CTX] = MAIL$_READERR;
2315 2431 2
2316 2432 3 IF (STATUS = $GET(RAB=MAIL$T_RD_RAB))
2317 2433 3 THEN BEGIN
2318 2434 3 BIND
2319 2435 3 HDREC = .MAIL$T_RD_RAB[RAB$L_RBF] : $BBLOCK;
2320 2436 3
2321 2437 3 NAME_DESC[0] = .HDREC[MFR_B_FILEKEYLEN];
2322 2438 3 NAME_DESC[1] = HDREC[MFR_T_FILEKEY];
2323 2439 3 END
2324 2440 2 ELSE IF .STATUS NEQ RMSS$RNF
2325 2441 2 THEN UTIL$REPORT_IO_ERROR(MAIL$T_RD_RAB);
2326 2442 2
2327 2443 2 RETURN .STATUS
2328 2444 1 END;
```

00FC 00000

.ENTRY MAIL\$INFO_FOLDER, Save R2,R3,R4,R5,R6,R7 ; 2394

57	00000000G	00	9E	00002	MOVAB	MAIL\$T_RD_RAB+4, R7	:	
5E		28	C2	00009	SUBL2	#40, SP	:	
56	04	AC	D0	0000C	MOVL	NAME_DESC, R6	:	2417
		66	D5	00010	TSTL	(R6)	:	
27	00	04	B6	09	13	00012	BEQL	1\$
				66	2C	00014	MOVCS	(R6), @4(R6), #0, #39, FOLDERBUF
				6E		0001A	:	2418
				06	11	0001B	BRB	2\$
27	00	6E		00	2C	0001D	MOVCS	#0, (SP), #0, #39, FOLDERBUF
				6E		00022	:	2419
	2C	A7		6E	9E	00023	2\$:	MOVAB
	30	A7	0127	8F	B0	00027		FOLDERBUF, MAIL\$T_RD_RAB+48
	1A	A7		01	90	0002D		#295, MAIL\$T_RD_RAB+52
		67	00110000	8F	D0	00031		#1, MAIL\$T_RD_RAB+30
		02		6C	91	00038		#1114112, MAIL\$T_RD_RAB+4
				0B	1F	0003B		(AP), #2
			08	AC	D5	0003D		3\$
				06	13	00040		8(AP)
		50	08	AC	D0	00042		3\$
				02	11	00046		SEARCH_MODE, R0
				50	D4	00048	3\$:	4\$
02	A7	01	05	50	F0	0004A	4\$:	CLRL
	50	A7	01	05	EF	00050		R0, #5, #1, MAIL\$T_RD_RAB+6
			50	50	D2	00056		#5, #1, MAIL\$T_RD_RAB+6, R0
02	A7	01	06	50	F0	00059		R0, R0
			14	A7	007E10B0	8F		R0, #6, #1, MAIL\$T_RD_RAB+6
					FC	A7		#8261808, MAIL\$T_RD_RAB+24
						01		MAIL\$T_RD_RAB
	00000000G	00		01	FB	0006A		#1, SYSSGET
		52		50	D0	00071		R0, STATUS
		0F		52	E9	00074		STATUS, 5\$
		50	24	A7	D0	00077		MAIL\$T_RD_RAB+40, R0
		66	08	A0	9A	0007B		8(R0), -(R6)
	04	A6	09	A0	9E	0007F		9(R0), 4(R6)
				13	11	00084		6\$
	000182B2	8F		52	D1	00086	5\$:	STATUS, #98994
				0A	13	0008D		6\$
			FC	A7	9F	0C08F		MAIL\$T_RD_RAB
	00000000G	00		01	FB	00092		#1, UTIL\$REPORT_IO_ERROR
		50		52	D0	00099	6\$:	STATUS, R0
				04	0009C			RET

; Routine Size: 157 bytes, Routine Base: \$CODE\$ + 1B1D


```
: 2330      2445 1 GLOBAL ROUTINE MAIL$ISAM_INIT =
: 2331      2446 1 |++
: 2332      2447 1 | FUNCTIONAL DESCRIPTION:
: 2333      2448 1 |
: 2334      2449 1 |         Read in mail file info records and set up data for later use
: 2335      2450 1 | --
: 2336      2451 2 BEGIN
: 2337      2452 2
: 2338      2453 2 LOCAL
: 2339      2454 2     DESC : VECTOR[2, LONG],
: 2340      2455 2     RECDISC : VECTOR[2, LONG];
: 2341      2456 2
: 2342      2457 2 |
: 2343      2458 2 |     Get the wastebasket folder name
: 2344      2459 2 |
: 2345      2460 2 DESC[0] = %CHARCOUNT ('WASTEBASKET');
: 2346      2461 2 DESC[1] = UPLIT(%ASCII 'WASTEBASKET');
: 2347      2462 2 IF MAIL$READ_INFO_REC(MAIL$T_RD_RAB, MFR_C_WASTENAME, 0, RECDISC)
: 2348      2463 3 THEN BEGIN
: 2349      2464 3     BIND
: 2350      2465 3     WASTENAME = .RECDISC[1] + $BYTEOFFSET(MFR_T_DATA) : VECTOR[BYTE];
: 2351      2466 3
: 2352      2467 3     IF .WASTENAME[0] NEQ 0
: 2353      2468 4     THEN BEGIN
: 2354      2469 4         DESC[0] = .WASTENAME[0];
: 2355      2470 4         DESC[1] = WASTENAME[1];
: 2356      2471 3     END;
: 2357      2472 2     END;
: 2358      2473 2 STR$COPY_DX(MAIL$Q_WASTEDESC, DESC);
: 2359      2474 2 RETURN TRUE
: 2360      2475 1 END;
```

```
00 54 45 4B 53 41 42 45 54 53 41 57 01BBA P.ABK: .BLKB 2
01BBC .ASCII \WASTEBASKET\<0>
```

				0004 00000	.ENTRY MAIL\$ISAM_INIT, Save R2	: 2445
	52	EF	AF	9E 00002	MOVAB P.ABK, R2	:
	5E		10	C2 00006	SUBL2 #16, SP	:
08	AE		0B	D0 00009	MOVL #11, DESC	: 2460
0C	AE		62	9E 0000D	MOVAB P.ABK, DESC+4	: 2461
			5E	DD 00011	PUSHL SP	: 2462
	7E		02	7D 00013	MOVQ #2, -(SP)	:
		00000000G	00	9F 00016	PUSHAB MAIL\$T_RD_RAB	:
	EDB3	CF	04	FB 0001C	CALLS #4, MAIL\$READ_INFO_REC	:
	12		50	E9 00021	BLBC R0, 1\$	:
50	04	AE	30	C1 00024	ADDL3 #48, RECDISC+4, R0	: 2465
			60	95 00029	TSTB (R0)	: 2467
			09	13 0002B	BEQL 1\$	:
	08	AE	60	9A 0002D	MOVZBL (R0), DESC	: 2469
	0C	AE	A0	9E 00031	MOVAB 1(R0), DESC+4	: 2470
		01	AE	9F 00036	PUSHAB DESC	: 2473
		08	00	9F 00039	PUSHAB MAIL\$Q_WASTEDESC	:
00000000G	00	00000000'	02	FB 0003F	CALLS #2, STR\$COPY_DX	:

E 11
16-Sep-1984 00:54:01 VAX-11 Bliss-32 V4.0-742 Page 96
14-Sep-1984 12:42:20 DISK\$VMSMASTER:[MAIL.SRC]ISUBS.B32;1 (31)

01 DO 00046
04 00049

```

MOVL    #1, R0
RET

```

: 2474
: 2475

```
; Routine Size: 74 bytes,    Routine Base: $CODE$ + 1BC8
```

MAI
V04

```
: 2362      2476 1 ROUTINE DESELECT_IF_WASTEBASKET : NOVALUE =  
: 2363      2477 1 |++  
: 2364      2478 1 | FUNCTIONAL DESCRIPTION:  
: 2365      2479 1 |  
: 2366      2480 1 |         Deselect current messages if current folder is wastebasket folder  
: 2367      2481 1 |  
: 2368      2482 1 |--  
: 2369      2483 2 BEGIN  
: 2370      2484 2 IF CH$EQL(.MAIL$Q_CURFOLDER[DSC$W_LENGTH],.MAIL$Q_CURFOLDER[DSC$A_POINTER],  
: 2371      2485 2     .MAIL$Q_WASTEDESC[DSC$W_LENGTH],.MAIL$Q_WASTEDESC[DSC$A_POINTER],0)  
: 2372      2486 2     THEN MAIL$DESELECT_MSGS();  
: 2373      2487 2 RETURN;  
: 2374      2488 1 END;
```

001C 00000 DESELECT_IF_WASTEBASKET:

		54	00000000'	00	9E	00002	WORD	Save R2,R3,R4	: 2476
		51		64	D0	00009	MOVAB	MAIL\$Q_CURFOLDER+4, R4	: 2484
		50	08	A4	D0	0000C	MOVL	MAIL\$Q_CURFOLDER+4, R1	: 2485
04	A4			61	FC	A4	2D	00010	: 2484
						60		00017	:
						05	12	00018	:
		F522	CF			00	FB	0001A	: 2486
						04	0001F	1\$:	: 2488
								RET	:

: Routine Size: 32 bytes, Routine Base: \$CODE\$ + 1C12

