

49

```

LL                               IIIII
LL                               IIIII
LL                               III
LL                               III
LL                               III
LL                               III
LL                               III
LL                               III
LL                               III
LL                               III
LL                               III
LL                               III
LL                               III
LL                               III
LL                               III
LLLLLLLLLLLLLLLL               IIIII
LLLLLLLLLLLLLLLL               IIIII

                                SSSSSSSS
                                SSSSSSSS
                                SS
                                SS
                                SS
                                SS
                                SSSSSS
                                SSSSSS
                                SS
                                SS
                                SS
                                SS
                                SSSSSSSS
                                SSSSSSSS

```

```
0001 0 MODULE MAIL$MAIL( %TITLE 'VAX/VMS PERSONAL MAIL UTILITY'
0002 0
0003 0      MAIN = MAIL,
0004 1      IDENT = 'V04-001' ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 *   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 *   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 *   ALL RIGHTS RESERVED.
0012 1 *
0013 1 *   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 *   ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 *   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 *   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 *   OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 *   TRANSFERRED.
0019 1 *
0020 1 *   THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 *   AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 *   CORPORATION.
0023 1 *
0024 1 *   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 *   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *****
0028 1
0029 1 ++
0030 1 FACILITY:      VAX/VMS MAIL UTILITY
0031 1
0032 1 ABSTRACT:      Entry point to MAIL. Dispatches according to type of
0033 1                  use: CLI command, interactive, or network slave job.
0034 1
0035 1                  This program was originally written as an exercise by the
0036 1                  author to learn VAX assembler coding. The first versions
0037 1                  were based on the "PNOTES" program written by Kim Mast for
0038 1                  the University of Illinois PLATO system.
0039 1
0040 1                  Two of the original goals of the design were simplicity of
0041 1                  use and simplicity of operation. The latter goal suggested
0042 1                  the use of the operating system's own indexed filing system,
0043 1                  namely file directories.
0044 1
0045 1                  Note also that this program does not purport to be a general
0046 1                  purpose "electronic mail system" or "office automation
0047 1                  system". It is simply a mail utility for normal users of
0048 1                  a VAX/VMS system.
0049 1
0050 1 ENVIRONMENT:    NATIVE/USER MODE
0051 1
0052 1 AUTHOR: Len Kawell, CREATION DATE: 20-FEB-1979
0053 1
0054 1 MODIFIED BY:
0055 1
0056 1      V04-001 BLS0252      Benn Schreiber 12-SEP-1984
0057 1      Do not drop NETMBX privilege.
```

58	0058	1	
59	0059	1	V03-027 ROP0021 Robert Posniak 12-JUL-1984
60	0060	1	If first character of MAIL\$EDIT is @
61	0061	1	then strip it. Change references of XYZ
62	0062	1	to TPU. Change SCR\$ERASE to SMG\$ERASE_PASTEBOARD
63	0063	1	
64	0064	1	V03-026 ROP0015 Robert Posniak 02-JUL-1984
65	0065	1	Do not concatenate the sender's username and
66	0066	1	nodename if the username string is null.
67	0067	1	
68	0068	1	V03-025 ROP0006 Robert Posniak 12-JUN-1984
69	0069	1	Clear proper bits in priv mask.
70	0070	1	
71	0071	1	V30-024 BLS0292 Benn Schreiber 8-APR-1984
72	0072	1	Implement SEND/ATTACH for Message Router
73	0073	1	
74	0074	1	V03-023 BLS0288 Benn Schreiber 21-MAR-1984
75	0075	1	Make raised priv. vector global
76	0076	1	
77	0077	1	V03-022 BLS0280 Benn Schreiber 4-MAR-1984
78	0078	1	Treat subject string of all blanks as no subject
79	0079	1	in network input.
80	0080	1	
81	0081	1	V03-021 BLS0272 Benn Schreiber 18-FEB-1984 13:34:05
82	0082	1	Implement server loop, enlarge SYS\$NET buffer.
83	0083	1	
84	0084	1	V03-020 BLS0263 Benn Schreiber 22-Jan-1984
85	0085	1	Complete default keypad. Disallow SPAWN if captive
86	0086	1	bit set. Send end-of-users flag immediately after
87	0087	1	sending all the user names to check.
88	0088	1	
89	0089	1	V03-019 BLS0255 Benn Schreiber 28-Dec-1983
90	0090	1	Convert trnlogs to trnlm. Convert getdev to
91	0091	1	getdviw. Use new global flags. Upcase to-list watching
92	0092	1	quotes. In slave mode, purge working set while waiting
93	0093	1	for first message record.
94	0094	1	
95	0095	1	V03-018 BLS0249 Benn Schreiber 8-Dec-1983
96	0096	1	Put back the deleting of MAIL.TMP files.
97	0097	1	
98	0098	1	V03-017 BLS0246 Benn Schreiber 8-Nov-1983
99	0099	1	Move new mail count to VMSMAIL.DAT. Modify prompt. Don't
100	0100	1	delete MAIL.TMP. Add support for cluster node enhancements.
101	0101	1	
102	0102	1	V03-016 BLS0240 Benn Schreiber 15-Sep-1983
103	0103	1	Complete alternate net protocol interfacing.
104	0104	1	
105	0105	1	V03-015 BLS0229 Benn Schreiber 16-Jul-1983
106	0106	1	Implement network block mode. Make pagesize
107	0107	1	very large if batch mode. Get new message count from
108	0108	1	mail file rather than SYSUAF.
109	0109	1	
110	0110	1	V03-014 BLS0227 Benn Schreiber 29-Jun-1983
111	0111	1	If file specified in REPLY command, disable cli defaulted
112	0112	1	/EXTRACT.
113	0113	1	
114	0114	1	V03-013 BLS0224 Benn Schreiber 1-Jun-1983

```
: 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 :
: 134      0134 1 :
: 135      0135 1 :
: 136      0136 1 :
: 137      0137 1 :
: 138      0138 1 :
: 139      0139 1 :
: 140      0140 1 :
: 141      0141 1 :
: 142      0142 1 :
: 143      0143 1 :
: 144      0144 1 :
: 145      0145 1 :
: 146      0146 1 :
: 147      0147 1 :
: 148      0148 1 :
: 149      0149 1 :
: 150      0150 1 :
: 151      0151 1 :
: 152      0152 1 :
: 153      0153 1 :
: 154      0154 1 :
: 155      0155 1 :
: 156      0156 1 :
: 157      0157 1 :
: 158      0158 1 :
: 159      0159 1 :
: 160      0160 1 :
: 161      0161 1 :
: 162      0162 1 :
: 163      0163 1 :
: 164      0164 1 :
: 165      0165 1 :
: 166      0166 1 :
: 167      0167 1 :
: 168      0168 1 :
: 169      0169 1 :
: 170      0170 1 :
: 171      0171 1 :

Correct argument to SMG$CREATE_VIRTUAL_KEYBOARD in cli_command

V03-012 BLS0222      Benn Schreiber  7-May-1983
Correct SEND for case of COPY_SELF SEND and
distribution list.

V03-011 BLS0220      Benn Schreiber  30-Apr-1983
Add support for SET AUTO, SET MAIL_DIRECTORY, keypad
input, and ERASE commands.

V03-010 BLS0218      Benn Schreiber  18-Apr-1983
Make default spec '.txt' in lookup_file

V03-009 BLS0217      Benn Schreiber  12-Apr-1983
Support SEND/SELF, etc.

V03-008 BLS0214      Benn Schreiber  27-Mar-1983
Add REPLY/EXTRACT. In cli_command, check addressees
first before getting message text if terminal input

V03-007 BLS0211      Benn Schreiber  13-Mar-1983
Extensive rework for isam mail files. Correct privilege
handling. Implement global CTRL/C handling. Lookup file
on SEND, REPLY before prompting for 'to'. Begin incorporation
of multi-threaded server

V03-006 BLS0197      Benn Schreiber  9-Dec-1982
Implement SET PERSONAL_NAME command. Break out SET, SHOW
commands and handling of VMSMAIL.DAT into separate files.

V03-005 BLS0193      Benn Schreiber  12-Nov-1982
Correct error handling in NETJOB reading message text. Collapse
GETJPI's so only one is done. Add exit handler to clean
up network links.

V03-004 BLS0189      Benn Schreiber  14-Oct-1982
Convert command parsing from TPARSE to CLISDCL_PARSE. Add a
few new features, such as show for/user=wilduser, search/file
and DELETE and READ command interlock on message.

V03-003 BLS0184      Benn Schreiber  20-Aug-1982
Delete MAIL.TMP files on entering mail in interactive mode.
This prevents inadvertently sending an old message left lying
around. Add SET FORW/USER= and SHOW FORW/USER=. Do not
put sender's node on his name if it's the same as our node.

V03-002 BLS0179      Benn Schreiber  26-Jul-1982
Complete automatic forwarding. Add check for infinite forward
loops.

V03-001 BLS0177      Benn Schreiber  20-JUN-1982
Add info on output device for listing size handling and
fix a few bugs. Add MAIL/EDIT DCL qualifier which implies
SEND/EDIT and REPLY/EDIT by default. Add support for automatic
forwarding, implemented by reading SYS$SYSTEM:VMSMAIL.DAT, an
ISAM file. Add SET [NO]FORWARD, and SHOW FORWARD commands.
Honor the DISMAIL flag in the UAF record.
```

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

: 172

0172 1 !--

L 13
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2 Page 4
(1)

MAI
V04


```
174 0173 1 |
175 0174 1 | TABLE OF CONTENTS:
176 0175 1 |
177 0176 1 |
178 0177 1 | INCLUDE FILES:
179 0178 1 |
180 0179 1 |     LIBRARY 'SYSS$LIBRARY:LIB';
181 0180 1 |     LIBRARY 'SYSS$LIBRARY:TPAMAC';
182 0181 1 |     REQUIRE 'SRC$MAILREQ';
183 0327 1 |     LIBRARY 'LIB$MAILDEF';
184 0328 1 |
185 0329 1 | FORWARD ROUTINE
186 0330 1 |     EXIT ROUTINE,
187 0331 1 |     MAIL$ENABLE_CTRL,
188 0332 1 |     CHECK_SEND,
189 0333 1 |     MAIL$GET_CLUNODES,
190 0334 1 |     GET_SEND_MSG,
191 0335 1 |     MAIL,
192 0336 1 |     CLI_COMMAND,
193 0337 1 |     NETJOB,
194 0338 1 |     INTJOB;
195 0339 1 |
196 0340 1 |
197 0341 1 | EQUATED SYMBOLS:
198 0342 1 |
199 0343 1 |
200 0344 1 | BUILTIN
201 0345 1 |     AP;
202 0346 1 |
203 0347 1 | LITERAL
204 0348 1 |     EDIT_FLAG = 0,
205 0349 1 |     LAST_FLAG = 1,
206 0350 1 |     FILE_FLAG = 2,
207 0351 1 |     EXIT_FLAG = 3,
208 0352 1 |     QUIT_FLAG = 4,
209 0353 1 |     F_SEND_EDIT = 0,
210 0354 1 |     F_REPLY_EDIT = 1,
211 0355 1 |     F_FORW_EDIT = 2;
212 0356 1 |
213 0357 1 | EXTERNAL REFERENCES:
214 0358 1 |
215 0359 1 | EXTERNAL LITERAL
216 0360 1 |     SMGS_EOF,
217 0361 1 |     CLIS_NEGATED,
218 0362 1 |     CLIS_LOCNEG;
219 0363 1 |
220 0364 1 | EXTERNAL ROUTINE
221 0365 1 |     CLISGET_VALUE,
222 0366 1 |     CLISDCL_PARSE,
223 0367 1 |     CLISDISPATCH,
224 0368 1 |     CLISPRESENT,
225 0369 1 |     LIB$CVT_DTB,
226 0370 1 |     LIB$DISABLE_CTRL,
227 0371 1 |     LIB$ENABLE_CTRL,
228 0372 1 |     LIB$GET_VM,
229 0373 1 |     LIB$SET_FILE_PROT,
230 0374 1 |     LIB$TPARSE,
```

clean up logical links
Enable for CTRL/C
Check if sending desired
Get cluster node names
Get and send message
Main entry point
CLI command processing
Network job processing
Interactive job processing

Argument pointer

/EDIT flag
/LAST flag
/FILE flag
EXIT command flag
QUIT command flag
Default SEND to SEND/EDIT
REPLY
FORWARD

end of file
Qualifier negated
Qualifier locally negated

Get value for qualifier
Parse command
Dispatch based on parse
Test if qualifier present
Convert decimal to binary
Disable CLI CTRL handling
Enable CLI CTRL handling
Allocate virtual memory
Change file protection
Table driven parser

231	0375	1	MAIL\$CMD_READ,	
232	0376	1	MAIL\$CMD_CONTINUE,	
233	0377	1	MAIL\$DEL_TMPF,	Delete old MAIL.TMP files
234	0378	1	MAIL\$DELEMLINK,	Delete links with no addressees
235	0379	1	MAIL\$EXTRACT_MESSAGE,	Copy message text to file
236	0380	1	MAIL\$FILTER_OUTPUT,	Turn escapes into '\$'
237	0381	1	MAIL\$FAO_GETMSG,	Get message text and fao it
238	0382	1	MAIL\$GETMSG,	Get message text
239	0383	1	MAIL\$GET_DEVDIR,	Get device and directory spec
240	0384	1	MAIL\$HANDLE,	Condition handler
241	0385	1	MAIL\$ADD_ADDR,	Add addressee to list
242	0386	1	MAIL\$CLOSE_FILE,	Close message file
243	0387	1	MAIL\$CONFIRM,	Confirm function
244	0388	1	MAIL\$DELETELINK,	Delete all logical links
245	0389	1	MAIL\$EDIT,	Edit a message
246	0390	1	MAIL\$ENTER,	Enter a message
247	0391	1	MAIL\$GET_INPUT,	Get from sys\$input
248	0392	1	MAIL\$GET_I_MSG,	Get record of current ISAMsg
249	0393	1	MAIL\$GET_MSG,	Get record of current message
250	0394	1	MAIL\$GET_NEWMAIL,	Get new mail count
251	0395	1	MAIL\$GET_USER,	Get uaf record
252	0396	1	MAIL\$INI_ADDR,	Initialize addressee list
253	0397	1	MAIL\$INFO_I_MSG,	Get ISAM message info
254	0398	1	MAIL\$INFO_MSG,	Get message information
255	0399	1	MAIL\$IOSUB_INIT,	Allocate mem for iosubs rtns
256	0400	1	MAIL\$NET_END_USERS,	Send end-of-users and to-list
257	0401	1	MAIL\$OPEN_IF,	Open default mail file
258	0402	1	MAIL\$OPEN_UAF,	Open system uaf
259	0403	1	MAIL\$OPEN_SYSMAL,	Open MAIL info file
260	0404	1	MAIL\$PUT_OUTPUT,	Put to sys\$output
261	0405	1	MAIL\$READ_FOREIGN_FILE,	Read input, foreign net
262	0406	1	MAIL\$REWIND_I_MSG,	Rewind current ISAM message
263	0407	1	MAIL\$REWIND_MSG,	Rewind current message
264	0408	1	MAIL\$SEND_ADDR,	Send message to addressee(s)
265	0409	1	MAIL\$SET_NAME,	Set full name string
266	0410	1	MAIL\$TEST_SUBJ,	Test if SUBJ: line present
267	0411	1	MAIL\$UPDATE_I_MSG,	Update message header record
268	0412	1	MAIL\$UPCASE,	Convert string to upper case
269	0413	1	MAIL\$UPCASE_Q,	Ditto, but not inside quotes
270	0414	1	SMG\$CREATE_PASTEBOARD,	
271	0415	1	SMG\$ERASE_PASTEBOARD,	
272	0416	1	SMG\$CREATE_VIRTUAL_KEYBOARD,	
273	0417	1	SMG\$CREATE_KEY_TABLE,	
274	0418	1	SMG\$DELETE_VIRTUAL_KEYBOARD,	
275	0419	1	SMG\$DEFINE_KEY,	
276	0420	1	STR\$APPEND,	String append
277	0421	1	STR\$CONCAT,	String copy and concatenate
278	0422	1	STR\$COPY_DX,	String copy
279	0423	1	STR\$FREE_DX,	Deallocate dynamic string
280	0424	1	STR\$TRIM,	Trim trailing blanks
281	0425	1	SYSSFAO,	Formatted ascii output
282	0426	1	UTIL\$GET_FILENAME,	Get filename from FAB/RAB
283	0427	1	UTIL\$REPORT_IO_ERROR;	Report I/O error
284	0428	1		
285	0429	1	EXTERNAL	
286	0430	1	MAIL\$T_RD_RAB : \$BBLOCK,	RAB for mail file i/o
287	0431	1	MAIL\$Q_MAIL_MAI : \$BBLOCK,	Descriptor for 'MAIL.MAI'


```
288 0432 1 MAIL$B_MSGTRAN : BYTE,      Next action for GET_I_MSG
289 0433 1 MAIL$COMMAND TABLE,      MAIL command tables
290 0434 1 MAIL$CURMSG : REF $BBLOCK,  Pointer to MMB for msg
291 0435 1 MAIL$USRSYSBUF : REF $BBLOCK, User's vmsmail record
292 0436 1 MAIL$MSG_NUM: WORD,        Current message number
293 0437 1 MAIL$MSG_RADR: LONG,       Current message record addr
294 0438 1 MAIL$MSG_RSIZ: WORD,       Current message record size
295 0439 1 MAIL$MSG_STS: LONG,       Current message status
296 0440 1 MAIL$Q_MSGNAME: BLOCK[,BYTE], Default message file name
297 0441 1 MAIL$Q_MSGTYPE: BLOCK[,BYTE], Default message file type
298 0442 1 MAIL$T_UAF_BUFF: REF $BBLOCK; Uaf record buffer
299 0443 1
300 0444 1 OWN
301 0445 1 EDITNAM_BUF: BLOCK[64,BYTE]; ! Edit logicalname buf
302 0446 1
303 0447 1 GLOBAL
304 0448 1 MAIL$GL_FLAGS : $BBLOCK[4],    MAIL control flags
305 0449 1 MAIL$GL_SMG_DISPLAY,          Display id from SMG
306 0450 1 MAIL$GL_SMG_KEYBOARD,        Keyboard id from SMG
307 0451 1 MAIL$GL_SMG_KEYTABLE,        Keytable id from SMG
308 0452 1 MAIL$GL_BIGBUFSIZ : LONG INITIAL(127), Size of big buffer
309 0453 1 MAIL$GL_BIGBUFFER : REF $BBLOCK, Address of buffer
310 0454 1 MAIL$GL_SYSFLAGS : $BBLOCK[4], system-wide control flags
311 0455 1 MAIL$GL_SIZLIM : LONG INITIAL(3), If > this, then write to file
312 0456 1 MAIL$GL_MYUIC,                UIC of user
313 0457 1 MAIL$GL_MYPID,                PID of user
314 0458 1 MAIL$Q_ATTDESC : DYN_DESC_DECL, Send/attach for mr
315 0459 1 MAIL$Q_CLUNODES : DYN_DESC_DECL, Descriptor of cluster nodes
316 0460 1 MAIL$Q_SYSPRVMSK: BLOCK[8,BYTE], Mask to enable/disable SYSPRV
317 0461 1 PRESET([PRV$V_SYSPRV] = 1),
318 0462 1 MAIL$Q_PROCPRV : $BBLOCK[8],    Process privileges
319 0463 1 MAIL$Q_INPTRAN: BLOCK[8,BYTE],  SYSS$INPUT translation
320 0464 1 MAIL$GL_INPCLASS : VOLATILE,    DVI$DEVCLASS for input
321 0465 1 MAIL$GL_INPDEVDP : VOLATILE,    DVI$DEVDEPEND2 for input
322 0466 1 MAIL$GL_OUTCLASS : VOLATILE,    DVI$DEVCLASS for output
323 0467 1 MAIL$GL_OUTDEVDP : VOLATILE,    DVI$DEVDEPEND2 for output
324 0468 1 MAIL$GL_OUTLINES,              Number of lines for output
325 0469 1 MAIL$Q_EDITNAM_DESC: VECTOR[2, LONG] Edit logical name desc
326 0470 1 INITIAL(LONG(64, EDITNAM_BUF)),
327 0471 1 MAIL$B_DFW : BYTE,              True if should use FOP=DFW
328 0472 1 MAIL$W_TTCHAN : WORD,          Channel for terminal
329 0473 1 MAIL$Q_USERNAME: BLOCK[8,BYTE], Username descriptor
330 0474 1 MAIL$G_CNCT : $BBLOCK[CNCT_C_LENGTH+RAB$C_BLN] !CNCT block
331 0475 1 PRESET([CNCT_B_FILRAT] = FAB$M_CR,
332 0476 1 [CNCT_B_FILRFM] = FAB$C_VAR),
333 0477 1 MAIL$Q_SYSNODE: $BBLOCK[DSC$C_S_BLN], Descriptor for SYSS$NODE
334 0478 1 translation
335 0479 1 MAIL$Q_SYSNODE2: $BBLOCK[DSC$C_S_BLN] Same thing with '::' on end
336 0480 1 PRESET([DSC$B_DTYPE] = DSC$K_DTYPE_T,
337 0481 1 [DSC$B_CLASS] = DSC$K_CLASS_D),
338 0482 1 MAIL$Q_CALEDTNAM : DYN_DESC_DECL, editor from 'callable_editn'
339 0483 1 MAIL$Q_PROTOCOL : DYN_DESC_DECL, descriptor of network protocol
340 0484 1 image file
341 0485 1 MAIL$Q_MAILDEF : DYN_DESC_DECL, dev/dir + 'MAIL.MAI'
342 0486 1 MAIL$Q_USERDEF: DYN_DESC_DECL; Descriptor of user's mail
343 0487 1 directory
344 0488 1 GLOBAL BIND
```

```
345 0489 1 MAIL$Q_TO = MAIL$G_CNCT[CNCT_Q_TODESC] : $BBLOCK, !"To" string
346 0490 1 MAIL$Q_SUBJ = MAIL$G_CNCT[CNCT_Q_SUBJDESC] : $BBLOCK, !"Subj" string
347 0491 1 MAIL$Q_FULLUSER = MAIL$G_CNCT[CNCT_Q_FROMDESC] : $BBLOCK, !and user
348 0492 1 MAIL$T_TMP_RAB = MAIL$G_CNCT[CNCT_T_RAB] : $BBLOCK;
349 0493 1
350 0494 1 BIND
351 0495 1     TMP_BUFF = MAIL$G_CNCT[CNCT_T_BUFFER] : $BBLOCK,
352 0496 1     TMP_RAB = MAIL$G_CNCT[CNCT_T_RAB] : $BBLOCK;
353 0497 1
354 0498 1 OWN
355 0499 1     CTRL_MASK,                ! Mask returned from dis_ctrl
356 0500 1     TPA_NUMBER,              ! Number parsed in tparse
357 0501 1     ENHANCED_PRIVS : VECTOR[2, LONG], ! Remember privs we added
358 0502 1     TMP_ESNAM_BUF : VECTOR[NAM$C_MAXRSS, BYTE], ! Temporary filename buffer
359 0503 1     CLI_FLAGS : BITVECTOR[32], ! CLI flags vector
360 0504 1     PROCSTS : LONG, ! Process status
361 0505 1     CLI_EDITQUAL : BITVECTOR[32], ! Flags from DCL MAIL/EDIT
362 0506 1     CLI_EXTRACT : BYTE, ! True if /EDIT=REPLY=EXTRACT
363 0507 1     CLI_SIGNALLED, ! True if CLI signalled parse error
364 0508 1     USERNAME : VECTOR[UAF$S_USERNAME, BYTE], ! Username buffer
365 0509 1     SYSNODEBUF : VECTOR[64, BYTE], ! SYS$NODE translation buffer
366 0510 1     TMP_NAM : $NAM_DECL, ! TMP file NAM block
367 0511 1     TMP_FAB : $FAB_DECL, ! and FAB
368 0512 1     TMP_XAB : $XABPRO_DECL,
369 0513 1     EXIT_STATUS,
370 0514 1     EXIT_BLOCK : VECTOR[4, LONG] ! For $DCLEXH
371 0515 1     INITIAL (0, EXIT_ROUTINE, 1, EXIT_STATUS),
372 0516 1     TMP_ESNAM_DESC : $BBLOCK[8] ! TMP filename descriptor
373 0517 1     PRESET([DSC$A_POINTER] = TMP_ESNAM_BUF),
374 0518 1     FILE_DESC : BLOCK[8, BYTE], ! Message file descriptor
375 0519 1     TMPFILE_DESC : DYN_DESC_DECL,
376 0520 1     TMPFILET_DESC : DYN_DESC_DECL,
377 0521 1     COMMAND_DESC : DYN_DESC_DECL,
378 0522 1     DYN_DESC : DYN_DESC_DECL,
379 0523 1     DYN_DESC_2 : DYN_DESC_DECL,
380 0524 1     INPUT_DESC : DYN_DESC_DECL;
381 0525 1
382 0526 1
383 0527 1 GLOBAL BIND
384 0528 1     MAIL$Q_ENHANCED_PRIVS = ENHANCED_PRIVS;
385 0529 1
386 0530 1 BIND
387 0531 1     NULL_CHAR = UPLIT('a'),
388 0532 1     COMMA_DESC = $DESCRIPTOR(',',),
389 0533 1     SLASH_DESC = $DESCRIPTOR('/',),
390 0534 1     SYSIN_DESC = $DESCRIPTOR('SYS$INPUT') : $BBLOCK,
391 0535 1     GETPROCSTS = UPLIT(WORD(4), ! Get process status $GETJPI
392 0536 1     WORD(JPI$STS), ! list
393 0537 1     LONG(PROCSTS),
394 0538 1     LONG(0),
395 0539 1     WORD(UAF$S_USERNAME), ! Get username list
396 0540 1     WORD(JPI$USERNAME),
397 0541 1     LONG(USERNAME),
398 0542 1     LONG(MAIL$Q_USERNAME),
399 0543 1     WORD(8), ! Process privs
400 0544 1     WORD(JPI$PROCPRIV),
401 0545 1     LONG(MAIL$Q_PROCPRIV),
```

```
402 0546 1 LONG(0),
403 0547 1 WORD(4), !Process PID
404 0548 1 WORD(JP($PID),
405 0549 1 LONG(MAIL$MYPID),
406 0550 1 LONG(0),
407 0551 1 LONG(0),
408 0552 1 TMPFILE_FAO_CDESC = $DESCRIPTOR('SYSS$SCRATCH:MAIL_!XL.TMP')! Temporary file name
409 0553 1 : BLOCK[BYTE]
410 0554 1 TMPFILE1_FAO_CDESC = $DESCRIPTOR('SYSS$SCRATCH:MAIL1!XL.TMP')
411 0555 1 : $BBLOCK ! Used for FOR/EDIT
412 0556 1 TMPFILE_FAO_DESC = $DESCRIPTOR('SYSS$SCRATCH:MAIL_!XW.TMP')! Temporary file name
413 0557 1 : BLOCK[BYTE]
414 0558 1 TMPFILE1_FAO_DESC = $DESCRIPTOR('SYSS$SCRATCH:MAIL1!XW.TMP')
415 0559 1 : $BBLOCK ! Used for FOR/EDIT
416 0560 1 CALLABLE_EDT = UPLIT (XASCII 'CALLABLE_') : VECTOR[BYTE],
417 0561 1 TOPIC_DESC = FILE_DESC; ! Help topic descriptor
418 0562 1
419 0563 1 GLOBAL BIND
420 0564 1 MAIL$Q_TMPFILE_DESC = TMPFILE_DESC; ! Make descriptor available
421 0565 1
422 0566 1 LITERAL
423 0567 1 PRO_SYS = $BITPOSITION(XAB$V_SYS),
424 0568 1 PRO_OWN = $BITPOSITION(XAB$V_OWN),
425 0569 1 PRO_GRP = $BITPOSITION(XAB$V_GRP),
426 0570 1 PRO_WLD = $BITPOSITION(XAB$V_WLD),
427 0571 1
428 0572 1 TMP_PRO = !Temp file protection
429 0573 1 ( XAB$M_NOEXE ) ^ PRO_SYS OR
430 0574 1 ( XAB$M_NOEXE ) ^ PRO_OWN OR
431 0575 1 (XAB$M_NOREAD OR XAB$M_NOWRITE OR XAB$M_NOEXE OR XAB$M_NODEL) ^ PRO_GRP OR
432 0576 1 (XAB$M_NOREAD OR XAB$M_NOWRITE OR XAB$M_NOEXE OR XAB$M_NODEL) ^ PRO_WLD;
433 0577 1
434 0578 1 GLOBAL LITERAL
435 0579 1 MAIL$C_TMP_PRO = TMP_PRO; !Make available globally
436 0580 1
437 0581 1 ! Declare static strings for qualifier names
438 0582 1
439 P 0583 1 SD(
440 P 0584 1 'ATTACHMENT',
441 P 0585 1 'HEADER',
442 P 0586 1 'EXTRACT',
443 P 0587 1 'FILE',
444 P 0588 1 'FILEM$',
445 P 0589 1 'EDT',
446 P 0590 1 'EDIT',
447 P 0591 1 'LNMS$SYSTEM',
448 P 0592 1 'LNMS$FILE_DEV',
449 P 0593 1 'MAIL$INIT',
450 P 0594 1 'MAIL$SYSTEM_FLAGS',
451 P 0595 1 'PROTOCOL',
452 P 0596 1 'REPLF',
453 P 0597 1 'SELF',
454 P 0598 1 'SEND1',
455 P 0599 1 'SUBJECT',
456 P 0600 1 'LAST',
457 P 0601 1 'NEXT',
458 P 0602 1 'TOLIST',
```

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

E 14
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2 Page 10
(2)

```

: 459      P 0603 1      'USER';
: 460      0604 1      'TPU');
: 461      0605 1
: 462      0606 1 GLOBAL BIND
: 463      0607 1      MAIL$SD_LNM_FILE_DEV = SD_LNM$FILE_DEV,
: 464      0608 1      MAIL$SD_LNM_SYSTEM = SD_LNM$SYSTEM,
: 465      0609 1      MAIL$SD_USER = SD_USER;
: 466      0610 1
: 467      0611 1      Define local names for shared messages
: 468      0612 1
: 469      P 0613 1 $SHR_MSGDEF(MAIL,126,LOCAL,
: 470      P 0614 1      (OPENIN,ERROR),
: 471      P 0615 1      (PARSEFAIL,WARNING),
: 472      P 0616 1      (OPENOUT,ERROR),
: 473      P 0617 1      (READERR,WARNING),
: 474      P 0618 1      (WRITEERR,WARNING),
: 475      P 0619 1      (CLOSEIN,WARNING),
: 476      0620 1      (CLOSEOUT,WARNING));
```

```

: 478      0621 1 |
: 479      0622 1 | TPARSE table to determine if a number is present as the command
: 480      0623 1 |
: 481      0624 1 $INIT_STATE (DECNUM_STATE,DECNUM_KEY);
: 482      0625 1 |
: 483      P 0626 1 $STATE (
: 484      0627 1   (TPAS_DECIMAL));
: 485      P 0628 1 $STATE (
: 486      0629 1   ('/',TPAS_EXIT),
: 487      0630 1   (TPAS_EOS,TPAS_EXIT));
: 488      0631 1 |
: 489      0632 1 | TPARSE table to parse /EDIT qualifier values
: 490      0633 1 |
: 491      0634 1 $INIT_STATE(EDITQUAL_STATE,EDITQUAL_KEY);
: 492      0635 1 |
: 493      P 0636 1 $STATE (
: 494      P 0637 1   ('SEND',TPAS_EXIT,,1^F_SEND_EDIT,CLI_EDITQUAL),
: 495      P 0638 1   ('REPLY',TPAS_EXIT,,1^F_REPLY_EDIT,CLI_EDITQUAL),
: 496      0639 1   ('FORWARD',TPAS_EXIT,,1^F_FORW_EDIT,CLI_EDITQUAL));
: 497      0640 1 |
: 498      0641 1 | TPARSE table to parse a number, either octal, decimal, or hex
: 499      0642 1 |
: 500      0643 1 $INIT_STATE(GETNUM_STATE,GETNUM_KEY);
: 501      P 0644 1 $STATE(
: 502      P 0645 1   (TPAS_DECIMAL,TPAS_EXIT,,,TPA_NUMBER),
: 503      0646 1   ('X',GET_OCTHEX));
: 504      P 0647 1 $STATE(GET_OCTHEX,
: 505      P 0648 1   ('O',GET_OCT),
: 506      0649 1   ('X',GET_HEX));
: 507      P 0650 1 $STATE(GET_OCT,
: 508      0651 1   (TPAS_OCTAL,TPAS_EXIT,,,TPA_NUMBER));
: 509      P 0652 1 $STATE(GET_HEX,
: 510      0653 1   (TPAS_HEX,TPAS_EXIT,,,TPA_NUMBER));
```

```
: 512 0654 1 GLOBAL ROUTINE MAIL$DISABLE_CTRLC : NOVALUE =  
: 513 0655 1 |++  
: 514 0656 1 | FUNCTIONAL DESCRIPTION:  
: 515 0657 1 |  
: 516 0658 1 | Disable CTRL/C ast  
: 517 0659 1 |  
: 518 0660 1 | --  
: 519 0661 2 BEGIN  
: 520 0662 2 MAIL$GL_FLAGS[MAIF V CTRLCFL] = 0;  
: 521 0663 2 IF .MAIL$W_ITCHAN NEQ 0  
: 522 0664 2 THEN $CANCEL(CHAN=.MAIL$W_ITCHAN);  
: 523 0665 2 RETURN;  
: 524 0666 1 END;
```

```
.TITLE MAIL$MAIL VAX/VMS PERSONAL MAIL UTILITY  
.IDENT \V04-001\
```