```
2376 2489 1 GLOBAL ROUTINE MAIL$MOVE_WASTEBASKET (DEST_DESC) =
2377 2490 1 1+
2378 2491 1 1- FUNCTIONAL DESCRIPTION:
2379 2492 1 1-
2380 2493 1 1- Move the contents of the wastebasket folder to folder DEST_DESC
2381 2494 1 1- --
2382 2495 2 BEGIN
2383 2496 2
2384 2497 2 MAP
2385 2498 2 DEST_DESC : REF $BBLOCK;
2386 2499 2
2387 2500 2 LOCAL
2388 2501 2 WASTENAMEBUF : $BBLOCK[MFR_S_FILEKEY],
2389 2502 2 RECBUF : REF $BBLOCK;
2390 2503 2
2391 2504 2 DESELECT IF WASTEBASKET();
2392 2505 2 CH$COPY(.MAIL$Q_WASTEDESC[DSC$W_LENGTH],.MAIL$Q_WASTEDESC[DSC$A_POINTER],
2393 2506 2 0,MFR_S_FILEKEY,WASTENAMEBUF);
2394 2507 2 MAIL$T_RD_RAB[RAB$L_R0P] = 0;
2395 2508 2 MAIL$T_RD_RAB[RAB$L_KBF] = WASTENAMEBUF;
2396 2509 2 MAIL$T_RD_RAB[RAB$B_KSZ] = MFR_S_FILEKEY;
2397 2510 2 MAIL$T_RD_RAB[RAB$B_KRF] = 1;
2398 2511 2 MAIL$T_RD_RAB[RAB$B_RAC] = RAB$C_KEY;
2399 2512 2 MAIL$T_RD_RAB[RAB$L_CTX] = MAIL$WRITEERR;
2400 2513 2
2401 2514 3 WHILE $GET(RAB=MAIL$T_RD_RAB)
2402 2515 3 DO BEGIN
2403 2516 3 RECBUF = .MAIL$T_RD_RAB[RAB$L_RBF];
2404 2517 3 RECBUF[MFR_B_FILEKEY[EN]] = .DEST_DESC[DSC$W_LENGTH];
2405 2518 3 CH$COPY(.DEST_DESC[DSC$W_LENGTH],.DEST_DESC[DSC$A_POINTER],0,
2406 2519 3 MFR_S_FILEKEY,RECBUF[MFR_T_FILEKEY]);
2407 2520 3 $UPDATE(RAB=MAIL$T_RD_RAB,ERR=UTIL$REPORT_IO_ERROR);
2408 2521 2 END;
2409 2522 2
2410 2523 2 RETURN TRUE
2411 2524 1 END;
```

			01FC 00000	.ENTRY	MAIL\$MOVE_WASTEBASKET, Save R2,R3,R4,R5,R6,-; 2489	
					R7,R8	
		58 00000000G	00 9E 00002	MOVAB	MAIL\$T_RD_RAB, R8	
		5E	28 C2 00009	SUBL2	#40, SP	
	DO	AF	00 FB 0000C	CALLS	#0, DESELECT IF WASTEBASKET	2504
		50 00000000'	00 D0 00010	MOVL	MAIL\$Q_WASTEDESC+4, R0	2505
27	00	60 00000000'	00 2C 00017	MOVCS	MAIL\$Q_WASTEDESC, (R0), #0, #39, -	
			6E 00020		WASTENAMEBUF	
		04	A8 D4 00021	CLRL	MAIL\$T_RD_RAB+4	2507
	30	A8	6E 9E 00024	MOVAB	WASTENAMEBUF, MAIL\$T_RD_RAB+48	2508
	34	A8 0127	8F B0 00028	MOVW	#295, MAIL\$T_RD_RAB+52	2509
	1E	A8	01 90 0002E	MOVW	#1, MAIL\$T_RD_RAB+30	2511
	18	A8 007E10D0	8F D0 00032	MOVL	#8261840, MAIL\$T_RD_RAB+24	2512
		57 04	AC D0 0003A	MOVL	DEST_DESC, R7	2518
			58 DD 0003E 1\$:	PUSHL	R8	2514
	00000000G 00		01 FB 00040	CALLS	#1, SYSSGET	

27	00	08 04	23 56 A6 B7	28 04 04 09	50 A8 BC BC A6 00 58 02 D1 01	E9 00047 D0 0004A 90 0004E 2C 00053 00 0005A 9F 0005C DD 00062 FB 00064 11 0006B D0 0006D 04 00070	2\$: BLBC R0, 2\$ MOVL MAIL\$T RD RAB+40, RECBUF MOVB @DEST_DEST, 8(RECBUF) MOVCS @DEST_DESC, @4(R7), #0, #39, 9(RECBUF) PUSHAB UTIL\$REPORT_IO_ERROR PUSHL R8 CALLS #2, SYSSUPDATE BRB 1\$ MOVL #1, R0 RET	2516 2517 2519 2520 2514 2523 2524
----	----	----------	----------------------	----------------------	--	--	---	--

; Routine Size: 113 bytes, Routine Base: \$CODE\$ + 1C32

```
2413 2525 1 ROUTINE UPDATE_DELCOUNT (MSGCOUNT) =
2414 2526 1 ++
2415 2527 1 FUNCTIONAL DESCRIPTION:
2416 2528 1
2417 2529 1 Update the count of deleted msg bytes in the fileinfo record
2418 2530 1
2419 2531 1 --
2420 2532 2 BEGIN
2421 2533 2 LOCAL
2422 2534 2 RECDISC : VECTOR[2, LONG],
2423 2535 2 RECPTR : REF $BBLOCK,
2424 2536 2 RECRFA : VECTOR[2, LONG],
2425 2537 2 TREC : $BBLOCK[$BYTEOFFSET(MFR_T_DATA)+4];
2426 2538 2
2427 2539 2 IF .MSGCOUNT EQL 0
2428 2540 2 THEN RETURN TRUE;
2429 2541 2
2430 2542 2 IF MAIL$READ_INFO_REC(MAIL$T_RD_RAB, MFR_C_FILEINFO, 1, RECDISC, RECRFA)
2431 2543 3 THEN BEGIN
2432 2544 3 RECPTR = .RECDISC[1];
2433 2545 3 (RECPTR[MFR_T_DATA])<0,32,0> = .(RECPTR[MFR_T_DATA])<0,32,0> + .MSGCOUNT;
2434 2546 3 MAIL$DELMSGCNT = .(RECPTR[MFR_T_DATA])<0,32,0>;
2435 2547 3 MAIL$UPDATE_INFO_REC(MAIL$T_RD_RAB, RECDISC, RECRFA);
2436 2548 3 END
2437 2549 3 ELSE BEGIN
2438 2550 3 (TREC[MFR_Q_DATIM])<0,32,0> = MFR_C_FILEINFO;
2439 2551 3 (TREC[MFR_Q_DATIM]+4) = 0;
2440 2552 3 CH$FILL(0, MFR_S_FILEKEY+1, TREC[MFR_B_FILKEYLEN]);
2441 2553 3 (TREC[MFR_T_DATA])<0,32,0> = .MSGCOUNT;
2442 2554 3 MAIL$DE[MSGCNT] = .MSGCOUNT;
2443 2555 3 RECDISC[0] = $BYTEOFFSET(MFR_T_DATA) + 4;
2444 2556 3 RECDISC[1] = TREC;
2445 2557 3 MAIL$UPDATE_INFO_REC(MAIL$T_RD_RAB, RECDISC)
2446 2558 2 END;
2447 2559 2
2448 2560 2 Check the threshold to determine if CONVERT/RECLAIM needs to be done
2449 2561 2
2450 2562 2 IF NOT .MAIL$GL_FLAGS[MAIF V RECL]
2451 2563 2 AND .MAIL$DELMSGCNT GTRU .MAIL$RECL_THRESHOLD
2452 2564 3 THEN BEGIN
2453 2565 3 SIGNAL(MAIL$RECLPLSWAIT);
2454 2566 3 MAIL$RECLAIMTO);
2455 2567 2 END;
2456 2568 2 RETURN TRUE
2457 2569 1 END;
```

03FC 00000 UPDATE_DELCOUNT:

59	00000000G	00	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9	: 2525
58	00000000'	00	9E	00009	MOVAB	MAIL\$T_RD_RAB, R9	:
57	ED95	CF	9E	00010	MOVAB	MAIL\$DELMSGCNT, R8	:
5E	BC	AE	9E	00015	MOVAB	MAIL\$UPDATE_INFO_REC, R7	:
56	04	AC	D0	00019	MOVL	-68(SP), SP	:
						MSGCOUNT, R6	: 2539

			70	13	0001D	BEQL	3\$		
		34	AE	9F	0001F	PUSHAB	RECRFA		2542
		40	AE	9F	00022	PUSHAB	RECDESC		
			01	DD	00025	PUSHL	#1		
			03	DD	00027	PUSHL	#3		
			59	DD	00029	PUSHL	R9		
	FF50	C7	05	FB	0002B	CALLS	#5, MAIL\$READ_INFO_REC		
		19	50	E9	00030	BLBC	R0, 1\$		
		50	40	AE	D0 00033	MOVL	RECDESC+4, RECPTR		2544
	30	A0	56	C0	00037	ADDL2	R6, 48(RECPTR)		2545
		68	30	A0	D0 0003B	MOVL	48(RECPTR), MAIL\$L_DEMSGCNT		2546
			34	AE	9F 0003F	PUSHAB	RECRFA		2547
			40	AE	9F 00042	PUSHAB	RECDESC		
			59	DD	00045	PUSHL	R9		
		67	03	FB	00047	CALLS	#3, MAIL\$UPDATE_INFO_REC		
			21	11	0004A	BRB	2\$		2542
		6E	03	7D	0004C	MOVQ	#3, TREC		2550
28		6E	00	2C	0004F	MOVCS	#0, (SP), #0, #40, TREC+8		2552
	00		08	AE	00054				
		30	AE	D0	00056	MOVL	R6, TREC+48		2553
		68	56	D0	0005A	MOVL	R6, MAIL\$L_DEMSGCNT		2554
		3C	AE	D0	0005D	MOVL	#52, RECDESC		2555
		40	AE	6E	9E 00061	MOVAB	TREC, RECDESC+4		2556
			3C	AE	9F 00065	PUSHAB	RECDESC		2557
			59	DD	00068	PUSHL	R9		
		67	02	FB	0006A	CALLS	#2, MAIL\$UPDATE_INFO_REC		
			00	95	0006D	TSTB	MAIL\$GL_FLAGS		2562
		00000000G	1A	19	00073	BLSS	3\$		
			68	D1	00075	CMPL	MAIL\$L_DEMSGCNT, MAIL\$L_RECL_THRESHOLD		2563
	04	A8	14	1B	00079	BLEQU	3\$		
			8F	DD	0007B	PUSHL	#8306355		2565
		007EBEB3	01	FB	00081	CALLS	#1, LIB\$SIGNAL		
	00000000G	00	7E	D4	00088	CLRL	-(SP)		2566
			01	FB	0008A	CALLS	#1, MAIL\$RECLAIM		
	0185	C7	01	D0	0008F	MOVL	#1, R0		2568
		50	01	D0	00092	RET			2569

; Routine Size: 147 bytes, Routine Base: \$CODE\$ + 1CA3

```
2459 2570 1 ROUTINE DELETE_EXTMSG(DATIM) : NOVALUE =
2460 2571 1 ++
2461 2572 1 FUNCTIONAL DESCRIPTION:
2462 2573 1
2463 2574 1 Delete an external message file
2464 2575 1
2465 2576 1 --
2466 2577 2 BEGIN
2467 2578 2
2468 2579 2 LOCAL
2469 2580 2     TMPFAB : $FAB_DECL,
2470 2581 2     TMPNAM : $NAM_DECL,
2471 2582 2     TMPXAB : $XABPRO_DECL,
2472 2583 2     NAMBUF : $BLOCK[NAM$_MAXRSS],
2473 2584 2     DESC : VECTOR[2,LONG];
2474 2585 2
2475 2586 2     DESC[0] = NAM$_MAXRSS;
2476 2587 2     DESC[1] = NAMBUF;
2477 2588 2     MAIL$FORM_EXTFILE(.DATIM,DESC);
2478 P 2589 2     $FAB_INIT(FAB=TMPFAB,
2479 P 2590 2         FNS=.DESC[0],
2480 P 2591 2         FNA=.DESC[1],
2481 P 2592 2         FAC=(GET,PUT),
2482 P 2593 2         CTX=MAIL$_FILNOTDEL,
2483 P 2594 2         XAB=TMPXAB,
2484 2595 2         NAM=TMPNAM);
2485 2596 2
2486 P 2597 2     $NAM_INIT(NAM=TMPNAM,
2487 P 2598 2         ESS=NAM$_MAXRSS,
2488 P 2599 2         ESA=NAMBUF,
2489 P 2600 2         RSS=NAM$_MAXRSS,
2490 2601 2         RSA=NAMBUF);
2491 2602 2
2492 P 2603 2     $XABPRO_INIT(XAB=TMPXAB,
2493 2604 2         NXT=0);
2494 2605 2
2495 P 2606 2     IF _ERR($OPEN(FAB=TMPFAB));
2496 2607 2         RETURN;
2497 2608 2
2498 2609 2     IF .TMPXAB[XAB$_UIC] NEQ .MAIL$_MYUIC
2499 2610 3     THEN BEGIN
2500 2611 3         $CLOSE(FAB=TMPFAB);
2501 2612 3         TMPFAB[FAB$_STS] = RMSS$_PRV;
2502 2613 3         TMPFAB[FAB$_STV] = SSS$_NOPRIV;
2503 2614 3         TMPFAB[FAB$_CTX] = MAIL$_CLOSEDEL;
2504 2615 3         UTIL$REPORT_IO_ERROR(TMPFAB);
2505 2616 3         RETURN;
2506 2617 2     END;
2507 2618 2
2508 2619 2     ! Close the file, changing the protection to allow delete access. Then
2509 2620 2     $ERASE the file.
2510 2621 2
2511 2622 2     TMPXAB[XAB$_UIC] = 0;
2512 2623 2     TMPXAB[XAB$_PRO] = DELETE_PRO;
2513 2624 2     $CLOSE(FAB=TMPFAB,ERR=UTIL$REPORT_IO_ERROR);
2514 2625 2     TMPFAB[FAB$_NAM] = 1;
2515 2626 2     $ERASE(FAB=TMPFAB,ERR=UTIL$REPORT_IO_ERROR);
```

!Need write access to file to fix prot

!Don't need to change uic

: 2516
: 2517

2627 2 RETURN;
2628 1 END;

.EXTRN SYSSERASE

00FC 00000 DELETE_EXTMSG:

.WORD Save R2,R3,R4,R5,R6,R7
MOVAB SYS\$CLOSE, R7
MOVAB UTIL\$REPORT_IO_ERROR, R6
MOVAB -524(SP), SP
MOVZBL #255, DESC
MOVAB NAMBUF, DESC+4
PUSHL SP
PUSHL DATIM
CALLS #2, MAIL\$FORM_EXTFILE
MOVCS #0, (SP), #0, #80, \$RMS_PTR
MOVW #20483, \$RMS_PTR
MOVB #3, \$RMS_PTR+22
MOVL #8262130, \$RMS_PTR+24
MOVB #2, \$RMS_PTR+31
MOVAB TMPXAB, \$RMS_PTR+36
MOVAB TMPNAM, \$RMS_PTR+40
MOVL DESC+4, \$RMS_PTR+44
MOVB DESC, \$RMS_PTR+52
MOVCS #0, (SP), #0, #96, \$RMS_PTR
MOVW #24578, \$RMS_PTR
MNEGB #1, \$RMS_PTR+2
MOVAB NAMBUF, \$RMS_PTR+4
MNEGB #1, \$RMS_PTR+10
MOVAB NAMBUF, \$RMS_PTR+12
MOVCS #0, (SP), #0, #88, \$RMS_PTR
MOVW #22547, \$RMS_PTR
PUSHAB TMPFAB
CALLS #1, SYSS\$OPEN
BLBC STATUS, 2\$
CML TMPXAB+12, MAIL\$MYUIC
BEQL 1\$
PUSHAB TMPFAB
CALLS #1, SYSS\$CLOSE
MOVL #98970, TMPFAB+8
MOVL #36, TMPFAB+12
MOVL #8262170, TMPFAB+24
PUSHAB TMPFAB
CALLS #1, UTIL\$REPORT_IO_ERROR
RET
CLRL TMPXAB+12
MOVW #-188, TMPXAB+8
PUSHL R6
PUSHAB TMPFAB
CALLS #2, SYSS\$CLOSE
BISB2 #1, TMPFAB+7
PUSHL R6

2570
2586
2587
2588
2595
2601
2604
2607
2609
2611
2612
2613
2614
2615
2610
2622
2623
2624
2625
2626

0050 8F 00 E429 CF 6E 00 00000000G 00 9E 00002
56 00000000G 00 9E 00009
5E FDF4 CE 9E 00010
7E FF 8F 9A 00015
04 AE 08 AE 9E 00019
04 AC DD 0001E
02 FB 00020
00 2C 00028
AD 0002F
8F B0 00031
03 90 00037
C6 AD 5003 8F D0 0003B
C8 AD 007E11F2 02 90 00043
CF AD FEF8 CD 9E 00047
D4 AD FF50 CD 9E 0004D
D8 AD 04 AE D0 00053
DC AD 6E 90 00058
E4 00 2C 0005C
CD 00063
FF50 CD 6002 8F B0 00066
FF52 CD 01 8E 0006D
FF54 CD 08 AE 9E 00072
FF5A CD 01 8E 00078
FF5C CD 08 AE 9E 0007D
00 2C 00083
CD 0008A
FEF8 CD 5813 8F B0 0008D
B0 AD 9F 00094
00000000G 00 01 FB 00097
4F 50 E9 0009E
00000000G 00 FF04 CD D1 000A1
21 13 000AA
AD 9F 000AC
67 B0 01 FB 000AF
B8 AD 0001829A 8F D0 000B2
BC AD 24 D0 000BA
C8 AD 007E121A 8F D0 000BE
B0 AD 9F 000C6
66 01 FB 000C9
04 000CC
FF04 CD D4 000CD 1\$:
FF00 CD FF44 8F B0 000D1
56 DD 000D8
AD 9F 000DA
67 02 FB 000DD
B7 AD 01 88 000E0
56 DD 000E4

MAIL\$ISAM_SUBS ISAM MAIL MESSAGE SUBROUTINES
V04-000

M 11
16-Sep-1984 00:54:01 VAX-11 Bliss-32 V4.0-742 Page 104
14-Sep-1984 12:42:20 DISK\$VMSMASTER:[MAIL.SRC]ISUBS.B32;1 (35)

00000000G 00 B0 AD 9F 000E6 PUSHAB TMPFAB
02 FB 000E9 CALLS #2, SY\$ERASE
04 000F0 2\$: RET ; 2628

; Routine Size: 241 bytes, Routine Base: \$CODE\$ + 1D36

; R

```
2519 2629 1 GLOBAL ROUTINE MAIL$EMPTY_WASTEBASKET =
2520 2630 1 ++
2521 2631 1 FUNCTIONAL DESCRIPTION:
2522 2632 1
2523 2633 1 This routine empties the wastebasket folder by deleting all the
2524 2634 1 messages it contains
2525 2635 1
2526 2636 1 --
2527 2637 2 BEGIN
2528 2638 2
2529 2639 2 LOCAL
2530 2640 2 PTR : REF $BBLOCK,
2531 2641 2 WASTENAMEBUF : $BBLOCK[MFR_S_FILEKEY],
2532 2642 2 KEYBUFFER : VECTOR[8,BYTE],
2533 2643 2 HDRFA : VECTOR[2, LONG],
2534 2644 2 MMH : REF $BBLOCK,
2535 2645 2 DELSPACE,
2536 2646 2 MSGCOUNT,
2537 2647 2 STATUS;
2538 2648 2
2539 2649 2 DESELECT_IF WASTEBASKET();
2540 2650 2 MSGCOUNT = 0;
2541 2651 2 DELSPACE = 0;
2542 2652 2 CH$COPY(.MAIL$Q_WASTEDESC[DSC$W_LENGTH],.MAIL$Q_WASTEDESC[DSC$A_POINTER],0,
2543 2653 2 MFR_S_FILEKEY,WASTENAMEBUF);
2544 2654 2
2545 2655 2 WHILE (
2546 2656 3 MAIL$T_RD_RAB[RAB$L_RFP] = 0;
2547 2657 3 MAIL$T_RD_RAB[RAB$L_KBF] = WASTENAMEBUF;
2548 2658 3 MAIL$T_RD_RAB[RAB$B_KSZ] = MFR_S_FILEKEY;
2549 2659 3 MAIL$T_RD_RAB[RAB$B_KRF] = 1;
2550 2660 3 MAIL$T_RD_RAB[RAB$B_RAC] = RAB$C_KEY;
2551 2661 4 $FIND(RAB=MAIL$T_RD_RAB)
2552 2662 3 )
2553 2663 3 DO BEGIN
2554 2664 3 MAIL$T_RD_RAB[RAB$B_RAC] = RAB$C_RFA;
2555 2665 3 MAIL$T_RD_RAB[RAB$L_KBF] = .MAIL$T_RD_RAB[RAB$L_RBF];
2556 2666 3 MAIL$T_RD_RAB[RAB$B_KSZ] = 8;
2557 2667 3 MAIL$T_RD_RAB[RAB$B_KRF] = 0;
2558 2668 3
2559 2669 3 Read the message header record
2560 2670 3
2561 2671 3 MAIL$T_RD_RAB[RAB$L_CTX] = MAIL$ READERR;
2562 2672 3 IF ERR$GET(RAB=MAIL$T_RD_RAB,ERR=UTIL$REPORT_IO_ERROR);,
2563 2673 3 UPDATE_DELCOUNT(.DELSpace);
2564 2674 3 RETURN .MSGCOUNT;
2565 2675 3
2566 2676 3 HDRFA[0] = .MAIL$T_RD_RAB[RAB$L_RFA0];
2567 2677 3 HDRFA[1] = .MAIL$T_RD_RAB[RAB$W_RFA4];
2568 2678 3 PTR = .MAIL$T_RD_RAB[RAB$L_RBF] + $BYTEOFFSET(MFR_T_DATA);
2569 2679 3 CH$MOVE(8,PTR[MMH_Q_DATID],KEYBUFFER);
2570 2680 3 MAIL$T_RD_RAB[RAB$L_KBF] = KEYBUFFER;
2571 2681 3 IF .PTR[MMH_V_EXTMSG]
2572 2682 3 THEN DELETE_EXTMSG(KEYBUFFER);
2573 2683 3 MAIL$T_RD_RAB[RAB$B_RAC] = RAB$C_KEY;
2574 2684 4 IF $FIND(RAB=MAIL$T_RD_RAB)
2575 2685 4 THEN BEGIN
2576 2685 4 MAIL$T_RD_RAB[RAB$L_RFP] = RAB$M_LIM;
```