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

```
00 00 00 40 00000 P.AAA: .ASCII \a\<0><0><0>  
2C 00004 P.AAC: .ASCII \,\,  
00005 .BLKB 3  
00000001 00008 P.AAB: .LONG 1  
00000000' 0000C P.AAB: .ADDRESS P.AAC  
2F 00010 P.AAE: .ASCII \/\,  
00011 .BLKB 3  
00000001 00014 P.AAD: .LONG 1  
00000000' 00018 P.AAD: .ADDRESS P.AAE  
54 55 50 4E 49 24 53 59 53 0001C P.AAG: .ASCII \SYS$INPUT\  
00025 .BLKB 3  
00000009 00028 P.AAF: .LONG 9  
00000000' 0002C P.AAF: .ADDRESS P.AAG  
0004 00030 P.AAH: .WORD 4  
0305 00032 .WORD 773  
00000000' 00034 P.AAH: .ADDRESS PROCSTS  
00000000 00038 .LONG 0  
0020 0003C .WORD 32  
0202 0003E .WORD 514  
00000000' 00040 P.AAH: .ADDRESS USERNAME  
00000000' 00044 P.AAH: .ADDRESS MAIL$Q_USERNAME  
0008 00048 .WORD 8  
0204 0004A .WORD 516  
00000000' 0004C P.AAH: .ADDRESS MAIL$Q_PROCPRIV  
00000000 00050 .LONG 0  
0004 00054 .WORD 4  
0319 00056 .WORD 793  
00000000' 00058 P.AAH: .ADDRESS MAIL$L_MYPID  
00000000 0005C .LONG 0  
00000000 00060 .LONG 0  
49 41 4D 3A 48 43 54 41 52 43 53 24 53 59 53 00064 P.AAJ: .ASCII \SYS$SCRATCH:MAIL_!XL.TMP\  
50 4D 54 2E 4C 58 21 5F 4C 00073  
00000018 0007C P.AAI: .LONG 24  
00000000' 00080 P.AAI: .ADDRESS P.AAJ  
49 41 4D 3A 48 43 54 41 52 43 53 24 53 59 53 00084 P.AAL: .ASCII \SYS$SCRATCH:MAIL1_!XL.TMP\  
50 4D 54 2E 4C 58 21 5F 31 4C 00093  
0009D .BLKB 3
```


MA1
V04

Line	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	Col 9	Col 10	Col 11	Col 12	Col 13	Col 14	Col 15	Col 16	Col 17	Col 18	Col 19	Col 20	Col 21	Col 22	Col 23	Col 24	Col 25	Col 26	Col 27	Col 28	Col 29	Col 30	Col 31	Col 32	Col 33	Col 34	Col 35	Col 36	Col 37	Col 38	Col 39	Col 40	Col 41	Col 42	Col 43	Col 44	Col 45	Col 46	Col 47	Col 48	Col 49	Col 50	Col 51	Col 52	Col 53	Col 54	Col 55	Col 56	Col 57	Col 58	Col 59	Col 60	Col 61	Col 62	Col 63	Col 64	Col 65	Col 66	Col 67	Col 68	Col 69	Col 70	Col 71	Col 72	Col 73	Col 74	Col 75	Col 76	Col 77	Col 78	Col 79	Col 80	Col 81	Col 82	Col 83	Col 84	Col 85	Col 86	Col 87	Col 88	Col 89	Col 90	Col 91	Col 92	Col 93	Col 94	Col 95	Col 96	Col 97	Col 98	Col 99	Col 100	Col 101	Col 102	Col 103	Col 104	Col 105	Col 106	Col 107	Col 108	Col 109	Col 110	Col 111	Col 112	Col 113	Col 114	Col 115	Col 116	Col 117	Col 118	Col 119	Col 120	Col 121	Col 122	Col 123	Col 124	Col 125	Col 126	Col 127	Col 128	Col 129	Col 130	Col 131	Col 132	Col 133	Col 134	Col 135	Col 136	Col 137	Col 138	Col 139	Col 140	Col 141	Col 142	Col 143	Col 144	Col 145	Col 146	Col 147	Col 148	Col 149	Col 150	Col 151	Col 152	Col 153	Col 154	Col 155	Col 156	Col 157	Col 158	Col 159	Col 160	Col 161	Col 162	Col 163	Col 164	Col 165	Col 166	Col 167	Col 168	Col 169	Col 170	Col 171	Col 172	Col 173	Col 174	Col 175	Col 176	Col 177	Col 178	Col 179	Col 180	Col 181	Col 182	Col 183	Col 184	Col 185	Col 186	Col 187	Col 188	Col 189	Col 190	Col 191	Col 192	Col 193	Col 194	Col 195	Col 196	Col 197	Col 198	Col 199	Col 200	Col 201	Col 202	Col 203	Col 204	Col 205	Col 206	Col 207	Col 208	Col 209	Col 210	Col 211	Col 212	Col 213	Col 214	Col 215	Col 216	Col 217	Col 218	Col 219	Col 220	Col 221	Col 222	Col 223	Col 224	Col 225	Col 226	Col 227	Col 228	Col 229	Col 230	Col 231	Col 232	Col 233	Col 234	Col 235	Col 236	Col 237	Col 238	Col 239	Col 240	Col 241	Col 242	Col 243	Col 244	Col 245	Col 246	Col 247	Col 248	Col 249	Col 250	Col 251	Col 252	Col 253	Col 254	Col 255	Col 256	Col 257	Col 258	Col 259	Col 260	Col 261	Col 262	Col 263	Col 264	Col 265	Col 266	Col 267	Col 268	Col 269	Col 270	Col 271	Col 272	Col 273	Col 274	Col 275	Col 276	Col 277	Col 278	Col 279	Col 280	Col 281	Col 282	Col 283	Col 284	Col 285	Col 286	Col 287	Col 288	Col 289	Col 290	Col 291	Col 292	Col 293	Col 294	Col 295	Col 296	Col 297	Col 298	Col 299	Col 300	Col 301	Col 302	Col 303	Col 304	Col 305	Col 306	Col 307	Col 308	Col 309	Col 310	Col 311	Col 312	Col 313	Col 314	Col 315	Col 316	Col 317	Col 318	Col 319	Col 320	Col 321	Col 322	Col 323	Col 324	Col 325	Col 326	Col 327	Col 328	Col 329	Col 330	Col 331	Col 332	Col 333	Col 334	Col 335	Col 336	Col 337	Col 338	Col 339	Col 340	Col 341	Col 342	Col 343	Col 344	Col 345	Col 346	Col 347	Col 348	Col 349	Col 350	Col 351	Col 352	Col 353	Col 354	Col 355	Col 356	Col 357	Col 358	Col 359	Col 360	Col 361	Col 362	Col 363	Col 364	Col 365	Col 366	Col 367	Col 368	Col 369	Col 370	Col 371	Col 372	Col 373	Col 374	Col 375	Col 376	Col 377	Col 378	Col 379	Col 380	Col 3
------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	-------

```
46 4C 50 45 52 001C8 P.ABQ: .ASCII \REPLF\
001CD .BLKB 3
00000005 001D0 P.ABP: .LONG 5
00000000 001D4 .ADDRESS P.ABQ
46 4C 45 53 001D8 P.ABS: .ASCII \SELF\
00000004 001DC P.ABR: .LONG 4
00000000 001E0 .ADDRESS P.ABS
31 44 4E 45 53 001E4 P.ABU: .ASCII \SEND1\
001E9 .BLKB 3
00000005 001EC P.ABT: .LONG 5
00000000 001F0 .ADDRESS P.ABU
54 43 45 4A 42 55 53 001F4 P.ABW: .ASCII \SUBJECT\
001FB .BLKB 1
00000007 001FC P.ABV: .LONG 7
00000000 00200 .ADDRESS P.ABW
54 53 41 4C 00204 P.ABY: .ASCII \LAST\
00000004 00208 P.ABX: .LONG 4
00000000 0020C .ADDRESS P.ABY
54 58 45 4E 00210 P.ACA: .ASCII \NEXT\
00000004 00214 P.ABZ: .LONG 4
00000000 00218 .ADDRESS P.ACA
54 53 49 4C 4F 54 0021C P.ACC: .ASCII \TOLIST\
00222 .BLKB 2
00000006 00224 P.ACB: .LONG 6
00000000 00228 .ADDRESS P.ACC
52 45 53 55 0022C P.ACE: .ASCII \USER\
00000004 00230 P.ACD: .LONG 4
00000000 00234 .ADDRESS P.ACE
55 50 54 00238 P.ACG: .ASCII \TPU\
0023B .BLKB 1
00000003 0023C P.ACF: .LONG 3
00000000 00240 .ADDRESS P.ACG

.PSECT _LIB$KEY1$,NOWRT, SHR, PIC,1

00000 ;TPASKEYSTO
U.8: .BLKB 0
44 4E 45 53 00000 ;TPASKEYST
U.10: .ASCII \SEND\
FF 00004 .BYTE -1
00005 ;TPASKEYSTO
U.15: .BLKB 0
59 4C 50 45 52 00005 ;TPASKEYST
U.17: .ASCII \REPLY\
FF 0000A .BYTE -1
0000B ;TPASKEYSTO
U.22: .BLKB 0
44 52 41 57 52 4F 46 0000B ;TPASKEYST
U.24: .ASCII \FORWARD\
FF 00012 .BYTE -1
FF 00013 ;TPASKEYFILL
U.29: .BYTE -1

.PSECT _LIB$STATES$,NOWRT, SHR, PIC,1

00000 DECNUM_STATE::
.BLKB 0
```



```
05F3 00000 ;TPASTYPE
      U.2: .WORD 1523
102F 00002 ;TPASTYPE
      U.3: .WORD 4143
FFFF 00004 ;TPASTARGET
      U.4: .WORD -1
15F7 00006 ;TPASTYPE
      U.5: .WORD 5623
FFFF 00008 ;TPASTARGET
      U.6: .WORD -1
      0000A .BLKB 2
      0000C EDITQUAL_STATE::
      .BLKB 0
7100 0000C ;TPASTYPE
      U.11: .WORD 28928
00000000* 0000E ;TPASADDR
      U.12: .LONG <<CLI_EDITQUAL-U.12>-4>
00000001 00012 ;TPASMASK
      U.13: .LONG 1
FFFF 00016 ;TPASTARGET
      U.14: .WORD -1
7101 00018 ;TPASTYPE
      U.18: .WORD 28929
00000000* 0001A ;TPASADDR
      U.19: .LONG <<CLI_EDITQUAL-U.19>-4>
00000002 0001E ;TPASMASK
      U.20: .LONG 2
FFFF 00022 ;TPASTARGET
      U.21: .WORD -1
7502 00024 ;TPASTYPE
      U.25: .WORD 29954
00000000* 00026 ;TPASADDR
      U.26: .LONG <<CLI_EDITQUAL-U.26>-4>
00000004 0002A ;TPASMASK
      U.27: .LONG 4
FFFF 0002E ;TPASTARGET
      U.28: .WORD -1
      00030 GETNUM_STATE::
      .BLKB 0
51F3 00030 ;TPASTYPE
      U.31: .WORD 20979
00000000* 00032 ;TPASADDR
      U.32: .LONG <<TPA_NUMBER-U.32>-4>
FFFF 00036 ;TPASTARGET
      U.33: .WORD -1
1425 00038 ;TPASTYPE
      U.34: .WORD 5157
0000* 0003A ;TPASTARGET
      U.36: .WORD <<U.35-U.36>-2>
      0003C GET_OCTHEX
      U.35: .BLKB 0
104F 0003C ;TPASTYPE
      U.37: .WORD 4175
0000* 0003E ;TPASTARGET
      U.39: .WORD <<U.38-U.39>-2>
1458 00040 ;TPASTYPE
      U.40: .WORD 5208
```

```
0000* 00042 :TPASTARGET
          U.42: .WORD    <<U.41-U.42>-2>
          00044 :GET_OCT
          U.38: .BLKB    0
          55F4 00044 :TPASTYPE
          U.43: .WORD    22004
          00000000* 00046 :TPAS$ADDR
          U.44: .LONG    <<TPA_NUMBER-U.44>-4>
          FFFF 0004A :TPASTARGET
          U.45: .WORD    -1
          0004C :GET_HEX
          U.41: .BLKB    0
          55F5 0004C :TPASTYPE
          U.46: .WORD    22005
          00000000* 0004E :TPAS$ADDR
          U.47: .LONG    <<TPA_NUMBER-U.47>-4>
          FFFF 00052 :TPASTARGET
          U.48: .WORD    -1

          .PSECT _LIB$KEY0$,NOWRT, SHR, PIC,1

          00000 DECNUM_KEY::
          .BLKB 0
          00000 :TPAS$KEY0
          U.1: .BLKB 0
          00000 EDITQUAL_KEY::
          .BLKB 0
          00000 :TPAS$KEY0
          U.7: .BLKB 0
          0000* 00000 :TPAS$KEY
          U.9: .WORD    <U.8-U.7>
          0000* 00002 :TPAS$KEY
          U.16: .WORD    <U.15-U.7>
          0000* 00004 :TPAS$KEY
          U.23: .WORD    <U.22-U.7>
          00006 .BLKB 2
          00008 GETNUM_KEY::
          .BLKB 0
          00008 :TPAS$KEY0
          U.30: .BLKB 0

          .PSECT $OWNS$,NOEXE,2

          00000 EDITNAM_BUF:
          .BLKB 64
          00040 CTRL_MASK:
          .BLKB 4
          00044 TPA_NUMBER:
          .BLKB 4
          00048 ENHANCED_PRIVS:
          .BLKB 8
          00050 TMP_ESNAM_BUF:
          .BLKB 255
          0014F .BLKB 1
          00150 CLIFLAGS:
          .BLKB 4
          00154 PROCSTS:.BLKB 4
```

```
00158 CLI_EDITQUAL:
      .BLKB 4
0015C CLI_EXTRACT:
      .BLKB 1
0015D      .BLKB 3
00160 CLI_SIGNALLED:
      .BLKB 4
00164 USERNAME:
      .BLKB 32
00184 SYSNODEBUF:
      .BLKB 64
001C4 TMP_NAM: .BLKB 96
00224 TMP_FAB: .BLKB 80
00274 TMP_XAB: .BLKB 88
002CC EXIT_STATUS:
      .BLKB 4
00000000 002D0 EXIT_BLOCK:
      .LONG 0
00000000V 002D4      .ADDRESS EXIT_ROUTINE
00000001 002D8      .LONG 1
00000000' 002DC      .ADDRESS EXIT_STATUS
00# 002E0 TMP_ESNAM_DESC:
      .BYTE 0[4]
00000000' 002E4      .ADDRESS TMP_ESNAM_BUF
002E8 FILE_DESC:
      .BLKB 8
00# 002F0 TMPFILE_DESC:
      .BYTE 0[2]
02 0E 002F2      .BYTE 14, 2
002F4      .BLKB 4
00# 002F8 TMPFILE1_DESC:
      .BYTE 0[2]
02 0E 002FA      .BYTE 14, 2
002FC      .BLKB 4
00# 00300 COMMAND_DESC:
      .BYTE 0[2]
02 0E 00302      .BYTE 14, 2
00304      .BLKB 4
00# 00308 DYN_DESC:
      .BYTE 0[2]
02 0E 0030A      .BYTE 14, 2
0030C      .BLKB 4
00# 00310 DYN_DESC_2:
      .BYTE 0[2]
02 0E 00312      .BYTE 14, 2
00314      .BLKB 4
00# 00318 INPUT_DESC:
      .BYTE 0[2]
02 0E 0031A      .BYTE 14, 2
0031C      .BLKB 4

      .PSECT $GLOBAL$,NOEXE,2

00000 MAIL$GL_FLAGS::
      .BLKB 4
00004 MAIL$SL_SMG_DISPLAY::
      .BLKB 4
```

```
00008 MAIL$ _SMG_KEYBOARD::
      .BLKB 4
0000C MAIL$ _SMG_KEYTABLE::
      .BLKB 4
0000007F 00010 MAIL$GL_BIGBUFSIZ::
      .LONG 127
00014 MAIL$GL_BIGBUFFER::
      .BLKB 4
00018 MAIL$GL_SYSFLAGS::
      .BLKB 4
00000003 0001C MAIL$ _SIZLIM::
      .LONG 3
00020 MAIL$ _MYUIC::
      .BLKB 4
00024 MAIL$ _MYPID::
      .BLKB 4
00# 00028 MAIL$Q_ATTDESC::
      .BYTE 0[2]
02 0E 0002A .BYTE 14, 2
      0002C .BLKB 4
00# 00030 MAIL$Q_CLUNODES::
      .BYTE 0[2]
02 0E 00032 .BYTE 14, 2
      00034 .BLKB 4
00# 00038 MAIL$Q_SYSPRVMSK::
      .BYTE 0[3]
10 0003B .BYTE 16
      0003C .BLKB 4
00040 MAIL$Q_PROCPRIV::
      .BLKB 8
00048 MAIL$Q_INPTRAN::
      .BLKB 8
00050 MAIL$ _INPCCLASS::
      .BLKB 4
00054 MAIL$ _INPDEVDP::
      .BLKB 4
00058 MAIL$ _OUTCLASS::
      .BLKB 4
0005C MAIL$ _OUTDEVDP::
      .BLKB 4
00060 MAIL$ _OUTLINES::
      .BLKB 4
00000040 00064 MAIL$Q_EDITNAM_DESC::
      .LONG 64
00000000' 00068 .ADDRESS EDITNAM_BUF
      0006C MAIL$B_DFW::
      .BLKB 1
      0006D .BLKB 1
      0006E MAIL$W_TTCHAN::
      .BLKB 2
00070 MAIL$Q_USERNAME::
      .BLKB 8
00# 00078 MAIL$G_CNCT::
      .BYTE 0[128]
02 02 000F8 .BYTE 2, 2
      000FA .BLKB 584
      00342 .BLKB 2
```

```
00344 MAIL$Q_SYSNODE::
      .BLKB 8
00# 0034C MAIL$Q_SYSNODE2::
      .BYTE 0[2]
02 0E 0034E      .BYTE 14, 2
      00350      .BLKB 4
00# 00354 MAIL$Q_CALEDTNAM::
      .BYTE 0[2]
02 0E 00356      .BYTE 14, 2
      00358      .BLKB 4
00# 0035C MAIL$Q_PROTOCOL::
      .BYTE 0[2]
02 0E 0035E      .BYTE 14, 2
      00360      .BLKB 4
00# 00364 MAIL$Q_MAILDEF::
      .BYTE 0[2]
02 0E 00366      .BYTE 14, 2
      00368      .BLKB 4
00# 0036C MAIL$Q_USERDEF::
      .BYTE 0[2]
02 0E 0036E      .BYTE 14, 2
      00370      .BLKB 4
```

```
MAIL$Q_TO==      MAIL$G_CNCT+16
MAIL$Q_SUBJ==     MAIL$G_CNCT+24
MAIL$Q_FULLUSER== MAIL$G_CNCT+8
MAIL$Q_TMP_RAB==  MAIL$G_CNCT+646
TMP_BUFF=        MAIL$G_CNCT+134
TMP_RAB=         MAIL$G_CNCT+646
MAIL$Q_ENHANCED_PRIVS==
                  ENHANCED_PRIVS
NULL_CHAR=       P.AAA
COMMA_DESC=      P.AAB
SLASH_DESC=      P.AAD
SYSIN_DESC=      P.AAF
GETPROCSTS=      P.AAH
TMPFILE_FAO_CDESC= P.AAI
TMPFILE_FAO_CDESC= P.AAK
TMPFILE_FAO_DESC= P.AAM
TMPFILE_FAO_DESC= P.AAO
CALLABLE_EDT=    P.AAQ
TOPIC_DESC=      FILE_DESC
MAIL$Q_TMPFILE_DESC==
                  TMPFILE_DESC
MAIL$Q_TMP_PRO==  65348
SD_ATTACHMENT=   P.AAR
SD_HEADER=       P.AAT
SD_EXTRACT=      P.AAV
SD_FILE=         P.AAX
SD_FILEMF=       P.AAZ
SD_EDT=          P.ABB
SD_EDIT=         P.ABD
SD_LNM$SYSTEM=   P.ABF
SD_LNM$FILE_DEV= P.ABH
SD_MAIL$INIT=    P.ABJ
SD_MAIL$SYSTEM_FLAGS=
                  P.ABL
```

```

SD_PROTOCOL=          P.ABN
SD_REPLF=             P.ABP
SD_SELF=              P.ABR
SD_SEND1=             P.ABT
SD_SUBJECT=           P.ABV
SD_LAST=              P.ABX
SD_NEXT=              P.ABZ
SD_TOLIST=            P.ACB
SD_USER=              P.ACD
SD_TPU=               P.ACF
MAIL$SD_LNM_FILE_DEV==
                        P.ABH
MAIL$SD_LNM_SYSTEM==P.ABF
MAIL$SD_USER==        P.ACD
.EXTRN SMGS_EOF, CLIS_NEGATED
.EXTRN CLIS_LOCNEG, CLISGET_VALUE
.EXTRN CLISDCL_PARSE, CLISDISPATCH
.EXTRN CLISPRESENT, LIB$CVT_DTB
.EXTRN LIB$DISABLE_CTRL
.EXTRN LIB$ENABLE_CTRL
.EXTRN LIB$GET_VM, LIB$SET_FILE_PROT
.EXTRN LIB$TPARSE, MAIL$CMD_READ
.EXTRN MAIL$CMD_CONTINUE
.EXTRN MAIL$DEL_TMPF, MAIL$DELEMP_INK
.EXTRN MAIL$EXTRACT_MESSAGE
.EXTRN MAIL$FILTER_OUTPUT
.EXTRN MAIL$FAO_GETMSG
.EXTRN MAIL$GETMSG, MAIL$GET_DEVDIR
.EXTRN MAIL$HANDLE, MAIL$ADD_ADDR
.EXTRN MAIL$CLOSE_FILE
.EXTRN MAIL$CONFIRM, MAIL$DELETELINK
.EXTRN MAIL$EDIT, MAIL$ENTER
.EXTRN MAIL$GET_INPUT, MAIL$GET_I_MSG
.EXTRN MAIL$GET_MSG, MAIL$GET_NEWMAIL
.EXTRN MAIL$GET_USER, MAIL$INT_ADDR
.EXTRN MAIL$INFO_I_MSG
.EXTRN MAIL$INFO_MSG, MAIL$IOSUB_INIT
.EXTRN MAIL$NET_END_USERS
.EXTRN MAIL$OPEN_IF, MAIL$OPEN_UAF
.EXTRN MAIL$OPEN_SYSMAL
.EXTRN MAIL$PUT_OUTPUT
.EXTRN MAIL$READ_FOREIGN_FILE
.EXTRN MAIL$REWIND_I_MSG
.EXTRN MAIL$REWIND_MSG
.EXTRN MAIL$SEND_ADDR, MAIL$SET_NAME
.EXTRN MAIL$TEST_SUBJ, MAIL$UPDATE_I_MSG
.EXTRN MAIL$UPCASE, MAIL$UPCASE_Q
.EXTRN SMG$CREATE_PASTEBOARD
.EXTRN SMG$ERASE_PASTEBOARD
.EXTRN SMG$CREATE_VIRTUAL_KEYBOARD
.EXTRN SMG$CREATE_KEY_TABLE
.EXTRN SMG$DELETE_VIRTUAL_KEYBOARD
.EXTRN SMG$DEFINE_KEY, STR$APPEND
.EXTRN STR$CONCAT, STR$COPY_DX
.EXTRN STR$FREE1_DX, STR$TRIM
.EXTRN SYSS$FAO, UTIL$GETFILENAME
.EXTRN UTIL$REPORT_IO_ERROR