```
2576 2686 4      |
2577 2687 4      | Read sequentially on the data records of the msg and delete them
2578 2688 4      |
2579 2689 4      | MAIL$RD_RAB[RAB$B_RAC] = RAB$C_SEQ;
2580 2690 5      | WHILE (STATUS = $GET(RAB=MAIL$RD_RAB))
2581 2691 5      |     AND (.STATUS NEQ RMSS_OK_LIM)
2582 2692 5      | DO BEGIN
2583 2693 5      |     DELSPACE = .DELSpace + .MAIL$RD_RAB[RAB$W_RSZ];
2584 2694 5      |     MAIL$RD_RAB[RAB$LCTX] = MAIL$WRITEERR;
2585 2695 5      |     $DELETE(RAB=MAIL$RD_RAB,ERR=UTIL$REPORT_IO_ERROR);
2586 2696 4      |     END;
2587 2697 3      | END;
2588 2698 3      |
2589 2699 3      | Delete the message header record
2590 2700 3      |
2591 2701 3      | MAIL$RD_RAB[RAB$L_RFA0] = .HDRFAC0;
2592 2702 3      | MAIL$RD_RAB[RAB$W_RFA4] = .HDRFAC1;
2593 2703 3      | MAIL$RD_RAB[RAB$B_RAC] = RAB$C_RFA;
2594 2704 3      | MAIL$RD_RAB[RAB$LCTX] = MAIL$READERR;
2595 2705 3      | IF_ERRT$GET(RAB=MAIL$RD_RAB,ERR=UTIL$REPORT_IO_ERROR);,
2596 2706 3      |     UPDATE_DELCOUNT(.DELSpace);
2597 2707 3      |     RETURN .MSGCOUNT;
2598 2708 3      | DELSPACE = .DELSpace + .MAIL$RD_RAB[RAB$W_RSZ];
2599 2709 3      | MMH = .MAIL$RD_RAB[RAB$L_RBF] + $BYTEOFFSET(MFR_T_DATA);
2600 2710 3      | IF .MMH[MMH_V_NEQMSG]
2601 2711 4      | THEN BEGIN
2602 2712 4      |     MMH[MMH_V_NEQMSG] = 0;
2603 2713 4      |     MAIL$DECR_NEWMAIL(MAIL$Q_USERNAME);
2604 2714 3      |     END;
2605 2715 3      | MAIL$RD_RAB[RAB$LCTX] = MAIL$WRITEERR;
2606 2716 3      | $DELETE(RAB=MAIL$RD_RAB,ERR=UTIL$REPORT_IO_ERROR);
2607 2717 3      | MSGCOUNT = .MSGCOUNT + 1;
2608 2718 3      | IF .MAIL$GL_FLAGS[MAIF_V_CTRLCLF]
2609 2719 3      | THEN EXITLOOP;
2610 2720 2      | END;
2611 2721 2      | UPDATE_DELCOUNT(.DELSpace);
2612 2722 2      | RETURN .MSGCOUNT
2613 2723 1      | END;
```

27

00

```
                                OFFC 00000
                                5B 00000000G 00 9E 00002
                                5E          38 C2 00009
FDDA CF          00 FB 0000C
                                58 D4 00011
                                56 D4 00013
                                50 00000000' 00 D0 00015
                                60 00000000' 00 2C 0001C
                                10 AE          00025
                                04 AB D4 00027 1$:
                                30 AB          10 AE 9E 0002A
                                34 AB          0127 8F B0 00C2F
                                1E AB          01 90 00035
```

.EXTRN SYSS\$DELETE

```
.ENTRY MAIL$EMPTY_WASTEBASKET, Save R2,R3,R4,R5,- : 2629
R6,R7,R8,R9,R10,R11
MOVAB MAIL$RD_RAB, R11
SUBL2 #56, SP
CALLS #0, DESELECT_IF_WASTEBASKET : 2649
CLRL MSGCOUNT : 2650
CLRL DELSPACE : 2651
MOVL MAIL$Q_WASTEDESC+4, R0 : 2652
MOVCS MAIL$Q_WASTEDESC, (R0), #0, #39, -
WASTENAMEBUF
CLRL MAIL$RD_RAB+4 : 2656
MOVAB WASTENAMEBUF, MAIL$RD_RAB+48 : 2657
MOVW #295, MAIL$RD_RAB+52 : 2658
MOVW #1, MAIL$RD_RAB+30 : 2660
```

			5B	DD	00039	PUSHL	R11	2661
			01	FB	0003B	CALLS	#1, SYSS\$FIND	
			50	E9	00042	BLBC	R0, 2\$	
	1E	AB	02	90	00045	MOVB	#2, MAIL\$T RD RAB+30	2664
	30	AB	08	DO	00049	MOVL	MAIL\$T RD RAB+40, MAIL\$T RD_RAB+48	2665
	34	AB	08	BO	0004E	MOVW	#8, MAIL\$T RD RAB+52	2666
	18	AB	8F	DO	00052	MOVL	#8261808, MAIL\$T RD RAB+24	2671
		007E10B0	00	9F	0005A	PUSHAB	UTIL\$REPORT_IO_ERROR	2674
		00000000G	5B	DD	00060	PUSHL	R11	
			02	FB	00062	CALLS	#2, SYSS\$GET	
			50	E8	00069	BLBS	STATUS, 3\$	
			00DF	31	0006C	BRW	8\$	
			AB	DO	0006F	MOVL	MAIL\$T RD_RAB+16, HDRFA	2675
	04	AE	14	AB	3C	MOVZWL	MAIL\$T RD_RAB+20, HDRFA+4	2676
08	57	28	30	C1	00078	ADDL3	#48, MAIL\$T RD RAB+40, PTR	2677
AE	08	A7	08	28	0007D	MOVCL3	#8, 8(PTR), KEYBUFFER	2678
	30	AB	08	9E	00083	MOVAB	KEYBUFFER, MAIL\$T RD_RAB+48	2679
08		67	03	E1	00088	BBC	#3, (PTR), 4\$	2680
			AE	9F	0008C	PUSHAB	KEYBUFFER	2681
	FE7B	CF	01	FB	0008F	CALLS	#1, DELETE_EXTMSG	
	1E	AB	01	90	00094	MOVB	#1, MAIL\$T RD_RAB+30	2682
			5B	DD	00098	PUSHL	R11	2683
			01	FB	0009A	CALLS	#1, SYSS\$FIND	
			50	E9	000A1	BLBC	R0, 6\$	
	04	AB	8F	3C	000A4	MOVZWL	#16384, MAIL\$T RD_RAB+4	2685
		4000	AB	94	000AA	CLRB	MAIL\$T RD_RAB+30	2689
		1E	5B	DD	000AD	PUSHL	R11	2690
			01	FB	000AF	CALLS	#1, SYSS\$GET	
			50	DO	000B6	MOVL	R0, STATUS	
			5A	E9	000B9	BLBC	STATUS, 6\$	
			5A	D1	000BC	CMPL	STATUS, #98385	2691
			20	13	000C3	BEQL	6\$	
			AB	3C	000C5	MOVZWL	MAIL\$T RD_RAB+34, R0	2693
			50	CO	000C9	ADDL2	R0, DE[SPACE	
	18	AB	8F	DO	000CC	MOVL	#8261840, MAIL\$T RD_RAB+24	2694
		007E10D0	00	9F	000D4	PUSHAB	UTIL\$REPORT_IO_ERROR	2695
		00000000G	5B	DD	000DA	PUSHL	R11	
			02	FB	000DC	CALLS	#2, SYSS\$DELETE	
			C8	11	000E3	BRB	5\$	2690
	10	AB	6E	DO	000E5	MOVL	HDRFA, MAIL\$T RD_RAB+16	2701
	14	AB	AE	BO	000E9	MOVW	HDRFA+4, MAIL\$T RD_RAB+20	2702
	1E	AB	02	90	000EE	MOVB	#2, MAIL\$T RD_RAB+30	2703
	18	AB	8F	DO	000F2	MOVL	#8261808, MAIL\$T RD_RAB+24	2704
		007E10B0	00	9F	000FA	PUSHAB	UTIL\$REPORT_IO_ERROR	2707
		00000000G	5B	DD	00100	PUSHL	R11	
			02	FB	00102	CALLS	#2, SYSS\$GET	
			50	E9	00109	BLBC	STATUS, 8\$	
			AB	3C	0010C	MOVZWL	MAIL\$T RD_RAB+34, R0	2708
			50	CO	00110	ADDL2	R0, DE[SPACE	
59	28	AB	30	C1	00113	ADDL3	#48, MAIL\$T RD_RAB+40, MMH	2709
			69	E9	00118	BLBC	(MMH), 7\$	2710
			01	8A	0011B	BICB2	#1, (MMH)	2712
			00	9F	0011E	PUSHAB	MAIL\$Q USERNAME	2713
		00000000G	01	FB	00124	CALLS	#1, MAIL\$DECR NEWMAIL	
			8F	DO	0012B	MOVL	#8261840, MAIL\$T RD_RAB+24	2715
		007E10D0	00	9F	00133	PUSHAB	UTIL\$REPORT_IO_ERROR	2716
		00000000G	5B	DD	00139	PUSHL	R11	

00000000G	00	02	FB	0013B	CALLS	#2, SYSS\$DELETE	:	
		58	D6	00142	INCL	MSGCOUNT	:	2717
	03 00000000G	00	E8	00144	BLBS	MAIL\$GL_FLAGS+1, 8\$	:	2718
		FED9	31	0014B	BRW	1\$	:	
		56	DD	0014E	PUSHL	DELSpace	:	2721
FD27	CF	01	FB	00150	CALLS	#1, UPDATE DELCOUNT	:	
	50	58	D0	00155	MOVL	MSGCOUNT, R0	:	2722
		04	00158	RET			:	2723

; Routine Size: 345 bytes, Routine Base: \$CODE\$ + 1E27


```
2615 2724 1 GLOBAL ROUTINE MAIL$CHECK_DATE(RAB,DATETIME,DEVDIRDESC) =
2616 2725 1 ++
2617 2726 1 FUNCTIONAL DESCRIPTION:
2618 2727 1
2619 2728 1 Check to see if the date/time specified is already used in the
2620 2729 1 file open on RAB. If so, bump DATETIME 512 times looking for
2621 2730 1 a free date/time. If none found, return error
2622 2731 1
2623 2732 1 DEVDIRDESC is an optional argument. If present, the external
2624 2733 1 file check is also made.
2625 2734 1 --
2626 2735 2 BEGIN
2627 2736 2
2628 2737 2 MAP
2629 2738 2 RAB : REF $BBLOCK,
2630 2739 2 DATETIME : REF VECTOR[,LONG];
2631 2740 2
2632 2741 2 BUILTIN
2633 2742 2 NULLPARAMETER;
2634 2743 2
2635 2744 2 LOCAL
2636 2745 2 FFLAG,
2637 2746 2 TMPFAB : $FAB DECL,
2638 2747 2 TMPDESC : VECTOR[2,LONG],
2639 2748 2 TMPBUF : $BBLOCK[NAM$C_MAXRSS];
2640 2749 2
2641 2750 2 FFLAG = NOT NULLPARAMETER(3);
2642 2751 2 TMPDESC[1] = TMPBUF;
2643 2752 2 RAB[RAB$B_RAC] = RAB$C_KEY;
2644 2753 2 RAB[RAB$L_ROP] = RAB$M_LOC OR RAB$M_NLK;
2645 2754 2 RAB[RAB$B_KRF] = 0;
2646 2755 2 RAB[RAB$B_KSZ] = 8;
2647 2756 2 RAB[RAB$L_KBF] = .DATETIME;
2648 2757 2 INCRU I FROM 0 TO 512
2649 2758 3 DO BEGIN
2650 2759 4 IF NOT $FIND(RAB=.RAB)
2651 2760 4 THEN BEGIN
2652 2761 4 IF NOT .FFLAG
2653 2762 4 THEN RETURN TRUE;
2654 2763 4 TMPDESC[0] = NAM$C_MAXRSS;
2655 2764 4 SYSS$FAO(MAIL$SD_EXTMSGNAM,TMPDESC,TMPDESC,.DEVDIRDESC,
2656 2765 4 .DATETIME[1],.DATETIME[0]);
2657 2766 4 $FAB_INIT(FAB=TMPFAB,
P 2767 4 FNS=.TMPDESC[0],
P 2768 4 FNA=.TMPDESC[1]);
2660 2769 4
2661 2770 5 IF NOT $OPEN(FAB=TMPFAB)
2662 2771 4 THEN RETURN TRUE;
2663 2772 4 $CLOSE(FAB=TMPFAB);
2664 2773 3 END;
2665 2774 3 DATETIME[0] = .DATETIME[0] + 1;
2666 2775 3 IF .DATETIME[0] EQL 0
2667 2776 3 THEN DATETIME[1] = .DATETIME[1] + 1;
2668 2777 2 END;
2669 2778 2
2670 2779 2 RETURN FALSE
2671 2780 1 END;
```

				03FC 00000	.ENTRY	MAIL\$CHECK_DATE, Save R2,R3,R4,R5,R6,R7,R8,-	
	5E	FEA8	CE	9E 00002	MOVAB	R9	2724
	03		6C	91 00007	CMPB	-344(SP), SP	
			05	1E 0000A	BGEQU	(AP), #3	2750
	56		01	D0 0000C	MOVL	1\$	
			09	11 0000F	BRB	#1, R6	
			56	D4 00011	CLRL	2\$	
		0C	AC	D5 00013	TSTL	R6	
			02	12 00016	BNEQ	12(AP)	
			56	D6 00018	INCL	2\$	
	56		56	D2 0001A	MCOML	R6, FFLAG	
AC	AD		6E	9E 0001D	MOVAB	TMPBUF, TMPDESC+4	2751
	58	04	AC	D0 00021	MOVL	RAB, R8	2752
1E	A8		01	90 00025	MOVB	#1, 30(R8)	
04	A8	00110000	8F	D0 00029	MOVL	#114112, 4(R8)	2753
34	A8		08	B0 00031	MOVW	#8, 52(R8)	2755
	57		AC	D0 00035	MOVL	DATETIME, R7	2756
30	A8	08	57	D0 00039	MOVL	R7, 48(R8)	
			59	D4 0003D	CLRL	I	2757
			58	DD 0003F	PUSHL	R8	2759
00000000G	00		01	FB 00041	CALLS	#1, SYSS\$FIND	
	5D		50	E8 00048	BLBS	R0, 6\$	
	4C		56	E9 0004B	BLBC	FFLAG, 4\$	2761
A8	AD	FF	8F	9A 0004E	MOVZBL	#255, TMPDESC	2763
			67	DD 00053	PUSHL	(R7)	2765
		04	A7	DD 00055	PUSHL	4(R7)	
		0C	AC	DD 00058	PUSHL	DEVDIRDESC	2764
		A8	AD	9F 0005B	PUSHAB	TMPDESC	
		A8	AD	9F 0005E	PUSHAB	TMPDESC	
		E06B	CF	9F 00061	PUSHAB	MAIL\$SD_EXTMSGNAM	
			06	FB 00065	CALLS	#6, SYSS\$FAO	
0050	8F	00	00	2C 0006C	MOVCS	#0, (SP), #0, #80, \$RMS_PTR	2768
			AD	00073			
	B0		8F	B0 00075	MOVW	#20483, \$RMS_PTR	
	C6	AD	02	90 0007B	MOVB	#2, \$RMS_PTR+22	
	CF	AD	02	90 0007F	MOVB	#2, \$RMS_PTR+31	
	DC	AD	AD	D0 00083	MOVL	TMPDESC+4, \$RMS_PTR+44	
	E4	AD	A8	90 00088	MOVB	TMPDESC, \$RMS_PTR+52	
		B0	AD	9F 0008D	PUSHAB	TMPFAB	2770
00000000G	00		01	FB 00090	CALLS	#1, SYSS\$OPEN	
	04		50	E8 00097	BLBS	R0, 5\$	
	50		01	D0 0009A	MOVL	#1, R0	2771
			04	0009D	RET		
		B0	AD	9F 0009E	PUSHAB	TMPFAB	2772
00000000G	00		01	FB 000A1	CALLS	#1, SYSS\$CLOSE	
			67	D6 000AB	INCL	(R7)	2774
			03	12 000AA	BNEQ	7\$	2775
		04	A7	D6 000AC	INCL	4(R7)	2776
			59	D6 000AF	INCL	I	2757
00000200	8F		59	D1 000B1	CMPL	I #512	
			85	1B 000B8	BLEQU	3\$	

MAI
V04

: 2779
: 2780

; Routine Size: 189 bytes, Routine Base: \$CODES + 1F80

SECRET

```
: 2673      2781 1 ROUTINE COPY_GETREC (OUTDESC) =
: 2674      2782 1 |++
: 2675      2783 1 | FUNCTIONAL DESCRIPTION:
: 2676      2784 1 |
: 2677      2785 1 |         Get next record of message
: 2678      2786 1 | --
: 2679      2787 2 BEGIN
: 2680      2788 2
: 2681      2789 2 MAP
: 2682      2790 2     OUTDESC : REF VECTOR[,LONG];
: 2683      2791 2
: 2684      2792 2 LOCAL
: 2685      2793 2     MORE_FLAG,
: 2686      2794 2     STATUS;
: 2687      2795 2
: 2688      2796 2 IF .MAIL$GL_FLAGS[MAIF_V_ISAM]
: 2689      2797 3 THEN BEGIN
: 2690      2798 3     STATUS = MAIL$GET_I_MSG(.OUTDESC,MORE_FLAG);
: 2691      2799 3     IF NOT .STATUS
: 2692      2800 3     THEN STATUS = RMSS_EOF;
: 2693      2801 3     END
: 2694      2802 3 ELSE BEGIN
: 2695      2803 4     IF (.STATUS = .MAIL$MSG_STS)
: 2696      2804 4     THEN (OUTDESC[0] = .MAIL$MSG_RSIZ;
: 2697      2805 4         OUTDESC[1] = .MAIL$MSG_RADR;
: 2698      2806 4         MAIL$GET_MSG())
: 2699      2807 3     ELSE STATUS = RMSS_EOF;
: 2700      2808 2     END;
: 2701      2809 2 RETURN .STATUS
: 2702      2810 1 END;
```

0004 00000 COPY_GETREC:						
	5E		04	C2 00002	.WORD	Save R2
	12	00000000G	00	E9 00005	SUBL2	#4, SP
			5E	DD 0000C	BLBC	MAIL\$GL_FLAGS, 1\$
		04	AC	DD 0000E	PUSHL	SP
F1A1	CF		02	FB 00011	PUSHL	OUTDESC
	52		50	D0 00016	CALLS	#2, MAIL\$GET_I_MSG
	2F		52	E8 00019	MOVL	R0, STATUS
			26	11 0001C	BLBS	STATUS, 3\$
	52	00000000G	00	D0 0001E 1\$:	BRB	2\$
	1C		52	E9 00025	MOVL	MAIL\$MSG_STS, STATUS
	50	04	AC	D0 00028	BLBC	STATUS, 2\$
	60	00000000G	00	3C 0002C	MOVL	OUTDESC, R0
04	A0	00000000G	00	D0 00033	MOVZWL	MAIL\$MSG_RSIZ, (R0)
00000000G	00		00	FB 0003B	MOVL	MAIL\$MSG_RADR, 4(R0)
			07	11 00042	CALLS	#0, MAIL\$GET_MSG
	52	0001827A	8F	D0 00044 2\$:	BRB	3\$
	50		52	D0 0004B 3\$:	MOVL	#98938, STATUS
			04	0004E	MOVL	STATUS, R0
					RET	

: Routine Size: 79 bytes, Routine Base: \$CODE\$ + 203D

MAIL\$ISAM_SUBS ISAM MAIL MESSAGE SUBROUTINES
V04-000

I 12
16-Sep-1984 00:54:01
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]ISUBS.B32;1 Page 113
(38)

MAI