```


.EXTRN MAILST_RD_RAB, MAIL\$Q_MAIL_MAI
.EXTRN MAIL\$B_MSGTRAN, MAIL\$COMMAND_TABLE
.EXTRN MAIL\$C_CURMSG, MAIL\$C_USR\$SYS\$JF
.EXTRN MAIL\$C_MSG_NUM, MAIL\$C_MSG_RADR
.EXTRN MAIL\$W_MSG_RSIZ
.EXTRN MAIL\$C_MSG_STS, MAIL\$Q_MSGNAME
.EXTRN MAIL\$Q_MSGTYPE, MAIL\$T_UAF_BUFF
.EXTRN SYSS\$CANCEL

4B

54

.PSECT \$CODE\$,NOWRT,2

00000000' 00 0000 00000
50 00000000' 01 8A 00002
00 3C 00009
09 13 00010
50 DD 00012
00000000G 00 01 FB 00014
04 0001B 1\$:

.ENTRY MAIL\$DISABLE_CTRLC, Save nothing : 0654
BICB2 #1, MAIL\$GL_FLAGS+1 : 0662
MOVZWL MAIL\$W_TCHAN, R0 : 0663
BEQL 1\$:
PUSHL R0 : 0664
CALLS #1, SYSS\$CANCEL :
RET : 0666

45

49

; Routine Size: 28 bytes, Routine Base: \$CODE\$ + 0244

46

2F
46

52
4F

45
49

44
3D

4C
22

57
2F

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

D 15
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2 Page 22
(5)

```

: 526      0667 1 ROUTINE CTRLCAST : NOVALUE =
: 527      0668 2 BEGIN
: 528      0669 2 :++
: 529      0670 2 : FUNCTIONAL DESCRIPTION:
: 530      0671 2 :
: 531      0672 2 :         Handles the CTRL/C ast, re-enables for next one
: 532      0673 2 :
: 533      0674 2 MAIL$ENABLE_CTRLC();
: 534      0675 2 MAIL$GL_FLAGS[MAIF_V_CTRLCFL] = 1;
: 535      0676 2 RETURN;
: 536      0677 1 END;
```

0000 00000 CTRLCAST:

00000000V 00
00000000' 00

00 FB 00002
01 88 00009
04 00010

.WORD
CALLS
BISB2
RET

Save nothing
#0, MAIL\$ENABLE_CTRLC
#1, MAIL\$GL_FLAGS+1

: 0667
: 0674
: 0675
: 0677

; Routine Size: 17 bytes, Routine Base: \$CODE\$ + 0260


```

: 538      0678 1 GLOBAL ROUTINE MAIL$ENABLE_CTRLC =
: 539      0679 1 ++
: 540      0680 1 FUNCTIONAL DESCRIPTION:
: 541      0681 1
: 542      0682 1      Enables control-c's on MAIL$W_TTCHAN
: 543      0683 1
: 544      0684 1 --
: 545      0685 2 BEGIN
: 546      0686 2 LOCAL
: 547      0687 2     IOSB : VECTOR[4,WORD];
: 548      0688 2
: 549      0689 2 MAIL$GL_FLAGS[MAIF_V_CTRLCL] = 0;
: 550      0690 2 IF .MAIL$GL_FLAGS[MAIF_V_TERM]
: 551      0691 3 THEN BEGIN
: 552      P 0692 3     IF_ERR($QIOW(CHAN=.MAIL$W_TTCHAN,
: 553      P 0693 3         FUNC=IOS$ SETMODE OR IOSM_CTRLCAST,
: 554      P 0694 3         IOSB=IOSB,
: 555      P 0695 3         P1=CTRLCAST,
: 556      P 0696 3         P2=0);
: 557      0697 3     SIGNAL(.STATUS));
: 558      0698 3     IF NOT .IOSB[0]
: 559      0699 3     THEN SIGNAL(.IOSB[0]);
: 560      0700 2     END;
: 561      0701 2
: 562      0702 2 RETURN TRUE
: 563      0703 1 END;
```

```

                                000C 00000
                                00 9E 00002
                                00 9E 00009
                                08 C2 00010
                                01 8A 00013
31      01      62      02 E1 00017
                                7E 7C 0001B
                                7E 7C 0001D
                                7E D4 0001F
                                CB AF 9F 00021
                                7E 7C 00024
                                20 AE 9F 00026
                                7E 0123 8F 3C 00029
                                7E 6E A2 3C 0002E
                                00000000G 00 0C FB 00034
                                05 50 E8 0003B
                                50 DD 0003E
                                63 01 FB 00040
                                06 6E E8 00043 1$:
                                7E 6E 3C 00046
                                63 01 FB 00049
                                50 01 D0 0004C 2$:
                                04 0004F
```

.EXTRN SYSSQIOW

```

.ENTRY MAIL$ENABLE_CTRLC, Save R2,R3      : 0678
MOVAB LIB$SIGNAL, R3
MOVAB MAIL$GL_FLAGS, R2
SUBL2 #8, SP
BICB2 #1, MAIL$GL_FLAGS+1      : 0689
BBC #2, MAIL$GL_FLAGS, 2$      : 0690
CLRQ -(SP)      : 0697
CLRQ -(SP)
CLRL -(SP)
PUSHAB CTRLCAST
CLRQ -(SP)
PUSHAB IOSB
MOVZWL #291, -(SP)
MOVZWL MAIL$W_TTCHAN, -(SP)
CLRL -(SP)
CALLS #12, SYSSQIOW
BLBS STATUS, 1$
PUSHL STATUS
CALLS #1, LIB$SIGNAL
BLBS IOSB, 2$      : 0698
MOVZWL IOSB, -(SP)      : 0699
CALLS #1, LIB$SIGNAL
MOVL #1, R0      : 0702
RET      : 0703
```

; Routine Size: 80 bytes, Routine Base: \$CODE\$ + 0271

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

~~14-SEP-1984~~ 01:01:06
~~14-SEP-1984~~ 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2 Page 24
(6)

MA
VO

```

: 565      0704 1 GLOBAL ROUTINE MAIL$DISABLE_CTRLY : NOVALUE =
: 566      0705 1 ++
: 567      0706 1 FUNCTIONAL DESCRIPTION:
: 568      0707 1
: 569      0708 1 Disable CLI control/y handling
: 570      0709 1
: 571      0710 1 --
: 572      0711 2 BEGIN
: 573      0712 2
: 574      P 0713 2 IF_ERR(LIB$DISABLE_CTRL(%REF(%X'02000000'),CTRL_MASK);,
: 575      0714 2 SIGNAL(.STATUS));
: 576      0715 2 RETURN;
: 577      0716 1 END;

```

			0000 00000	.ENTRY	MAIL\$DISABLE_CTRLY, Save nothing	: 0704
	5E		04 C2 00002	SUBL2	#4, SP	
		00000000'	00 9F 00005	PUSHAB	CTRL_MASK	: 0714
04	AE	02000000	8F D0 00008	MOVL	#33554432, 4(SP)	
		04	AE 9F 00013	PUSHAB	4(SP)	
00000000G	00		02 FB 00016	CALLS	#2, LIB\$DISABLE_CTRL	
	09		50 E8 0001D	BLBS	STATUS, 1\$	
			50 DD 00020	PUSHL	STATUS	
00000000G	00		01 FB 00022	CALLS	#1, LIB\$SIGNAL	
			04 00029 1\$:	RET		: 0716

: Routine Size: 42 bytes. Routine Base: \$CODE\$ + 02C1

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

H 15
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2 Page 26
(8)

MA
V0

```
: 579      0717 1 GLOBAL ROUTINE MAIL$ENABLE_CTRLY : NOVALUE =
: 580      0718 1 :++
: 581      0719 1 : FUNCTIONAL DESCRIPTION:
: 582      0720 1 :
: 583      0721 1 :         Enable CLI control/y handling
: 584      0722 1 :
: 585      0723 1 :--
: 586      0724 2 BEGIN
: 587      0725 2
: 588      P 0726 2 IF_ERR(LIB$ENABLE_CTRL(CTRL_MASK);,
: 589      0727 2     SIGNAL (.STATUS));
: 590      0728 2 RETURN;
: 591      0729 1 END;
```

			0000 00000	.ENTRY	MAIL\$ENABLE_CTRLY, Save nothing	: 0717
			00 9F 00002	PUSHAB	CTRL_MASK	: 0727
00000000G	00	00000000'	01 FB 00008	CALLS	#1, LIB\$ENABLE_CTRL	:
	09		50 E8 0000F	BLBS	STATUS, 1\$	:
			50 DD 00012	PUSHL	STATUS	:
00000000G	00		01 FB 00014	CALLS	#1, LIB\$SIGNAL	:
			04 0001B 1\$:	RET		: 0729

; Routine Size: 28 bytes, Routine Base: \$CODE\$ + 02EB

```
: 593 0730 1 ROUTINE call_tpase(string,states,keys) =  
: 594 0731 2 BEGIN  
: 595 0732 2  
: 596 0733 2  
: 597 0734 2 This routine calls tpase given the string, states and keys  
: 598 0735 2  
: 599 0736 2 Inputs:  
: 600 0737 2  
: 601 0738 2 string address of string descriptor for string to parse  
: 602 0739 2 states address of tpase states table  
: 603 0740 2 keys address of tpase keys table  
: 604 0741 2  
: 605 0742 2  
: 606 0743 2 MAP  
: 607 0744 2 string : REF $BBLOCK;  
: 608 0745 2  
: 609 0746 2 LOCAL  
: 610 0747 2 tpablock : $bblock[tpa$length0],  
: 611 0748 2 status;  
: 612 0749 2  
: 613 0750 2  
: 614 0751 2 Initialize the tpase block  
: 615 0752 2  
: 616 0753 2 CH$FILL(0,tpa$length0,tpablock);  
: 617 0754 2 tpablock[tpa$l_count] = tpa$length0;  
: 618 0755 2 tpablock[tpa$l_options] = tpa$m_abbrev;  
: 619 0756 2 tpablock[tpa$l_stringcnt] = .string[dsc$w_length];  
: 620 0757 2 tpablock[tpa$l_stringptr] = .string[dsc$a_pointer];  
: 621 0758 2  
: 622 0759 2 RETURN lib$tpase(tpablock,.states,.keys)  
: 623 0760 2  
: 624 0761 1 END;
```

				003C 00000 CALL_TPASE:			
		5E	24 C2 00002	.WORD	Save R2,R3,R4,R5	:	0730
24	00	6E	00 2C 00005	SUBL2	#36, SP	:	
			6E 0000A	MOVC5	#0, (SP), #0, #36, TPABLOCK	:	0753
		6E	08 D0 0000B	MOVL	#8, TPABLOCK	:	0754
	04	AE	02 D0 0000E	MOVL	#2, TPABLOCK+4	:	0755
		50	04 AC D0 00012	MOVL	STRING, R0	:	0756
	08	AE	60 3C 00016	MOVZWL	(R0), TPABLOCK+8	:	
	0C	AE	04 A0 D0 0001A	MOVL	4(R0), TPABLOCK+12	:	0757
		7E	08 AC 7D 0001F	MOVQ	STATES, -(SP)	:	0759
			08 AE 9F 00023	PUSHAB	TPABLOCK	:	
	00000000G	00	03 FB 00026	CALLS	#3, LIB\$TPASE	:	
			04 0002D	RET		:	0761

; Routine Size: 46 bytes, Routine Base: \$CODE\$ + 0307

```

: 626 0762 1 ROUTINE PARSE_EDIT_QUAL (CLIDEFAULT) =
: 627 0763 2 BEGIN
: 628 0764 2 ++
: 629 0765 2 FUNCTIONAL DESCRIPTION:
: 630 0766 2
: 631 0767 2 This routine parses the /EDIT qualifier on SEND and REPLY
: 632 0768 2 commands, taking into account the DCL MAIL/EDIT qualifier.
: 633 0769 2
: 634 0770 2 INPUTS:
: 635 0771 2
: 636 0772 2 CLIDEFAULT is the state of /EDIT from the DCL MAIL/EDIT command
: 637 0773 2 --
: 638 0774 2 LOCAL
: 639 0775 2 EDIT_STATUS;
: 640 0776 2
: 641 0777 2 Set the edit flag if either MAIL/EDIT, or /EDIT on this command. Then,
: 642 0778 2 if user explicitly said /NOEDIT, clear the flag
: 643 0779 2
: 644 0780 2 EDIT_STATUS = CLISPRESENT(SD_EDIT);
: 645 0781 2 CLIFLAGS[EDIT_FLAG] = .CLIDEFAULT OR .EDIT_STATUS;
: 646 0782 2 IF .EDIT_STATUS EQL CLIS_NEGATED
: 647 0783 2 OR .EDIT_STATUS EQL CLIS_LOCNEG
: 648 0784 2 THEN CLIFLAGS[EDIT_FLAG] = 0;
: 649 0785 2
: 650 0786 2 IF NOT .MAIL$GL_FLAGS[MAIF_V_TERM]
: 651 0787 2 THEN (EDIT_STATUS = 0; CLIFLAGS[EDIT_FLAG] = 0);
: 652 0788 2
: 653 0789 2 RETURN .EDIT_STATUS
: 654 0790 1 END;
```

```

                                0004 00000 PARSE_EDIT_QUAL:
                                .WORD Save R2
                                MOVAB CLIFLAGS, R2
                                PUSHAB SD_EDIT
                                CALLS #1, CLISPRESENT
                                BISB3 EDIT_STATUS, CLIDEFAULT, R1
                                INSV R1, #0, #1, CLIFLAGS
                                CMPL EDIT_STATUS, #CLIS_NEGATED
                                BEQL 1$
                                CMPL EDIT_STATUS, #CLIS_LOCNEG
                                BNEQ 2$
                                BICB2 #1, CLIFLAGS
                                BBS #2, MAIL$GL_FLAGS, 3$
                                CLRL EDIT_STATUS
                                BICB2 #1, CLIFLAGS
                                RET

62 51 00000000G 00 00000000' 00 9E 00002 .WORD Save R2
    01 04 AC 50 89 00014 MOVAB CLIFLAGS, R2
    00 00 51 F0 00019 PUSHAB SD_EDIT
    00000000G 8F 50 D1 0001E CALLS #1, CLISPRESENT
    00000000G 8F 09 13 00025 BISB3 EDIT_STATUS, CLIDEFAULT, R1
    00000000G 8F 50 D1 00027 INSV R1, #0, #1, CLIFLAGS
    00000000G 62 03 12 0002E CMPL EDIT_STATUS, #CLIS_NEGATED
    05 00000000' 00 01 8A 00030 1$: BEQL 1$
    00 02 E0 00033 2$: CMPL EDIT_STATUS, #CLIS_LOCNEG
    62 50 D4 0003B BNEQ 2$
    01 8A 0003D BICB2 #1, CLIFLAGS
    04 00040 3$: BBS #2, MAIL$GL_FLAGS, 3$
                                CLRL EDIT_STATUS
                                BICB2 #1, CLIFLAGS
                                RET
```

; Routine Size: 65 bytes, Routine Base: \$CODE\$ + 0335

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

K 15
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2
Page 29
(11)

MA
V0

; 656 0791 1 GLOBAL ROUTINE MAIL\$CMD_SET_EXIT = (CLIFLAGS[EXIT_FLAG] = 1;RETURN 1);

00000000'	00	0000 0000	.ENTRY MAIL\$CMD_SET_EXIT, Save nothing	: 0791
	50	08 88 00002	BISB2 #8, CLIFLAGS	:
		01 D0 00009	MOVL #1, R0	:
		04 0000C	RET	:

; Routine Size: 13 bytes, Routine Base: \$CODE\$ + 0376

; 657 0792 1 GLOBAL ROUTINE MAIL\$CMD_SET_QUIT = (CLIFLAGS[QUIT_FLAG] = 1;RETURN 1);

00000000'	00	0000 0000	.ENTRY MAIL\$CMD_SET_QUIT, Save nothing	: 0792
	50	10 88 00002	BISB2 #16, CLIFLAGS	:
		01 D0 00009	MOVL #1, R0	:
		04 0000C	RET	:

; Routine Size: 13 bytes, Routine Base: \$CODE\$ + 0383

; 658 0793 1 GLOBAL ROUTINE MAIL\$CMD_NULL = RETURN 1;

	50	0000 0000	.ENTRY MAIL\$CMD_NULL, Save nothing	: 0793
		01 D0 00002	MOVL #1, R0	:
		04 00005	RET	:

; Routine Size: 6 bytes, Routine Base: \$CODE\$ + 0390

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

L 15
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2 Page 30
(12)

MA
VO

```
: 660      0794 1 GLOBAL ROUTINE MAIL$CMD_ERASE =  
: 661      0795 1 |++  
: 662      0796 1 | FUNCTIONAL DESCRIPTION:  
: 663      0797 1 |  
: 664      0798 1 |           Erase the screen.  
: 665      0799 1 |--  
: 666      0800 2 BEGIN  
: 667      0801 2 LOCAL  
: 668      0802 2     DESC : VECTOR[2,LONG];  
: 669      0803 2  
: 670      0804 2 !DESC[0] = 0;  
: 671      0805 2 !DESC[1] = DESC[0];  
: 672      0806 2 !MAIL$PUT_OUTPUT(DESC);  
: 673      0807 2 RETURN SMG$ERASE_PASTEBOARD(MAIL$L_SMG_DISPLAY);  
: 674      0808 1 END;
```

```
                                0000 00000  
                                08 C2 00002  
SE 000000000' 00 9F 00005  
00000000G 00 01 FB 0000B  
                                04 00012
```

```
.ENTRY MAIL$CMD_ERASE, Save nothing  
SUBL2 #8, SP  
PUSHAB MAIL$L_SMG_DISPLAY  
CALLS #1, SMG$ERASE_PASTEBOARD  
RET
```

```
: 0794  
: 0807  
: 0808
```

; Routine Size: 19 bytes, Routine Base: \$CODE\$ + 0396

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

M 15
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2 Page 31
(13)

MA
V0

```
: 676      0809 1 ROUTINE IGNORE_SIGNALS =  
: 677      0810 2 BEGIN  
: 678      0811 2 |++  
: 679      0812 2 | FUNCTIONAL DESCRIPTION:  
: 680      0813 2 |  
: 681      0814 2 |         Ignore any signals by setting SS$_CONTINUE  
: 682      0815 2 |--  
: 683      0816 2 |  
: 684      0817 2 CLI_SIGNALLED = 1;  
: 685      0818 2 RETURN SS$_CONTINUE  
: 686      0819 1 END;
```

0000 00000 IGNORE_SIGNALS:

00000000' 00
50

01 D0 00002
01 D0 00009
04 0000C

.WORD Save nothing
MOVL #1, CLI_SIGNALLED
MOVL #1, R0
RET

: 0809
: 0817
: 0818
: 0819

; Routine Size: 13 bytes, Routine Base: \$CODE\$ + 03A9

```

: 688      0820 1 ROUTINE DEFINE_DEFAULT_KEYPAD : NOVALUE =
: 689      0821 1 ++
: 690      0822 1 FUNCTIONAL DESCRIPTION:
: 691      0823 1
: 692      0824 1 Define the default keypad.
: 693      0825 1
: 694      0826 1 --
: 695      0827 2 BEGIN
: 696      0828 2
: 697      0829 2 BIND
: 698      0830 2 KEYDEFTAB = PLIT(
: 699      0831 2 $DESCRIPTOR('DEF/K PF1 '' /NOTERM/SET=GOLD/NOLOCK'),
: 700      0832 2 $DESCRIPTOR('DEF/K PF2 'HELP'/TERM'),
: 701      0833 2 $DESCRIPTOR('DEF/K PF3 'EXTRACT/MAIL'/TERM'),
: 702      0834 2 $DESCRIPTOR('DEF/K PF4 'ERASE' /TERM /NOECHO'),
: 703      0835 2 $DESCRIPTOR('DEF/K KP7 'SEND'/TERM'),
: 704      0836 2 $DESCRIPTOR('DEF/K KP8 'REPLY'/TERM'),
: 705      0837 2 $DESCRIPTOR('DEF/K KP9 'FORWARD'/TERM'),
: 706      0838 2 $DESCRIPTOR('DEF/K KP4 'CURRENT'/TERM'),
: 707      0839 2 $DESCRIPTOR('DEF/K KP5 'FIRST'/TERM'),
: 708      0840 2 $DESCRIPTOR('DEF/K KP6 'LAST'/TERM'),
: 709      0841 2 $DESCRIPTOR('DEF/K KP3 'DIR'/TERM'),
: 710      0842 2 $DESCRIPTOR('DEF/K KP2 'PRINT'/TERM'),
: 711      0843 2 $DESCRIPTOR('DEF/K KP1 'BACK'/TERM'),
: 712      0844 2 $DESCRIPTOR('DEF/K KP0 'NEXT'/TERM'),
: 713      0845 2 $DESCRIPTOR('DEF/K MINUS 'READ/NEW'/TERM'),
: 714      0846 2 $DESCRIPTOR('DEF/K COMMA 'DIR/NEW'/TERM'),
: 715      0847 2 $DESCRIPTOR('DEF/K PERIOD 'FILE ''/NOTERM'),
: 716      0848 2 $DESCRIPTOR('DEF/K PF2 'DIR/FOLDER'/TERM/IF=GOLD'),
: 717      0849 2 $DESCRIPTOR('DEF/K PF3 'EXTRACT'/TERM/IF=GOLD'),
: 718      0850 2 $DESCRIPTOR('DEF/K PF4 'SELECT MAIL'/TERM/IF=GOLD'),
: 719      0851 2 $DESCRIPTOR('DEF/K KP7 'SEND/EDIT'/TERM/IF=GOLD'),
: 720      0852 2 $DESCRIPTOR('DEF/K KP8 'REPLY/EDIT/EXTRACT'/TERM/IF=GOLD'),
: 721      0853 2 $DESCRIPTOR('DEF/K KP9 'FORWARD/EDIT'/TERM/IF=GOLD'),
: 722      0854 2 $DESCRIPTOR('DEF/K KP4 'CURRENT/EDIT'/TERM/IF=GOLD'),
: 723      0855 2 $DESCRIPTOR('DEF/K KP5 'FIRST/EDIT'/TERM/IF=GOLD'),
: 724      0856 2 $DESCRIPTOR('DEF/K KP6 'LAST/EDIT'/TERM/IF=GOLD'),
: 725      0857 2 $DESCRIPTOR('DEF/K KP3 'DIR/START=999999'/TERM/IF=GOLD'),
: 726      0858 2 $DESCRIPTOR('DEF/K KP2 'PRINT/PRINT/NOTIFY'/TERM/IF=GOLD'),
: 727      0859 2 $DESCRIPTOR('DEF/K KP1 'BACK/EDIT'/TERM/IF=GOLD'),
: 728      0860 2 $DESCRIPTOR('DEF/K KP0 'NEXT/EDIT'/TERM/IF=GOLD'),
: 729      0861 2 $DESCRIPTOR('DEF/K MINUS 'SHOW NEW'/TERM/IF=GOLD'),
: 730      0862 2 $DESCRIPTOR('DEF/K COMMA 'DIR MAIL'/TERM/IF=GOLD'),
: 731      0863 2 $DESCRIPTOR('DEF/K PERIOD 'DELETE ''/NOTERM/IF=GOLD'),
: 732      0864 2 $DESCRIPTOR('DEF/K ENTER 'SELECT ''/NOTERM/IF=GOLD'),
: 733      0865 2 ) : VECTOR[.LONG],
: 734      0866 2 NUMBER_KEYDEFINES = KEYDEFTAB[-1] : LONG;
: 735      0867 2
: 736      0868 2 INCRU I FROM 0 TO .NUMBER_KEYDEFINES-1
: 737      0869 2 DO SMG$DEFINE_KEY(MAIL$$_SMG_KEYTABLE,.KEYDEFTAB[I]);
: 738      0870 2
: 739      0871 2 RETURN;
: 740      0872 1 END;
```

4E 2F 20 22 22 20 31 46 50 20 4B 2F 46 45 44 003B6 P.ACJ: .ASCII \DEF/K PF1 '' /NOTERM/SET=GOLD/NOLOCK\ ;

B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

[illegible]

MAILSMAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

C 16
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2

Page 34
(14)