```
2704 2811 1 ROUTINE COPY_MSG (FOLDER_DESC,OUTTAB)=
2705 2812 1 ++
2706 2813 1 FUNCTIONAL DESCRIPTION:
2707 2814 1
2708 2815 1 Copy one ISAM message to the message file
2709 2816 1
2710 2817 1 --
2711 2818 2 BEGIN
2712 2819 2
2713 2820 2 MAP
2714 2821 2 OUTTAB : REF $BBLOCK;
2715 2822 2
2716 2823 2 OWN
2717 2824 2
2718 2825 2 FLINK through MSGBLKS are the start of a CNCT block which is passed to
2719 2826 2 MAIL$WRITE_MSG. Note that MAIL$WRITE_MSG doesn't refer to anything else
2720 2827 2 in the CNCT block so this is a short one.
2721 2828 2
2722 2829 2 FLINK : VECTOR[2,LONG], !Note that this must parallel CNCTDEF
2723 2830 2 SENDER_DESC : DYN_DESC_DECL,
2724 2831 2 TO_DESC : DYN_DESC_DECL,
2725 2832 2 SUBJ_DESC : DYN_DESC_DECL,
2726 2833 2 CC_DESC : DYN_DESC_DECL, !Always null string
2727 2834 2 DUMMY_ADRIST : VECTOR[2,LONG],
2728 2835 2 DUMMY_LNKLIST : VECTOR[2,LONG],
2729 2836 2 DUMMY_NADDRS,
2730 2837 2 MSGBLKS,
2731 2838 2 DATE_DESC : DYN_DESC_DECL;
2732 2839 2
2733 2840 2 LOCAL
2734 2841 2 MSGFLAGS : $BBLOCK[4],
2735 2842 2 DATEBUF : $BBLOCK[32],
2736 2843 2 DATE_DESC2 : VECTOR[2,LONG],
2737 2844 2 DATETIME : VECTOR[2,LONG];
2738 2845 2
2739 2846 2 DATE_DESC2[0] = DATE_DESC2[1] = 0;
2740 2847 2 MSGFLAGS = MCR_M_USE_DATIM OR MCR_M_TELL_NEW;
2741 2848 2 IF .MAIL$GL_FLAGS[MAIF_V_ISAM]
2742 2849 3 THEN BEGIN
2743 P 2850 3 IF_ERR(MAIL$REWIND_I_MSG());,
2744 2851 3 RETURN .STATUS;
2745 2852 3 MAIL$B MSGTRAN = MMB_C_TEXT;
2746 2853 3 MAIL$INFO_I_MSG(0,SUBJ_DESC,0,0,SENDER_DESC,0,TO_DESC);
2747 2854 3 MSGBLKS = .MAIL$CURMSG[MMB_L_NBLKS];
2748 2855 3 CH$MOVE(8,MAIL$CURMSG[MMB_T_RDREC],DATETIME);
2749 2856 3 END
2750 2857 3 ELSE BEGIN
2751 P 2858 3 IF_ERR(MAIL$INFO_MSG(SENDER_DESC,SUBJ_DESC,DATE_DESC,TO_DESC,MSGBLKS);,
2752 2859 3 RETURN .STATUS);
2753 2860 3 MSGBLKS = (.MSGBLKS + 511)/512; !Convert bytes to blocks
2754 P 2861 3 IF_ERR(MAIL$REWIND_MSG());,
2755 2862 3 RETURN .STATUS;
2756 2863 3 IF MAIL$TEST_SEQ_NEW()
2757 2864 3 THEN MSGFLAGS[MCR_V_SND_NEWMSG] = TRUE;
2758 2865 4 INCRU I FROM 0 TO (IF MAIL$TEST_SUBJ() THEN 3 ELSE 2)
2759 2866 3 DO IF NOT MAIL$GET_MSG() THEN EXITLOOP;
2760 2867 3 DATE_DESC2[0] = 32;
```

```
: 2761      2868 3    DATE_DESC2[1] = DATEBUF;
: 2762      2869 3
: 2763      2870 3    ! Fill out date/time so that current defaults are not used
: 2764      2871 3
: 2765      2872 3    $FAO($DESCRIPTOR('!17AS:00.00'),DATE_DESC2,DATE_DESC2,DATE_DESC);
: 2766      2873 3    $BINTIM(TIMBUF=DATE_DESC2,TIMADR=DATETIME);
: 2767      2874 2    END;
: 2768      2875 2    IF NOT MAIL$CHECK_DATE(.OUTRAB,DATETIME)
: 2769      2876 3    THEN BEGIN
: 2770      2877 3        CH$MOVE(DSC$C_S_BLN,UTIL$GETFILENAME(.OUTRAB[RAB$S_L_FAB]),DATE_DESC2);
: 2771      2878 3        SIGNAL(MAIL$_DATIMUSED,2,DATETIME,DATE_DESC2);
: 2772      2879 3        RETURN MAIL$_DATIMUSED
: 2773      2880 2    END;
: 2774      2881 2    RETURN MAIL$WRITE_MSG(COPY_GETREC,FLINK,.FOLDER_DESC,
: 2775      2882 2        .MSGFLAGS,.OUTRAB,DATETIME)
: 2776      2883 2
: 2777      2884 1    END;
```

```
30 30 2E 30 30 3A 53 41 37 31 21 0208C P.ABM: .ASCII \!17AS:00.00\
                                02097 .BLKB 1
                                0000000B 02098 P.ABL: .LONG 11
                                00000000 0209C .ADDRESS P.ABM
                                .PSECT $OWNS$,NOEXE,2
                                000B4 FLINK: .BLKB 8
00# 000BC SENDER_DESC:
                                .BYTE 0[2]
02 0E 000BE .BYTE 14, 2
                                000C0 .BLKB 4
00# 000C4 TO_DESC: .BYTE 0[2]
02 0E 000C6 .BYTE 14, 2
                                000C8 .BLKB 4
00# 000CC SUBJ_DESC:
                                .BYTE 0[2]
02 0E 000CE .BYTE 14, 2
                                000D0 .BLKB 4
00# 000D4 CC_DESC: .BYTE 0[2]
02 0E 000D6 .BYTE 14, 2
                                000D8 .BLKB 4
                                000DC DUMMY_ADRLST:
                                .BLKB 8
                                000E4 DUMMY_LNKLST:
                                .BLKB 8
                                000EC DUMMY_NADDRS:
                                .BLKB 4
                                000F0 MSGBLKS: .BLKB 4
00# 000F4 DATE_DESC:
                                .BYTE 0[2]
02 0E 000F6 .BYTE 14, 2
                                000F8 .BLKB 4
                                .EXTRN SYSS$BINTIM
                                .PSECT $CODE$,NOWRT,2
```

```
03FC 00000 COPY_MSG:
59      FOFB      CF 9E 00002      .WORD      Save R2,R3,R4,R5,R6,R7,R8,R9      : 2811
58 00000000'    00 9E 00007      MOVAB      MAIL$REWIND_I_MSG, R9      :
5E      08      30 C2 0000E      MOVAB      MSGBLKS, R8      :
      08      AE 7C 00011      SUBL2      #48, SP      :
57      30      D0 00014      CLRQ      DATE_DESC2      : 2846
34 00000000G    00 E9 00017      MOVL      #48, MSGFLAGS      : 2847
69      00      FB 0001E      BLBC      MAIL$GL_FLAGS, 1$      : 2848
50      50      E9 00021      CALLS      #0, MAIL$REWIND_I_MSG      : 2851
00000000' 00      05 90 00024      BLBC      STATUS, 2$      :
      D4      A8 9F 0002B      MOVAB      #5, MAIL$B_MSGTRAN      : 2852
      7E      D4 0002E      PUSHAB     TO_DESC      : 2853
      CC      A8 9F 00030      CLRL      -(SP)      :
      DC      7E 7C 00033      PUSHAB     SENDER_DESC      :
      7E      A8 9F 00035      CLRQ      -(SP)      :
      7E      D4 00038      PUSHAB     SUBJ_DESC      :
038B      C9      07 FB 0003A      CLRL      -(SP)      :
50 00000000'    00 D0 0003F      CALLS      #7, MAIL$INFO_I_MSG      :
68      3F      A0 D0 00046      MOVL      MAIL$L_CURMSG, R0      : 2854
6E      47      A0 08 28 0004A      MOVL      63(R0), MSGBLKS      :
      0090      31 0004F      MOVCB      #8, 71(R0), DATETIME      : 2855
      D4      A8 9F 00054      BRW      10$      : 2848
      04      A8 9F 00057      PUSHL      R8      : 2859
      DC      A8 9F 0005A      PUSHAB     TO_DESC      :
      CC      A8 9F 0005D      PUSHAB     DATE_DESC      :
      05      FB 00060      PUSHAB     SUBJ_DESC      :
50000000G 00      05 FB 00060      PUSHAB     SENDER_DESC      :
17      50      E9 00067      CALLS      #5, MAIL$INFO_MSG      :
5000001FF 8F C1 0006A      BLBC      STATUS, 2$      :
68      50 00000200 8F C7 00072      ADDL3     #511, MSGBLKS, R0      : 2860
00000000G 00      00 FB 0007A      DIVL3     #512, R0, MSGBLKS      :
01      50      E8 00081      CALLS      #0, MAIL$REWIND_MSG      : 2862
      04 00084      BLBS      STATUS, 3$      :
00000000G 00      00 FB 00085      RET      :
03      50      E9 0008C      CALLS      #0, MAIL$TEST_SEQ_NEW      : 2863
57      01 88 0008F      BLBC      R0, 4$      :
00000000G 00      00 FB 00092      BISB2     #1, MSGFLAGS      : 2864
05      50      E9 00099      CALLS      #0, MAIL$TEST_SUBJ      : 2865
52      03      D0 0009C      BLBC      R0, 5$      :
      03      11 0009F      MOVL      #3, R2      :
      52      02 D0 000A1      BRB      6$      :
      53      D4 000A4      MOVL      #2, R2      :
      0C      11 000A6      CLRL      I      :
00000000G 00      00 FB 000A8      BRB      8$      :
07      50      E9 000AF      CALLS      #0, MAIL$GET_MSG      : 2866
      53      D6 000B2      BLBC      R0, 9$      :
      52      53      D1 000B4      INCL      I      :
      EF      1B 000B7      CMPL      I, R2      :
      08 AE      20 D0 000B9      BLEQU     7$      :
      0C AE      AE 9E 000BD      MOVL      #32, DATE_DESC2      : 2867
      10      AE 9F 000C2      MOVAB     DATEBUF, DATE_DESC2+4      : 2868
      04      AE 9F 000C5      PUSHAB     DATE_DESC      : 2872
      10      AE 9F 000C8      PUSHAB     DATE_DESC2      :
      FF29      CF 9F 000CB      PUSHAB     DATE_DESC2      :
00000000G 00      04 FB 000CF      PUSHAB     P.ABC      :
      04      FB 000CF      CALLS      #4, SYSS$FAO      :
```


			0C	5E DD 000D6	PUSHL SP	: 2873
				AE 9F 000D8	PUSHAB DATE_DESC2	
	00000000G	00		02 FB 000D8	CALLS #2, SYS\$BINTIM	
		56	08	5E DD 000E2 10\$:	PUSHL SP	: 2875
				AC D0 000E4	MOVL OUTRAB, R6	
				56 DD 000E8	PUSHL R6	
	ODDF	C9		02 FB 000EA	CALLS #2, MAIL\$CHECK_DATE	
		2C		50 E8 000EF	BLBS R0, 11\$	
			3C	A6 DD 000F2	PUSHL 60(R6)	: 2877
08	AE	00000000G	00	01 FB 000F5	CALLS #1, UTIL\$GETFILENAME	
			60	08 28 000FC	MOVC3 #8, (R0), DATE_DESC2	
			08	AE 9F 00101	PUSHAB DATE_DESC2	: 2878
			04	AE 9F 00104	PUSHAB DATETIME	
				02 DD 00107	PUSHL #2	
				8F DD 00109	PUSHL #8290466	
	00000000G	00	007E80A2	04 FB 0010F	CALLS #4, LIB\$SIGNAL	
		50	007E80A2	8F D0 00116	MOVL #8290466, R0	: 2879
				04 0011D	RET	
			4040	8F BB 0011E 11\$:	PUSHR #*M<R6,SP>	: 2882
				57 DD 00122	PUSHL MSGFLAGS	
			04	AC DD 00124	PUSHL FOLDER_DESC	: 2881
			C4	A8 9F 00127	PUSHAB FLINK	
			0E9C	C9 9F 0012A	PUSHAB COPY_GETREC	
	00000000G	00		06 FB 0012E	CALLS #6, MAIL\$WRITE_MSG	
				04 00135	RET	: 2884

; Routine Size: 310 bytes, Routine Base: \$CODE\$ + 20A0