4B	43	41	42	22	20	31	50	4B	20	4B	2F	46	45	44	00000000'	0054C		.ADDRESS P.ADF		
									4D	52	45	54	2F	22	00550	P.ADH:	.ASCII \DEF/K KP1 'BACK'/TERM\			
															0055F					
															00565		.BLKB 3			
															00568	P.ADG:	.LONG 21			
54	58	45	4E	22	20	30	50	4B	20	4B	2F	46	45	44	00000000'	0056C		.ADDRESS P.ADH		
									4D	52	45	54	2F	22	00570	P.ADJ:	.ASCII \DEF/K KP0 'NEXT'/TERM\			
															0057F					
															00585		.BLKB 3			
															00588	P.ADI:	.LONG 21			
45	52	22	20	53	55	4E	49	4D	20	4B	2F	46	45	44	00000000'	0058C		.ADDRESS P.ADJ		
			4D	52	45	54	2F	22	57	45	4E	2F	44	41	00590	P.ADL:	.ASCII \DEF/K MINUS 'READ/NEW'/TERM\			
															0059F					
															005AB		.BLKB 1			
															005AC	P.ADK:	.LONG 27			
49	44	22	20	41	4D	4D	4F	43	20	4B	2F	46	45	44	00000000'	005B0		.ADDRESS P.ADL		
				4D	52	45	54	2F	22	57	45	4E	2F	52	005B4	P.ADN:	.ASCII \DEF/K COMMA 'DIR/NEW'/TERM\			
															005C3					
															005CE		.BLKB 2			
															005D0	P.ADM:	.LONG 26			
46	22	20	44	4F	49	52	45	50	20	4B	2F	46	45	44	00000000'	005D4		.ADDRESS P.ADN		
			4D	52	45	54	4F	4E	2F	22	20	45	4C	49	005D8	P.ADP:	.ASCII \DEF/K PERIOD 'FILE '/NOTERM\			
															005E7					
															005F3		.BLKB 1			
2F	52	49	44	22	20	32	46	50	20	4B	2F	46	45	44	00000000'	005F4	P.ADO:	.LONG 27		
46	49	2F	4D	52	45	54	2F	22	52	45	44	4C	4F	46	005F8		.ADDRESS P.ADP			
										44	4C	4F	47	3D	005FC	P.ADR:	.ASCII \DEF/K PF2 'DIR/FOLDER'/TERM/IF=GOLD\			
															0060B					
															0061A					
															0061F		.BLKB 1			
															00620	P.ADQ:	.LONG 35			
															00624		.ADDRESS P.ADR			
52	54	58	45	22	20	33	46	50	20	4B	2F	46	45	44	00000000'	00628	P.ADT:	.ASCII \DEF/K PF3 'EXTRACT'/TERM/IF=GOLD\		
4F	47	3D	46	49	2F	4D	52	45	54	2F	22	54	43	41	00637					
															00646					
															00648	P.ADS:	.LONG 32			
															0064C		.ADDRESS P.ADT			
45	4C	45	53	22	20	34	46	50	20	4B	2F	46	45	44	00000000'	00650	P.ADV:	.ASCII \DEF/K PF4 'SELECT MAIL'/TERM/IF=GOLD\		
49	2F	4D	52	45	54	2F	22	4C	49	41	4D	20	54	43	0065F					
									44	4C	4F	47	3D	46	0066E					
															00674	P.ADU:	.LONG 36			
															00678		.ADDRESS P.ADV			
44	4E	45	53	22	20	37	50	4B	20	4B	2F	46	45	44	0067C	P.ADX:	.ASCII \DEF/K KP7 'SEND/EDIT'/TERM/IF=GOLD\			
3D	46	49	2F	4D	52	45	54	2F	22	54	49	44	45	2F	0068B					
											44	4C	4F	47	0069A					
															0069E		.BLKB 2			
															006A0	P.ADW:	.LONG 34			
															006A4		.ADDRESS P.ADX			
4C	50	45	52	22	20	38	50	4B	20	4B	2F	46	45	44	006A8	P.ADZ:	.ASCII \DEF/K KP8 'REPLY/EDIT/EXTRACT'/TERM/IF=G\			
22	54	43	41	52	54	58	45	2F	54	49	44	45	2F	59	006B7					
					47	3D	46	49	2F	4D	52	45	54	2F	006C6					
												44	4C	4F	006D0		.ASCII \OLD\			
															006D3		.BLKB 1			
															006D4	P.ADY:	.LONG 43			
															006D8		.ADDRESS P.ADZ			
57	52	4F	46	22	20	39	50	4B	20	4B	2F	46	45	44	006DC	P.AEB:	.ASCII \DEF/K KP9 'FORWARD/EDIT'/TERM/IF=GOLD\			
2F	4D	52	45	54	2F	22	54	49	44	45	2F	44	52	41	006EB					

[illegible]

														00000023	008A3	P.AES:	.BLKB 1
														00000000	008A4		.LONG 35
44	22	20	44	4F	49	52	45	50	20	4B	2F	46	45	44	008A8	P.AEV:	.ADDRESS P.AET
2F	4D	52	45	54	4F	4E	2F	22	20	45	54	45	4C	45	008AC		.ASCII \DEF/K PERIOD 'DELETE '/NOTERM/IF=GOLD\
								44	4C	4F	47	3D	46	49	008BB		
															008CA		
															008D1		.BLKB 3
															008D4	P.AEU:	.LONG 37
															008D8		.ADDRESS P.AEV
45	53	22	20	52	45	54	4E	45	20	4B	2F	46	45	44	008DC	P.AEX:	.ASCII \DEF/K ENTER 'SELECT '/NOTERM/IF=GOLD\
49	2F	4D	52	45	54	4F	4E	2F	22	20	54	43	45	4C	008EB		
								44	4C	4F	47	3D	46		008FA		
															00900	P.AEW:	.LONG 36
															00904		.ADDRESS P.AEX
															00908		.LONG 34
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	0090C	P.ACH:	.ADDRESS P.ACI, P.ACK, P.ACM, P.ACO, P.ACQ, -
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00924		P.ACS, P.ACU, P.ACW, P.ACY, P.ADA, P.ADC, -
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	0093C		P.ADE, P.ADQ, P.ADI, P.ADK, P.ADM, P.ADO, -
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00954		P.AEQ, P.AES, P.AEU, P.AEW,
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	0096C		P.AEC, P.AEE, P.AEG, P.AEI, P.AEK, P.AEM, -
															00984		P.AEO, P.AEQ, P.AES, P.AEU, P.AEW

```
KEYDEFTAB=      P.ACH
NUMBER_KEYDEFINES= P.ACH-4
```

				000C 00000		DEFINE_DEFAULT_KEYPAD:			
53	FF6D	CF	01	C3	00002		.WORD	Save R2,R3	: 0820
			52	D4	00008		SUBL3	#1, NUMBER_KEYDEFINES, R3	: 0868
			14	11	0000A		CLRL	1	:
			42	DD	0000C	1\$:	BRB	2\$	:
		FF67, CF	00	9F	00011		PUSHL	KEYDEFTAB[I]	: 0869
	00000000		02	FB	00017		PUSHAB	MAIL\$1 SMG KEYTABLE	:
00000000G	00		52	D6	0001E		CALLS	#2, SMG\$DEFINE_KEY	:
			52	D1	00020	2\$:	INCL	1	:
	53		E7	1B	00023		CMPL	1, R3	:
				04	00025		BLEQU	1\$	:
							RET		: 0872

; Routine Size: 38 bytes, Routine Base: \$CODE\$ + 0994


```

742 0873 1 ROUTINE LOAD_KEY_DEFS =
743 0874 1 ++
744 0875 1 FUNCTIONAL DESCRIPTION:
745 0876 1
746 0877 1 If MAIL$INIT is defined, load the specified file
747 0878 1
748 0879 1 --
749 0880 2 BEGIN
750 0881 2 LOCAL
751 0882 2 DESC : VECTOR[2, LONG] VOLATILE,
752 0883 2 STATUS,
753 0884 2 TMPFAB : $FAB_DECL,
754 0885 2 TMPRAB : $RAB_DECL,
755 0886 2 TMPNAM : $NAM_DECL,
756 0887 2 NAMBUF : $BLOCK[NAM$C_MAXRSS],
757 0888 2 TRNLNMLST : $ITMLST_DECL(ITEMS=1),
758 0889 2 IBUF : $BLOCK[255];
759 0890 2
760 0891 2 BIND
761 0892 2 DPTR = DESC[1] : REF VECTOR[BYTE];
762 0893 2
763 0894 2 DESC[0] = NAM$C_MAXRSS;
764 0895 2 DESC[1] = IBUF;
765 P 0896 2 $ITMLST_INIT(ITMLST=TRNLNMLST,
766 0897 2 (ITMCOD=LNMS_STRING, BUFADR=.DESC[1], BUFSIZ=.DESC[0], RETLEN=DESC[0]));
767 0898 2
768 P 0899 2 IF NOT $TRNLNM(ATTR=XREF(LNMSM_CASE_BLIND),
769 P 0900 2 TABNAM=MAIL$SD_LNM_FILE_DEV,
770 P 0901 2 LOGNAM=SD_MAIL$INIT,
771 0902 2 ITMLST=TRNLNMLST)
772 0903 2 THEN RETURN TRUE;
773 0904 2
774 P 0905 2 $FAB_INIT(FAB=TMPFAB,
775 P 0906 2 FNS=.DESC[0],
776 P 0907 2 FNA=.DESC[1],
777 P 0908 2 DNM='MAIL$INIT.INI',
778 P 0909 2 CTX=MAIL$_OPENIN,
779 P 0910 2 FAC=GET,
780 0911 2 FOP=SQO);
781 0912 2
782 P 0913 2 $NAM_INIT(NAM=TMPNAM,
783 P 0914 2 ESS=NAM$C_MAXRSS,
784 P 0915 2 ESA=NAMBUF,
785 P 0916 2 RSS=NAM$C_MAXRSS,
786 0917 2 RSA=NAMBUF);
787 0918 2
788 F 0919 2 IF_ERR($OPEN(FAB=TMPFAB, ERR=UTIL$REPORT_IO_ERROR);,
789 0920 2 RETURN .STATUS);
790 0921 2
791 P 0922 2 $RAB_INIT(RAB=TMPRAB,
792 P 0923 2 FAB=TMPFAB,
793 P 0924 2 CTX=MAIL$_OPENIN,
794 P 0925 2 UBF=IBUF,
795 0926 2 USZ=512);
796 0927 2
797 P 0928 2 IF_ERR($CONNECT(RAB=TMPRAB, ERR=UTIL$REPORT_IO_ERROR);,
798 P 0929 2 $CLOSE(FAB=TMPFAB);
```

```
799      0930      2      RETURN .STATUS);
800      0931      2
801      0932      2      TMPRAB[RAB$L CTX] = MAIL$ READERR;
802      0933      2      WHILE $GET(RAB=TMPRAB,ERR=UTIL$REPORT_IO_ERROR)
803      0934      2      DO BEGIN
804      0935      2          DESC[0] = .TMPRAB[RAB$W_RSZ];
805      0936      2          DESC[1] = .TMPRAB[RAB$L_RBF];
806      0937      2          IF .DESC[0] NEQ 0
807      0938      2              AND .DPTR[0] NEQ XC!!
808      P 0939      2              THEN IF_ERR(SMG$DEFINE_KEY(MAIL$L_SMG_KEYTABLE,DESC);,
809      0940      2                  SIGNAL(MAIL$_PARSEFAIL,1,DESC,.STATUS));
810      0941      2      END;
811      0942      2
812      0943      2      $CLOSE(FAB=TMPFAB);
813      0944      2      RETURN TRUE
814      0945      1      END;
```

49 4E 49 2E 54 49 4E 49 24 4C 49 41 4D 009BA P.AEY: .ASCII \MAIL\$INIT.INI\

```
.EXTRN SYS$TRNLNM, SYS$OPEN
.EXTRN SYS$CONNECT, SYS$CLOSE
.EXTRN SYS$GET
```

01FC 00000 LOAD_KEY_DEFS:

WORD Save R2,R3,R4,R5,R6,R7,R8

MOVAB SYS\$CLOSE, R8

MOVAB SD MAIL\$INIT, R7

MOVAB UTIL\$REPORT_IO_ERROR, R6

MOVAB -784(SP), SP

MOVZBL #255, DESC

MOVAB IBUF, DESC+4

MOVAB TRNLNMLST, \$\$ITMBLKPTR

MOVW DESC, (\$\$ITMBLKPTR)+

MOVW #2, (\$\$ITMBLKPTR)+

MOVL DESC+4, (\$\$ITMBLKPTR)+

MOVAB DESC, (\$\$ITMBLKPTR)+

CLRL (\$\$ITMBLKPTR)+

PUSHAB TRNLNMLST

CLRL -(SP)

PUSHL R7

PUSHAB MAIL\$SD_LNM_FILE_DEV

MOVL #33554432, T6(SP)

PUSHAB 16(SP)

CALLS #5, SYS\$TRNLNM

BLBS R0, 1\$

BRW 6\$

MOVCS #0, (SP), #0, #80, \$RMS_PTR

MOVW #20483, \$RMS_PTR

MOVZBL #64, \$RMS_PTR+4

MOVW #2, \$RMS_PTR+22

MOVL #8261786, \$RMS_PTR+24

MOVW #2, \$RMS_PTR+3T

MOVL DESC+4, \$RMS_PTR+44

MOVAB P.AEY, \$RMS_PTR+48

0873

0894

0895

0897

0902

0911

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

H 16
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2

Page 39
(15)

0060	8F	00	DC	AD	F8	AD	90	0008C	MOVB	DESC, \$RMS_PTR+52	:	
			DD	AD		0D	90	00091	MOVB	#13, \$RMS_PTR+53	:	
			6E	6E		00	2C	00095	MOVCS	#0, (SP), #0, #96, \$RMS_PTR	:	0917
					FF04	CD		0009C			:	
			FF04	CD	6002	8F	B0	0009F	MOVW	#24578, \$RMS_PTR	:	
			FF06	CD		01	8E	000A6	MNEGB	#1, \$RMS_PTR+2	:	
			FF08	CD	0114	CE	9E	000AB	MOVAB	NAMBUF, \$RMS_PTR+4	:	
			FF0E	CD		01	8E	000B2	MNEGB	#1, \$RMS_PTR+10	:	
			FF10	CD	0114	CE	9E	000B7	MOVAB	NAMBUF, \$RMS_PTR+12	:	
					A8	56	DD	000BE	PUSHL	R6	:	0920
		00000000G				AD	9F	000C0	PUSHAB	TMFAB	:	
			00			02	FB	000C3	CALLS	#2, SYS\$OPEN	:	
			01			50	E8	000CA	BLBS	STATUS, 2\$	:	
							04	000CD	RET		:	
0044	8F	00		6E		00	2C	000CE	MOVCS	#0, (SP), #0, #68, \$RMS_PTR	:	0926
					FF64	CD		000D5			:	
			FF64	CD	4401	8F	B0	000D8	MOVW	#17409, \$RMS_PTR	:	
			FF7C	CD	007E109A	8F	D0	000DF	MOVL	#8261786, \$RMS_PTR+24	:	
			84	AD	0200	8F	B0	000E8	MOVW	#512, \$RMS_PTR+32	:	
			88	AD	04	AE	9E	000EE	MOVAB	IBUF, \$RMS_PTR+36	:	
			A0	AD	A8	AD	9E	000F3	MOVAB	TMFAB, \$RMS_PTR+60	:	
					FF64	56	DD	000F8	PUSHL	R6	:	0930
		00000000G				CD	9F	000FA	PUSHAB	TMPRAB	:	
			00			02	FB	000FE	CALLS	#2, SYS\$CONNECT	:	
			52			50	D0	00105	MOVL	R0, STATUS	:	
			0A			52	E8	00108	BLBS	STATUS, 3\$	:	
					A8	AD	9F	0010B	PUSHAB	TMFAB	:	
			68			01	FB	0010E	CALLS	#1, SYS\$CLOSE	:	
			50			52	D0	00111	MOVL	STATUS, R0	:	
							04	00114	RET		:	
		FF7C	CD	007E10B0		8F	D0	00115	MOVL	#8261808, TMPRAB+24	:	0932
					FF64	56	DD	0011E	PUSHL	R6	:	0933
						CD	9F	00120	PUSHAB	TMPRAB	:	
		00000000G	00			02	FB	00124	CALLS	#2, SYS\$GET	:	
			3E			50	E9	0012B	BLBS	R0, 5\$	:	
			F8	AD	86	AD	3C	0012E	MOVZWL	TMPRAB+34, DESC	:	0935
			FC	AD	8C	AD	D0	00133	MOVL	TMPRAB+40, DESC+4	:	0936
					F8	AD	D5	00138	TSTL	DESC	:	0937
						E1	13	0013B	BEQL	4\$	:	
			21		FC	BC	91	0013D	CMPB	@DPTR, #33	:	0938
						DB	13	00141	BEQL	4\$	:	
					F8	AD	9F	00143	PUSHAB	DESC	:	0940
					00000000	00	9F	00146	PUSHAB	MAIL\$ SMG KEYTABLE	:	
		00000000G	00			02	FB	0014C	CALLS	#2, SMG\$DEFINE_KEY	:	
			C8			50	E8	00153	BLBS	STATUS, 4\$	:	
						50	DD	00156	PUSHL	STATUS	:	
					F8	AD	9F	00158	PUSHAB	DESC	:	
						01	DD	0015B	PUSHL	#1	:	
					007E1248	8F	DD	0015D	PUSHL	#8262216	:	
		00000000G	00			04	FB	00163	CALLS	#4, LIB\$SIGNAL	:	
						B2	11	0016A	BRB	4\$	:	0933
					A8	AD	9F	0016C	PUSHAB	TMFAB	:	0943
			68			01	FB	0016F	CALLS	#1, SYS\$CLOSE	:	
			50			01	D0	00172	MOVL	#1, RU	:	0944
							04	00175	RET		:	0945