```
2779 2885 1 GLOBAL ROUTINE MAIL$COPY_MSGS(FILE_DESC,FOLDER_DESC,COPY_FLAG,SIGNAL_RTN) =
2780 2886 1 ++
2781 2887 1 FUNCTIONAL DESCRIPTION:
2782 2888 1
2783 2889 1 This routine copies either messages from sequential file into
2784 2890 1 an ISAM file or from one ISAM message file to another
2785 2891 1
2786 2892 1 ALL_FLAG =
2787 2893 1 MAIL$V_ALL if to copy all current messages
2788 2894 1 MAIL$V_NIL if exclusive access needed
2789 2895 1 MAIL$V_DELMMSG if message should be deleted after copying
2790 2896 1 MAIL$V_CONFNEW if confirmation of new folder creation
2791 2897 1 --
2792 2898 2 BEGIN
2793 2899 2
2794 2900 2 MAP
2795 2901 2 FILE_DESC : REF $BBLOCK,
2796 2902 2 FOLDER_DESC : REF $BBLOCK,
2797 2903 2 COPY_FLAG : $BBLOCK;
2798 2904 2
2799 2905 2 BUILTIN
2800 2906 2 NULLPARAMETER;
2801 2907 2
2802 2908 2 LOCAL
2803 2909 2 IFAB : $FAB_DECL,
2804 2910 2 IRAB : $RAB_DECL,
2805 2911 2 INAM : $NAM_DECL,
2806 2912 2 RABPTR : REF $BBLOCK,
2807 2913 2 DESC : VECTOR[2, LONG],
2808 2914 2 NAMESS : $BBLOCK[NAM$C_MAXRSS],
2809 2915 2 FOLDERBUF : $BBLOCK[MFR_S_FILEKEY],
2810 2916 2 DELETE_FLAG,
2811 2917 2 SAVE_MSGNUM,
2812 2918 2 OPENIT,
2813 2919 2 EXCLUSIVE_ACCESS,
2814 2920 2 NEW_FILE,
2815 2921 2 STATUS;
2816 2922 2
2817 2923 2 RABPTR = IRAB;
2818 2924 2 $NAM_INIT(NAM=INAM,
P $RSS=NAM$C_MAXRSS,
P $RSA=NAMESS,
P $ESS=NAM$C_MAXRSS,
2928 2 $ESA=NAMESS);
2823 2929 2
2824 2930 2 IF .FOLDER_DESC[DSC$W_LENGTH] EQL 0
2825 2931 2 THEN STR$COPY_DX(.FOLDER_DESC,$DESCRIPTOR('MAIL'));
2826 2932 2
2827 2933 2 See if file specified. If not, and sequential file, set up default
2828 2934 2
2829 2935 2 NEW_FILE = 0;
2830 2936 2 OPENIT = 1;
2831 2937 2 IF .FILE_DESC[DSC$W_LENGTH] EQL 0
2832 2938 2 THEN BEGIN
2833 2939 3 IF NOT .MAIL$GL_FLAGS[MAIF_V_ISAM]
2834 2940 4 THEN BEGIN
2835 2941 4 STR$COPY_DX(.FILE_DESC,$DESCRIPTOR('MAIL'));
```



```
2836 2942 4      END
2837 2943 4      ELSE BEGIN
2838 2944 4          RABPTR = MAIL$RD_RAB;
2839 2945 4          OPENIT = 0;
2840 2946 3      END;
2841 2947 2      END;
2842 2948 2
2843 2949 2      Try to open existing file by specifying CIF
2844 2950 2
2845 2951 2      IF .OPENIT
2846 2952 3      THEN BEGIN
2847 2953 3          IFAB[IFAB$L CTX] = MAIL$ OPENOUT;
2848 2954 3          EXCLUSIVE_ACCESS = .COPY_FLAG[MCR_V NIL];
2849 2955 3          STATUS = MAIL$CREATE_ISAM(IFAB,.RABPTR,INAM,.FILE_DESC,0,
2850 2956 3              ?EXCLUSIVE_ACCESS OR MCR_M_CIF));
2851 2957 3      IF NOT .STATUS
2852 2958 4      THEN BEGIN
2853 2959 4          IF .STATUS NEQ RMSS_SHR
2854 2960 4              THEN RETURN UTI$REPORT_IO_ERROR(IFAB);
2855 2961 4          STATUS = MAIL$CREATE_ISAM(IFAB,.RABPTR,INAM,.FILE_DESC,0,
2856 2962 4              .EXCLUSIVE_ACCESS);
2857 2963 4      IF NOT .STATUS
2858 2964 4          THEN RETURN UTI$REPORT_IO_ERROR(IFAB);
2859 2965 4      NEW_FILE = 1;
2860 2966 3      END;
2861 2967 3      CH$MOVE(DSC$C_S_BLN,UTI$GETFILENAME(IFAB),DESC);
2862 2968 5      IF ((.IFAB[IFAB$L STS] EQL RMSS_CREATED)
2863 2969 4          OR .NEW_FILE)
2864 2970 3          AND NOT NUL[PARAMETER(4)
2865 2971 3          THEN (.SIGNAL_RTN)(DESC);
2866 2972 2      END;
2867 2973 2
2868 2974 2      File the messages
2869 2975 2
2870 2976 2      DELETE_FLAG = .COPY_FLAG[MCR_V_DELMSG];
2871 2977 2
2872 2978 2      If new folder confirmation requested, then see if the folder exists.
2873 2979 2
2874 2980 2      IF .COPY_FLAG[MCR_V_CONFNEW]
2875 2981 3      THEN BEGIN
2876 2982 3          CH$COPY(.FOLDER_DESC[DSC$W_LENGTH],.FOLDER_DESC[DSC$A_POINTER],
2877 2983 3              0,MFR_S_FILEKEY,FOLDERBUF);
2878 2984 3          RABPTR[RAB$B_RAC] = RAB$C_KEY;
2879 2985 3          RABPTR[RAB$L_ROP] = RAB$M_LOC OR RAB$M_NLK;
2880 2986 3          RABPTR[RAB$B_KRF] = 1;
2881 2987 3          RABPTR[RAB$B_KSZ] = MFR_S_FILEKEY;
2882 2988 3          RABPTR[RAB$L_KBF] = FOLDERBUF;
2883 2989 4          IF NOT $FIND(RAB=.RABPTR)
2884 2990 3              THEN IF NOT MAIL$CONFIRM(YN_DESC,
2885 2991 3                  MAIL$FAO_GETMSG(MAIL$_CRENEWFOLD,.FOLDER_DESC))
2886 2992 3                  THEN RETURN TRUE;
2887 2993 2      END;
2888 2994 2      IF .MAIL$GL_FLAGS[MAIF_V_ISAM]
2889 2995 3      THEN BEGIN
2890 2996 3          IF NOT .COPY_FLAG[MCR_V_ALL]
2891 2997 4          THEN (IF COPY_MSG(.FOLDER_DESC,.RABPTR)
2892 2998 4              AND .DELETE_FLAG
```

```
2893 2999 4 THEN MAIL$DELETE_I_MSG()  
2894 3000 4 ELSE BEGIN  
2895 3001 4 SAVE MSGNUM = .MAIL$ MSG_NUM;  
2896 3002 4 INCRD I FROM 1 TO .MAIL$NMSG  
2897 3003 5 DO BEGIN  
2898 3004 5 MAIL$ MSG_NUM = .I;  
2899 3005 5 IF (MAIL$CURMSG = .MAIL$MSGTBL[I-1]) NEQ 0 THEN  
2900 3006 5 IF COPY_MSG(.FOLDER_DESC,.RABPTR)  
2901 3007 5 AND .DELETE_FLAG  
2902 3008 5 THEN MAIL$DELETE_I_MSG();  
2903 3009 4 END;  
2904 3010 4 MAIL$ MSG_NUM = .SAVE_MSGNUM;  
2905 3011 4 IF .SAVE_MSGNUM GTU 0  
2906 3012 4 THEN MAIL$CURMSG = .MAIL$MSGTBL[.SAVE_MSGNUM-1];  
2907 3013 4 END  
2908 3014 3 END  
2909 3015 3 ELSE BEGIN  
2910 3016 3 IF NOT .COPY_FLAG[MCR_V ALL]  
2911 3017 3 THEN COPY_MSG(.FOLDER_DESC,.RABPTR)  
2912 3018 4 ELSE BEGIN  
2913 3019 4 IF NOT MAIL$FIND_MSG(1)  
2914 3020 5 THEN (SIGNALT.MAIL$MSG_STS);  
2915 3021 5 IF .OPENIT  
2916 3022 5 THEN $CLOSE(FAB=IFAB);  
2917 3023 4 RETURN .MAIL$MSG_STS;  
2918 3024 4 DO COPY_MSG(.FOLDER_DESC,.RABPTR)  
2919 3025 4 UNTIL NOT MAIL$FIND_MSG(0);  
2920 3026 4 END  
2921 3027 2 END;  
2922 3028 2  
2923 3029 2 IF .OPENIT  
2924 3030 2 THEN $CLOSE(FAB=IFAB);  
2925 3031 2  
2926 3032 2 RETURN TRUE  
2927 3033 1 END;
```