; Routine Size: 374 bytes, Routine Base: \$CODE\$ + 09C7

MAILMAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

1 16
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2 Page 40
(15)

```
0946 1 ROUTINE MAIL_DISPATCH =
0947 1 ++
0948 1 FUNCTIONAL DESCRIPTION:
0949 1
0950 1 This routine attempts to parse the input file spec from DCL
0951 1 and then dispatches to process either a CLI, or interactive command
0952 1
0953 2 BEGIN
0954 2 BUILTIN
0955 2 FP;
0956 2
0957 2 LOCAL
0958 2 DESC : $BBLOCK[DSC$C_S_BLN],
0959 2 STATUS;
0960 2
0961 2 .FP = IGNORE_SIGNALS;
0962 2
0963 2 STATUS = 0;
0964 2 STATUS = CLISGET_VALUE(SD_FILE, INPUT_DESC);
0965 2
0966 2 .FP = 0;
0967 2
0968 2 INIT_SDESC(DESC);
0969 2 DESC[DSC$W_LENGTH] = .MAIL$L_USRSYSBUF[VMD_B_FNMLNG];
0970 2 DESC[DSC$A_POINTER] = MAIL$L_USRSYSBUF[VMD_T_FWDNAM] +
0971 2 .MAIL$L_USRSYSBUF[VMD_B_FWDLNG];
0972 2
0973 2 MAIL$SET_NAME(DESC);
0974 2
0975 3 RETURN (IF .STATUS
0976 3 AND NOT .CLI_SIGNALLED
0977 3 THEN CLI_COMMAND()
0978 2 ELSE INTJOB()) OR STS$M_INHIB_MSG
0979 1 END;
```

```
000C 00000 MAIL_DISPATCH:
53 F866 CF 9E 00002 .WORD Save R2,R3
5E 08 C2 00007 MOVAB IGNORE_SIGNALS, R3
6D 63 9E 0000A SUBL2 #8, SP
00000000' 52 D4 0000D MOVAB IGNORE_SIGNALS, (FP)
F5DA CF 9F 0000F CLRL STATUS
00000000G 00 9F 00015 PUSHAB INPUT_DESC
52 02 FB 00019 PUSHAB SD_FILE
00000000G 50 D0 00020 CALLS #2, CLISGET_VALUE
6E 50 D0 00023 MOVL R0, STATUS
51 42 A0 9B 0002E CLRL (FP)
04 AE 44 A140 9E 00036 CLRQ DESC
00000000G 00 D0 00027 MOVL MAIL$L_USRSYSBUF, R0
6E 42 A0 9B 0002E MOVZBW 66(R0), DESC
51 43 A0 9A 00032 MOVZBL 67(R0), R1
04 AE 44 A140 9E 00036 MOVAB 68(R1)(R0), DESC+4
00000000G 00 5E DD 0003C PUSHL SP
01 FB 0003E CALLS #1, MAIL$SET_NAME
```

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

K 16

16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2

Page 42
(16)

		10		52	E9	00045	BLBC	STATUS, 1\$	:	0975
		09	00000000'	00	E8	00048	BLBS	CLI_SIGNALLED, 1\$	:	0976
		00		00	FB	0004F	CALLS	#0, -CLI_COMMAND	:	0977
		07		07	11	00056	BRB	2\$	:	
50		00		00	FB	00058	CALLS	#0, INTJOB	:	0978
	01	1C		01	FO	0005F	INSV	#1, #28, #1, R0	:	
				04	00064		RET		:	0979

; Routine Size: 101 bytes, Routine Base: \$CODE\$ + 0B3D

```

: 851 0980 1 ROUTINE CHECK_SERVER : NOVALUE =
: 852 0981 1 |++
: 853 0982 1 | FUNCTIONAL DESCRIPTION:
: 854 0983 1 |
: 855 0984 1 |         If /PROTOCOL specified on the command line, and the process
: 856 0985 1 |         has SYSPRV and DETACH, then pretend it's a network server.
: 857 0986 1 |         This allows foreign protocol input without network connections.
: 858 0987 1 | --
: 859 0988 2 BEGIN
: 860 0989 2 BUILTIN FP;
: 861 0990 2
: 862 0991 2 .FP = IGNORE SIGNALS;
: 863 0992 2 IF CL$PRESENT(SD_PROTOCOL)
: 864 0993 2     AND .MAIL$Q-PROCPRIV[PRV$V-SYSPRV]
: 865 0994 2     AND .MAIL$Q-PROCPRIV[PRV$V-DETACH]
: 866 0995 2 THEN MAIL$GL_FLAGS[MAIF_V-NETJOB] = 1;
: 867 0996 2 RETURN;
: 868 0997 1 END;
```

```

                                000C 00000 CHECK_SERVER:
                                .WORD      Save R2,R3
                                MOVAB      IGNORE_SIGNALS, R3
                                MOVAB      MAIL$Q-PROCPRIV, R2
                                MOVAB      IGNORE_SIGNALS, (FP)
                                PUSHAB     SD_PROTOCOL
                                CALLS      #1, CL$PRESENT
                                BLBC       R0, 1$
                                BBC        #4, MAIL$Q-PROCPRIV+3, 1$
                                BBC        #5, MAIL$Q-PROCPRIV, 1$
                                BISB2     #2, MAIL$GL_FLAGS
                                RET
                                04 0002C 1$:

      53      F801      CF  9E 00002      .WORD      Save R2,R3
      52 00000000'  00  9E 00007      MOVAB      IGNORE_SIGNALS, R3
      6D      F609      63  9E 0000E      MOVAB      MAIL$Q-PROCPRIV, R2
                                01  FB 00015      MOVAB      IGNORE_SIGNALS, (FP)
                                50  E9 0001C      PUSHAB     SD_PROTOCOL
                                04  E1 0001F      CALLS      #1, CL$PRESENT
      08      03  A2      04  E1 00024      BLBC       R0, 1$
      04      C0  A2      05  E1 00028      BBC        #4, MAIL$Q-PROCPRIV+3, 1$
                                02  88 00028      BBC        #5, MAIL$Q-PROCPRIV, 1$
                                04 0002C 1$:      BISB2     #2, MAIL$GL_FLAGS
                                RET
```

; Routine Size: 45 bytes, Routine Base: \$CODE\$ + 0BA2

```

: 870      0998 1 GLOBAL ROUTINE MAIL =
: 871      0999 1
: 872      1000 1 ++
: 873      1001 1 FUNCTIONAL DESCRIPTION:
: 874      1002 1
: 875      1003 1 Main entry point to MAIL utility. Determines type of use: CLI command
: 876      1004 1 processor, interactive mail command processor, or slave process for
: 877      1005 1 network mail processing.
: 878      1006 1
: 879      1007 1 INPUTS:
: 880      1008 1
: 881      1009 1 Command language interpreter parameters.
: 882      1010 1
: 883      1011 1 OUTPUTS:
: 884      1012 1
: 885      1013 1 Mail utility executed.
: 886      1014 1 --
: 887      1015 2 BEGIN
: 888      1016 2 LOCAL
: 889      1017 2 DESC : VECTOR[2, LONG] VOLATILE,
: 890      1018 2 PTR : REF $BBLOCK,
: 891      1019 2 PRIVMASK : $BBLOCK[8],
: 892      1020 2 TRNLNMLST : $ITMLST_DECL(ITEMS=1),
: 893      1021 2 STATUS;
: 894      1022 2
: 895      1023 2 ENABLE MAIL$HANDLE; ! Enable condition handler
: 896      1024 2
: 897      1025 2 Determine if process is an interactive or network job
: 898      1026 2
: 899      P 1027 2 IF_ERR($GETJPIW(ITMLST=GETPROCSTS);,
: 900      1028 2 RETURN STATUS;);
: 901      1029 2 MAIL$GL_FLAGS[MAIF_V_BATCH] = .PROCSTS<$BITPOSITION(PCB$V_BATCH), 1>;
: 902      1030 2 MAIL$GL_FLAGS[MAIF_V_NETJOB] = .PROCSTS<$BITPOSITION(PCB$V_NETWRK), 1>;
: 903      1031 2 CHECK_SERVER();
: 904      1032 2 IF NOT .MAIL$GL_FLAGS[MAIF_V_NETJOB]
: 905      1033 3 THEN BEGIN
: 906      1034 3 !
: 907      1035 3 ! Disable privs that user doesn't have
: 908      1036 3 !
: 909      1037 3 PRIVMASK<0, 32, 0> = 0;
: 910      1038 3 PRIVMASK[PRV$V_WORLD] = NOT .MAIL$Q_PROCPRIV[PRV$V_WORLD];
: 911      1039 3 PRIVMASK[PRV$V_OPER] = NOT .MAIL$Q_PROCPRIV[PRV$V_OPER];
: 912      1040 3 PRIVMASK[PRV$V_SYSNAM] = NOT .MAIL$Q_PROCPRIV[PRV$V_SYSNAM];
: 913      1041 3 PRIVMASK[PRV$V_SYSPRV] = NOT .MAIL$Q_PROCPRIV[PRV$V_SYSPRV];
: 914      1042 3 PRIVMASK<32, 32, 0> = 0;
: 915      1043 3 $SETPRV(ENBFLG=0, PRVADR=PRIVMASK);
: 916      1044 2 END;
: 917      1045 2 !
: 918      1046 2 ! Allocate some buffers
: 919      1047 2 !
: 920      P 1048 2 IF_ERR(LIB$GET_VM(%REF(UAF$C_LENGTH), MAIL$T_UAF_BUFF);,
: 921      1049 2 RETURN SIGNAL(.STATUS));
: 922      P 1050 2 IF_ERR(LIB$GET_VM(%REF(.MAIL$GL_BIGBUFSIZ*512), MAIL$GL_BIGBUFFER);,
: 923      1051 2 RETURN SIGNAL(.STATUS));
: 924      P 1052 2 IF_ERR(MAIL$IOSUB_INIT();,
: 925      1053 2 RETURN STATUS);
: 926      1054 3 MAIL$Q_INPTRAN[DSC$W_LENGTH] = (IF .MAIL$GL_FLAGS[MAIF_V_NETJOB]
```



```

927      1055 3      THEN 2*NAM$C_MAXRSS
928      1056 2      ELSE NAM$C_MAXRSS);
929      1057 2
930      P 1058 2 IF_ERR(LIB$GET_VM(MAIL$Q_INPTRAN[DSC$W_LENGTH],
931      P 1059 2 MAIL$Q_INPTRAN[DSC$A_POINTER]));,
932      1060 2      RETURN SIGNAL(.STATUS));
933      1061 2
934      1062 2 : Initialize global CNCT block
935      1063 2
936      1064 2 MAIL$Q_USERNAME[DSC$A_POINTER] = USERNAME;
937      1065 2 MAIL$G_CNCT[CNCT_L_FLINK] = MAIL$G_CNCT[CNCT_L_FLINK];
938      1066 2 MAIL$G_CNCT[CNCT_L_BLINK] = MAIL$G_CNCT[CNCT_L_FLINK];
939      1067 2 MAIL$G_CNCT[CNCT_Q_ADRLIST]+4 =
940      1068 2 (MAIL$G_CNCT[CNCT_Q_ADRLIST])<0,32,0> = MAIL$G_CNCT[CNCT_Q_ADRLIST];
941      1069 2 MAIL$G_CNCT[CNCT_Q_LNKLIST]+4 =
942      1070 2 (MAIL$G_CNCT[CNCT_Q_LNKLIST])<0,32,0> = MAIL$G_CNCT[CNCT_Q_LNKLIST];
943      1071 2 PTR = MAIL$G_CNCT[CNCT_Q_FROMDESC];
944      1072 2 INIT_DDESC(PTR);
945      1073 2 PTR = MAIL$G_CNCT[CNCT_Q_TODESC];
946      1074 2 INIT_DDESC(PTR);
947      1075 2 PTR = MAIL$G_CNCT[CNCT_Q_SUBJDESC];
948      1076 2 INIT_DDESC(PTR);
949      1077 2
950      1078 2 : Translate MAIL$SYSTEM_FLAGS
951      1079 2
952      1080 2 DESC[0] = 0;
953      1081 2 DESC[1] = .MAIL$GL_BIGBUFFER;
954      P 1082 2 $ITMLST_INIT(ITMLST=TRNLNMLST,
955      P 1083 2 (ITMCD=LNMS_STRING,BUFADR=.MAIL$GL_BIGBUFFER,
956      1084 2 BUFSIZ=64,RETLEN=DESC));
957      1085 2
958      P 1086 2 IF $TRNLNM(ATTR=$REF(LNMSM_CASE_BLIND),
959      P 1087 2 TABNAM=SD_LNMS$SYSTEM,
960      P 1088 2 LOGNAM=SD_MAIL$SYSTEM_FLAGS,
961      P 1089 2 ACMODE=$REF(PSL$C_EXEC),
962      1090 3 ITMLST=TRNLNMLST)
963      1091 2 THEN IF CALL TPARSE(DESC,GETNUM_STATE,GETNUM_KEY)
964      1092 2 THEN MAIL$GL_SYSFLAGS = .TPA_NUMBER;
965      1093 2
966      1094 2
967      1095 2 : Translate SYSS$INPUT or SYSS$NET logical name and get device's characteristics
968      1096 2
969      P 1097 2 $ITMLST_INIT(ITMLST=TRNLNMLST,
970      P 1098 2 (ITMCD=LNMS_STRING,BUFADR=.MAIL$Q_INPTRAN[DSC$A_POINTER],
971      1099 2 BUFSIZ=.MAIL$Q_INPTRAN[DSC$W_LENGTH],RETLEN=MAIL$Q_INPTRAN));
972      1100 2
973      P 1101 2 STATUS = $TRNLNM(ATTR=$REF(LNMSM_CASE_BLIND),
974      P 1102 2 TABNAM=MAIL$SD_LRM_FICE_DEV,
975      P 1103 2 LOGNAM=(IF .MAIL$G_FLAGS[MAIF_V_NETJOB]
976      P 1104 2 THEN $DESCRIPTOR('SYSS$NET')
977      P 1105 2 ELSE SYSSIN_DESC),
978      1106 2 ITMLST=TRNLNMLST);
979      1107 2 IF CH$RCHAR(.MAIL$Q_INPTRAN[DSC$A_POINTER]) EQL ESC ! Check for RMS escape
980      1108 3 THEN BEGIN ! If so, strip it off name
981      1109 3 MAIL$Q_INPTRAN[DSC$W_LENGTH] = .MAIL$Q_INPTRAN[DSC$W_LENGTH] - 4;
982      1110 3 MAIL$Q_INPTRAN[DSC$A_POINTER] = .MAIL$Q_INPTRAN[DSC$A_POINTER] + 4;
983      1111 2 END;
```

```

984      1112 2  !
985      1113 2  ! Translate SYSSNODE. Strip off the leading '-' and trailing colons
986      1114 2  ! if they are present.
987      1115 2  !
988      1116 2  MAIL$Q_SYSSNODE[DSC$W_LENGTH] = 64;
989      1117 2  MAIL$Q_SYSSNODE[DSC$A_POINTER] = SYSSNODEBUF;
990      1118 2  $ITMLST_INIT(ITMLST=TRNLNMLST,
991      1119 2  (ITMCOD=LNMS_STRING,BUFADR=SYSSNODEBUF,BUFSIZ=64,RETLEN=MAIL$Q_SYSSNODE));
992      1120 2  !
993      1121 2  IF NOT (STATUS = $STRNLNM(ATTR=%REF(LNMSM_CASE_BLIND),
994      1122 2  TABNAM=SD_LNM$SYSTEM,
995      1123 2  LOGNAM=$DESCRIPTOR('SYSSNODE'),
996      1124 2  ACMODE=%REF(PSL$C_EXEC),
997      1125 2  ITMLST=TRNLNMLST))
998      1126 2  THEN MAIL$Q_SYSSNODE[DSC$W_LENGTH] = 0
999      1127 2  ELSE BEGIN
1000      1128 2  IF .SYSSNODEBUF[0] EQL '-'
1001      1129 2  THEN BEGIN
1002      1130 2  MAIL$Q_SYSSNODE[DSC$W_LENGTH] = .MAIL$Q_SYSSNODE[DSC$W_LENGTH] - 1;
1003      1131 2  MAIL$Q_SYSSNODE[DSC$A_POINTER] = .MAIL$Q_SYSSNODE[DSC$A_POINTER] + 1;
1004      1132 2  END;
1005      1133 2  INCRU I FROM 0 TO 1
1006      1134 2  DO IF NOT CH$FAIL(CH$FIND CH(.MAIL$Q_SYSSNODE[DSC$W_LENGTH],
1007      1135 2  .MAIL$Q_SYSSNODE[DSC$A_POINTER],':'))
1008      1136 2  THEN MAIL$Q_SYSSNODE[DSC$W_LENGTH] = .MAIL$Q_SYSSNODE[DSC$W_LENGTH] - 1;
1009      1137 2  !
1010      1138 2  ! Make a copy with colons on the end for forwarding loop detection
1011      1139 2  !
1012      1140 2  STR$CONCAT(MAIL$Q_SYSSNODE2,MAIL$Q_SYSSNODE,$DESCRIPTOR('::'));
1013      1141 2  END;
1014      1142 2  !
1015      1143 2  ! Get SYSSINPUT and SYSSOUTPUT device characteristics
1016      1144 2  !
1017      1145 2  BEGIN
1018      1146 2  LOCAL
1019      1147 2  GETDVILST : $ITMLST_DECL(ITEMS=2);
1020      1148 2  !
1021      1149 2  $ITMLST_INIT(ITMLST=GETDVILST,
1022      1150 2  (ITMCOD=DVIS_DEVCLASS,BUFADR=MAIL$L_INPCLASS),
1023      1151 2  (ITMCOD=DVIS_DEVDEPEND,BUFADR=MAIL$L_INPDEVDP));
1024      1152 2  !
1025      1153 2  $GETDVIW(DEVNAM=MAIL$Q_INPTRAN,
1026      1154 2  ITMLST=GETDVILST);
1027      1155 2  !
1028      1156 2  $ITMLST_INIT(ITMLST=GETDVILST,
1029      1157 2  (ITMCOD=DVIS_DEVCLASS,BUFADR=MAIL$L_OUTCLASS),
1030      1158 2  (ITMCOD=DVIS_DEVDEPEND,BUFADR=MAIL$L_OUTDEVDP));
1031      1159 2  !
1032      1160 2  $GETDVIW(DEVNAM=$DESCRIPTOR('SYSSOUTPUT'),
1033      1161 2  ITMLST=GETDVILST);
1034      1162 2  END;
1035      1163 2  !
1036      1164 2  IF .MAIL$L_INPCLASS EQL DCS_TERM
1037      1165 2  THEN MAIL$GL_FLAGS[MAIF_V_ITEM] = 1;
1038      1166 2  !
1039      1167 2  IF .MAIL$L_OUTCLASS EQL DCS_TERM
1040      1168 2  THEN BEGIN
```



```
1041 1169 3 MAIL$GL_FLAGS[MAIF_V_OTERM] = 1;
1042 1170 3 MAIL$GL_FLAGS[MAIF_V_SCOPE] = .MAIL$L_OUTDEVDP<$BITPOSITION(TT$V_SCOPE),1>;
1043 1171 2 END;
1044 1172 2
1045 1173 2 MAIL$L_OUTLINES =
1046 1174 2 .MAIL$L_OUTDEVDP<$BITPOSITION(TT$V_PAGE),$FIELDWIDTH(TT$V_PAGE)>;
1047 1175 2 IF .MAIL$GL_FLAGS[MAIF_V_BATCH] !No need for pages in batch
1048 1176 2 OR .MAIL$GL_OUTLINES EQL 0
1049 1177 2 THEN MAIL$C_OUTLINES = %X'7FFFFFFF';
1050 1178 2
1051 1179 2 Open the system user authorization file and the mail system data file
1052 1180 2
1053 1181 2 IF_ERR(MAIL$OPEN_UAF(),RETURN .STATUS);
1054 1182 2 IF_ERR(MAIL$OPEN_SYSMail(),RETURN .STATUS);
1055 1183 2
1056 1184 2 Create the string descriptors for temporary files
1057 1185 2
1058 1186 2 DESC[0] = NAM$C_MAXRSS;
1059 1187 2 DESC[1] = .MAIL$GL_BIGBUFFER;
1060 1188 3 SYS$FAO((IF .MAIL$C_MYPID<16,16,0> NEQ 0
1061 1189 3 THEN TMPFILE_FAO_CDESC
1062 1190 2 ELSE TMPFILE_FAO_DESC),DESC,DESC,.MAIL$C_MYPID);
1063 1191 2 STR$COPY_DX(TMPFILE_DESC,DESC);
1064 1192 2 DESC[0] = NAM$C_MAXRSS;
1065 1193 2 DESC[1] = .MAIL$C_L_BIGBUFFER;
1066 1194 3 SYS$FAO((IF .MAIL$C_MYPID<16,16,0> NEQ 0
1067 1195 3 THEN TMPFILE1_FAO_CDESC
1068 1196 2 ELSE TMPFILE1_FAO_DESC),DESC,DESC,.MAIL$C_MYPID);
1069 1197 2 STR$COPY_DX(TMPFILE1_DESC,DESC);
1070 1198 2
1071 1199 2 If this is a network process, dispatch to the network server code
1072 1200 2
1073 1201 2 IF .MAIL$GL_FLAGS[MAIF_V_NETJOB]
1074 1202 2 THEN RETURN NETJOB();
1075 1203 2
1076 1204 2 Declare an exit handler to clean up any logical links on exit
1077 1205 2
1078 1206 2 $DCLEXH(DESBLK=EXIT_BLOCK);
1079 1207 2
1080 1208 2 Dispatch according to whether this is a CLI command request, an interactive
1081 1209 2 mail utility session.
1082 1210 2
1083 1211 2
1084 1212 2 RETURN MAIL_DISPATCH()
1085 1213 2
1086 1214 1 END;
```

```
54 45 4E 24 53 59 53 00BCF P.AFA: .ASCII \SYS$NET\
00BD6 .BLKB 2
00000007 00BD8 P.AEZ: .LONG 7
00000000 00BDC .ADDRESS P.AFA
45 44 4F 4E 24 53 59 53 00BE0 P.AFC: .ASCII \SYS$NODE\
00000008 00BE8 P.AFB: .LONG 8
00000000 00BEC .ADDRESS P.AFC
3A 3A 00BF0 P.AFE: .ASCII \::\
```