```
4C 49 41 4D 021D6 P.ABO: .ASCII \MAIL\  
021DA .BLKB 2  
00000004 021DC P.ABN: .LONG 4  
00000000' 021E0 .ADDRESS P.ABO  
4C 49 41 4D 021E4 P.ABQ: .ASCII \MAIL\  
00000004 021E8 P.ABP: .LONG 4  
00000000' 021EC .ADDRESS P.ABQ
```

0060	8F	00	FF0C	CD	6002	8F	B0	00022	.ENTRY MAIL\$COPY MSGS, Save R2,R3,R4,R5,R6,R7,R8,- R9,R10,R11 MOVAB MAIL\$MSGTBL, R11 MOVAB COPY_MSG, R10 MOVAB -548(SP), SP MOVAB IRAB, RABPTR MOVCS #0, (SP), #0, #96, \$RMS_PTR MOVW #24578, \$RMS_PTR	2885 2923 2928
			00	FF0C	00000000'	00	9E	00002		
			5A	FEA3	CF	9E	00009			
			5E	FDDC	CE	9E	0000E			
			56	FF6C	CD	9E	00013			
			6E		00	2C	00018			
				FF0C	CD		0001F			

	FF0E	CD	01	8E	00029	MNEGB	#1, SRMS_PTR+2	:	
	FF10	CD	28	AE	9E 0002E	MOVAB	NAMESS, SRMS_PTR+4	:	
	FF16	CD		01	8E 00034	MNEGB	#1, SRMS_PTR+10	:	
	FF18	CD	28	AE	9E 00039	MOVAB	NAMESS, SRMS_PTR+12	:	
		57	08	AC	D0 0003F	MOVL	FOLDER_DESC, R7	:	2930
				67	B5 00043	TSTW	(R7)	:	
				0C	12 00045	BNEQ	1\$	:	
			A2	AF	9F 00047	PUSHAB	P.ABN	:	2931
				57	DD 0004A	PUSHL	R7	:	
	00000000G	00		02	FB 0004C	CALLS	#2, STR\$COPY_DX	:	
		59		58	D4 00053	CLRL	NEW_FILE	:	2935
		52	04	01	D0 00055	MOVL	#1, OPENIT	:	2936
				AC	D0 00058	MOVL	FILE_DESC, R2	:	2937
				62	B5 0005C	TSTW	(R2)	:	
				1E	12 0005E	BNEQ	3\$	:	
		0E 00000000G	00	E8	00060	BLBS	MAIL\$GL_FLAGS, 2\$	:	2939
			8E	AF	9F 00067	PUSHAB	P.ABP	:	2941
				52	DD 0006A	PUSHL	R2	:	
	00000000G	00		02	FB 0006C	CALLS	#2, STR\$COPY_DX	:	
		56 00000000G	00	09	11 00073	BRB	3\$	:	2939
				00	9E 00075	MOVAB	MAIL\$T_RD_RAB, RABPTR	:	2944
				59	D4 0007C	CLRL	OPENIT	:	2945
		03		59	E8 0007E	BLBS	OPENIT, 4\$	:	2951
			008A	31	00081	BRW	9\$	:	
				8F	D0 00084	MOVL	#8261794, IFAB+24	:	2953
54				03	EF 0008C	EXTZV	#3, #1, COPY_FLAG, EXCLUSIVE_ACCESS	:	2954
				04	C9 00092	BISL3	#4, EXCLUSIVE_ACCESS, -(SP)	:	2956
				7E	D4 00096	CLRL	-(SP)	:	2955
				52	DD 00098	PUSHL	R2	:	
			FF0C	CD	9F 0009A	PUSHAB	INAM	:	
				56	DD 0009E	PUSHL	RABPTR	:	
			B0	AD	9F 000A0	PUSHAB	IFAB	:	
	E66C	CA		06	FB 000A3	CALLS	#6, MAIL\$CREATE_ISAM	:	
		53		50	D0 000A8	MOVL	R0, STAT'IS	:	
		31		53	E8 000AB	BLBS	STATUS, 7\$	:	2957
	000186B4	8F		53	D1 C00AE	CMPL	STATUS, #100020	:	2959
				1A	12 000B5	BNEQ	5\$	:	
				54	DD 000B7	PUSHL	EXCLUSIVE_ACCESS	:	2962
				7E	D4 000B9	CLRL	-(SP)	:	2961
				52	DD 000B8	PUSHL	R2	:	
			FF0C	CD	9F 000BD	PUSHAB	INAM	:	
				56	DD 000C1	PUSHL	RABPTR	:	
			B0	AD	9F 000C3	PUSHAB	IFAB	:	
	E66C	CA		06	FB 000C6	CALLS	#6, MAIL\$CREATE_ISAM	:	
		53		50	D0 000CB	MOVL	R0, STATUS	:	
		0B		53	E8 000CE	BLBS	STATUS, 6\$	:	2963
			B0	AD	9F 000D1	PUSHAB	IFAB	:	2964
	00000000G	00		01	FB 000D4	CALLS	#1, UTIL\$REPORT_IO_ERROR	:	
		58		04	000DB	RET		:	
				01	D0 000DC	MOVL	#1, NEW_FILE	:	2965
			B0	AD	9F 000DF	PUSHAB	IFAB	:	2967
				01	FB 000E2	CALLS	#1, UTIL\$GETFILENAME	:	
	00000000G	00		08	28 000E9	MOVC3	#8, (R0), DESC	:	
FF04		60		AD	D1 000EF	CMPL	IFAB+8, #67097	:	2968
	00010619	8F	B8	03	13 000F7	BEQL	8\$	:	
		12		58	E9 000F9	BLBC	NEW_FILE, 9\$	:	2969
		04		6C	91 000FC	CMPB	(APT), #4	:	2970

58	OC	AC	10	BC	FF04	0D	1F	000FF	BLSSU	9\$		
				01		AC	D5	00101	TSTL	16(AP)		
				08		08	13	00104	BEQL	9\$		
				01		CD	9F	00106	PUSHAB	DESC		2971
				07		01	FB	0010A	CALLS	#1, @SIGNAL RTN		
				07		07	EF	0010E	EXTZV	#7, #1, COPY_FLAG, DELETE_FLAG		2976
				07		AC	E9	00114	BLBC	COPY_FLAG+1, 10\$		2980
27		00	04	B7		67	2C	00118	MOVCS	(R7), @4(R7), #0, #39, FOLDERBUF		2982
						6E		0011E				
				1E		01	90	0011F	MOVB	#1, 30(RABPTR)		2984
				04		8F	D0	00123	MOVL	#114112, 4(RABPTR)		2985
				34		8F	B0	0012B	MOVW	#295, 52(RABPTR)		2987
				30		6E	9E	00131	MOVAB	FOLDERBUF, 48(RABPTR)		2988
						56	DD	00135	PUSHL	RABPTR		2989
						01	FB	00137	CALLS	#1, SYSS\$FIND		
				24		50	E8	0013E	BLBS	R0, 10\$		
						57	DD	00141	PUSHL	R7		2991
						8F	DD	00143	PUSHL	#8314680		
						02	FB	00149	CALLS	#2, MAIL\$FAO_GETMSG		
						50	DD	00150	PUSHL	R0		
						00	9F	00152	PUSHAB	YN_DESC		2990
						02	FB	00158	CALLS	#2, MAIL\$CONFIRM		
				03		50	E8	0015F	BLBS	R0, 10\$		
						00	D0	31	BRW	21\$		
				01		01	EF	00165	EXTZV	#1, #1, COPY_FLAG, R0		2996
				50		50	D2	0016B	MCOML	R0, R0		
				66		00	E9	0016E	BLBC	MAIL\$GL_FLAGS, 15\$		2994
				14		50	E9	00175	BLBC	R0, 11\$		2996
						56	DD	00178	PUSHL	RABPTR		2997
						57	DD	0017A	PUSHL	R7		
				6A		02	FB	0017C	CALLS	#2, COPY_MSG		
				63		50	E9	0017F	BLBC	R0, 16\$		
				60		58	E9	00182	BLBC	DELETE_FLAG, 16\$		2998
				F943		00	FB	00185	CALLS	#0, MAIL\$DELETE_I_MSG		2999
						59	11	0018A	BRB	16\$		2997
				53		00	D0	0018C	MOVL	MAIL\$MSG_NUM, SAVE_MSGNUM		3001
				54		AB	D0	00193	MOVL	MAIL\$NMSG, R4		3002
				52		01	D0	00197	MOVL	#1, I		
						26	11	0019A	BRB	14\$		
						52	D0	0019C	MOVL	I, MAIL\$MSG_NUM		3004
				00		6B	D0	001A3	MOVL	MAIL\$MSGTBL, R0		3005
				50		D0	D0	001A6	MOVL	-4(R0)[I], MAIL\$CURMSG		
				FC		12	13	001AC	BEQL	13\$		
						56	DD	001AE	PUSHL	RABPTR		3006
						57	DD	001B0	PUSHL			
				6A		02	FB	001B2	CALLS	# COPY_MSG		
				08		50	E9	001B5	BLBC	R0, 13\$		
				05		58	E9	001B8	BLBC	DELETE_FLAG, 13\$		3007
				F943		00	FB	001BB	CALLS	#0, MAIL\$DELETE_I_MSG		3008
						52	D6	001C0	INCL	I		3002
				54		52	D1	001C2	CMPL	I, R4		
						D5	1B	001C5	BLEQU	12\$		
						53	D0	001C7	MOVL	SAVE_MSGNUM, MAIL\$MSG_NUM		3010
				00		58	13	001CE	BEQL	20\$		3011
				50		6B	D0	001D0	MOVL	MAIL\$MSGTBL, R0		3012
				FC		D0	D0	001D3	MOVL	-4(R0)[SAVE_MSGNUM], MAIL\$CURMSG		
						4D	11	001D9	BRB	20\$		2995

	09		50	E9	001DB	15\$:	BLBC	R0, 17\$		3017
			56	DD	001DE		PUSHL	RABPTR		
			57	DD	001E0		PUSHL	R7		
	6A		02	FB	001E2		CALLS	#2, COPY_MSG		
			41	11	001E5	16\$:	BRB	20\$		
			01	DD	001E7	17\$:	PUSHL	#1		3019
00000000G	00		01	FB	001E9		CALLS	#1, MAIL\$FIND_MSG		
	22		50	E8	001F0		BLBS	R0, 19\$		
		00000000G	00	DD	001F3		PUSHL	MAIL\$ MSG_STS		3020
00000000G	00		01	FB	001F9		CALLS	#1, LIB\$SIGNAL		
	0A		59	E9	00200		BLBC	OPENIT, 18\$		3021
		B0	AD	9F	00203		PUSHAB	IFAB		3022
00000000G	00		01	FB	00206		CALLS	#1, SYS\$CLOSE		
	50	00000000G	00	D0	0020D	18\$:	MOVL	MAIL\$ MSG_STS, R0		3023
			04	00214		RET				
			56	DD	00215	19\$:	PUSHL	RABPTR		3024
			57	DD	00217		PUSHL	R7		
	6A		02	FB	00219		CALLS	#2, COPY_MSG		
			7E	D4	0021C		CLRL	-(SP)		3025
00000000G	00		01	FB	0021E		CALLS	#1, MAIL\$FIND_MSG		
	ED		50	E8	00225		BLBS	R0, 19\$		
	0A		59	E9	00228	20\$:	BLBC	OPENIT, 21\$		3029
		B0	AD	9F	0022B		PUSHAB	IFAB		3030
00000000G	00		01	FB	0022E		CALLS	#1, SYS\$CLOSE		
	50		01	D0	00235	21\$:	MOVL	#1, R0		3032
			04	00238		RET				3033

; Routine Size: 569 bytes, Routine Base: \$CODE\$ + 21F0

: 2929 3034 1 END
: 2930 3035 0 ELUDOM

.EXTRN LIB\$SIGNAL, SYSS\$UNWIND

PSECT SUMMARY

Name	Bytes	Attributes
\$GLOBALS	44	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$OWNS	252	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODES	9257	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	289	1	1000	00:01.4
_\$255\$DUA28:[MAIL.OBJ]MAILDEF.L32;1	457	85	18	26	00:00.4

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:ISUBS/OBJ=OBJ\$:ISUBS MSRC\$:ISUBS/UPDATE=(ENH\$:ISUBS)
: Size: 8883 code + 670 data bytes
: Run Time: 01:47.5
: Elapsed Time: 06:54.7
: Lines/CPU Min: 1693
: Lexemes/CPU-Min: 45905
: Memory Used: 413 pages
: Compilation Complete

0229 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