[illegible]

08	AE	10	A3	14	08	02	FB	000D7	CALLS	#2, LIB\$GET_VM	
			3C			50	E9	000DA	BLBC	STATUS, 6\$	1051
						A3	9F	000DD	PUSHAB	MAIL\$GL_BIGBUFFER	
						09	78	000E0	ASHL	#9, MAIL\$GL_BIGBUFSIZ, 8(SP)	
						AE	9F	000E6	PUSHAB	8(SP)	
			66			02	FB	000E9	CALLS	#2, LIB\$GET_VM	
			2A			50	E9	000EC	BLBC	STATUS, 6\$	1053
			00			00	FB	000EF	CALLS	#0, MAIL\$IOSUB_INIT	
			01			50	E8	000F6	BLBS	STATUS, 3\$	
							04	000F9	RET		
07			63			01	E1	000FA	BBC	#1, MAIL\$GL_FLAGS, 4\$	1054
			50	01FE		8F	3C	000FE	MOVZWL	#510, R0	1055
						04	11	00103	BRB	5\$	
			50	FF		8F	9A	00105	MOVZBL	#255, R0	1054
		48	A3			50	B0	00109	MOVW	R0, MAIL\$Q_INPTRAN	
				4C		A3	9F	0010D	PUSHAB	MAIL\$Q_INPTRAN+4	1060
				48		A3	9F	00110	PUSHAB	MAIL\$Q_INPTRAN	
			66			02	FB	00113	CALLS	#2, LIB\$GET_VM	
			0A			50	E8	00116	BLBS	STATUS, 7\$	
						50	DD	00119	PUSHL	STATUS	
			00			01	FB	0011B	CALLS	#1, LIB\$SIGNAL	
							04	00122	RET		
		74	A3	E0		A5	9E	00123	MOVAB	USERNAME, MAIL\$Q_USERNAME+4	1064
		78	A3	78		A3	9E	00128	MOVAB	MAIL\$G_CNCT, MAIL\$G_CNCT	1065
		7C	A3	78		A3	9E	0012D	MOVAB	MAIL\$G_CNCT, MAIL\$G_CNCT+4	1066
			50	00A0		C3	9E	00132	MOVAB	MAIL\$G_CNCT+40, R0	1068
		00A0	C3			50	D0	00137	MOVL	R0, MAIL\$G_CNCT+40	
		00A4	C3			50	D0	0013C	MOVL	R0, MAIL\$G_CNCT+44	
			50	00A8		C3	9E	00141	MOVAB	MAIL\$G_CNCT+48, R0	1070
		00A8	C3			50	D0	00146	MOVL	R0, MAIL\$G_CNCT+48	
		00AC	C3			50	D0	0014B	MOVL	R0, MAIL\$G_CNCT+52	
			50	0080		C3	9E	00150	MOVAB	MAIL\$G_CNCT+8, PTR	1071
			60	020E0000		8F	D0	00155	MOVL	#34471936, (PTR)	1072
				04		A0	D4	0015C	CLRL	4(PTR)	
			50	0088		C3	9E	0015F	MOVAB	MAIL\$G_CNCT+16, PTR	1073
			60	020E0000		8F	D0	00164	MOVL	#34471936, (PTR)	1074
				04		A0	D4	0016B	CLRL	4(PTR)	
			50	0090		C3	9E	0016E	MOVAB	MAIL\$G_CNCT+24, PTR	1075
			60	020E0000		8F	D0	00173	MOVL	#34471936, (PTR)	1076
				04		A0	D4	0017A	CLRL	4(PTR)	
				3C		AE	D4	0017D	CLRL	DESC	1080
			51	14		A3	D0	00180	MOVL	MAIL\$GL_BIGBUFFER, R1	1081
			AE			51	D0	00184	MOVL	R1, DESC+4	
			50	24		AE	9E	00188	MOVAB	TRNLNMLST, \$\$ITMBLKPTR	1084
			80	00020040		8F	D0	0018C	MOVL	#131136, (\$\$ITMBLKPTR)+	
			80			51	D0	00193	MOVL	R1, (\$\$ITMBLKPTR)+	
			80	3C		AE	9E	00196	MOVAB	DESC, (\$\$ITMBLKPTR)+	
						80	D4	0019A	CLRL	(\$\$ITMBLKPTR)+	
				24		AE	9F	0019C	PUSHAB	TRNLNMLST	1090
			AE			01	D0	0019F	MOVL	#1, 8(SP)	
				08		AE	9F	001A3	PUSHAB	8(SP)	
				44		A4	9F	001A6	PUSHAB	SD_MAIL\$SYSTEM_FLAGS	
						54	DD	001A9	PUSHL	R4	
			10	AE	02000000	8F	D0	001AB	MOVL	#33554432, 16(SP)	
				10		AE	9F	001B3	PUSHAB	16(SP)	
			67			05	FB	001B6	CALLS	#5, SYS\$TRNLNM	
			1D			50	E9	001B9	BLBC	R0, 8\$	

		00000000'	00	9F	001BC	PUSHAB	GETNUM_KEY	1091
		00000000'	00	9F	001C2	PUSHAB	GETNUM_STATE	
		44	AE	9F	001C8	PUSHAB	DESC	
F527	CF		03	FB	001CB	CALLS	#3, CALL_TPARSE	
	06		50	E9	001D0	BLBC	R0, 8\$	
18	A3	FECO	C5	D0	001D3	MOVL	TPA_NUMBER, MAIL\$GL_SYSFLAGS	1092
	50	24	AE	9E	001D9	MOVAB	TRNLNMLST, \$\$ITMBLKPTR	1099
	80	48	A3	B0	001DD	MOVW	MAIL\$Q_INPTRAN, (\$\$ITMBLKPTR)+	
	80		02	B0	001E1	MOVW	#2, (\$\$ITMBLKPTR)+	
	80	4C	A3	D0	001E4	MOVL	MAIL\$Q_INPTRAN+4, (\$\$ITMBLKPTR)+	
	80	48	A3	9E	001E8	MOVAB	MAIL\$Q_INPTRAN, (\$\$ITMBLKPTR)+	
			80	D4	001EC	CLRL	(\$\$ITMBLKPTR)+	
		24	AE	9F	001EE	PUSHAB	TRNLNMLST	1106
			7E	D4	001F1	CLRL	-(SP)	
C7	63		01	E1	001F3	BBC	#1, MAIL\$GL_FLAGS, 9\$	
	50	0A6C	C4	9E	0C1F7	MOVAB	P.AEZ, R0	
			05	11	001FC	BRB	10\$	
	50	FEBC	C4	9E	001FE	MOVAB	SYSIN_DESC, R0	
			50	DD	00203	PUSHL	R0	
		14	A4	9F	00205	PUSHAB	MAIL\$SD_LNM_FILE_DEV	
14	AE	02000000	8F	D0	00208	MOVL	#33554432, 20(SP)	
		14	AE	9F	00210	PUSHAB	20(SP)	
	67		05	FB	00213	CALLS	#5, SYS\$TRNLNM	
	51	4C	A3	D0	00216	MOVL	MAIL\$Q_INPTRAN+4, R1	1107
	1B		61	91	0021A	CMPB	(R1), #27	
			08	12	0021D	BNEQ	11\$	
48	A3		04	A2	0021F	SUBW2	#4, MAIL\$Q_INPTRAN	1109
4C	A3		04	C0	00223	ADDL2	#4, MAIL\$Q_INPTRAN+4	1110
0344	C3	40	8F	9B	00227	MOVZBL	#64, MAIL\$Q_SYSNODE	1116
0348	C3		65	9E	0022D	MOVAB	SYSNODEBUF, MAIL\$Q_SYSNODE+4	1117
	51	24	AE	9E	00232	MOVAB	TRNLNMLST, \$\$ITMBLKPTR	1119
	81	00020040	8F	D0	00236	MOVL	#131136, (\$\$ITMBLKPTR)+	
	81		65	9E	0023D	MOVAB	SYSNODEBUF, (\$\$ITMBLKPTR)+	
	81	0344	C3	9E	00240	MOVAB	MAIL\$Q_SYSNODE, (\$\$ITMBLKPTR)+	
			81	D4	0C245	CLRL	(\$\$ITMBLKPTR)+	
		24	AE	9F	00247	PUSHAB	TRNLNMLST	1125
08	AE		01	D0	0024A	MOVL	#1, 8(SP)	
		08	AE	9F	0024E	PUSHAB	8(SP)	
		0A7C	C4	9F	00251	PUSHAB	P.AFB	
			54	DD	00255	PUSHL	R4	
10	AE	02000000	8F	D0	00257	MOVL	#33554432, 16(SP)	
		10	AE	9F	0025F	PUSHAB	16(SP)	
	67		05	FB	00262	CALLS	#5, SYS\$TRNLNM	
	06		50	E8	00265	BLBS	STATUS, 12\$	
		0344	C3	B4	00268	CLRW	MAIL\$Q_SYSNODE	1126
			41	11	0026C	BRB	17\$	
5F	8F		65	91	0026E	CMPB	SYSNODEBUF, #95	1128
			08	12	00272	BNEQ	13\$	
		0344	C3	B7	00274	DECW	MAIL\$Q_SYSNODE	1130
		0348	C3	D6	00278	INCL	MAIL\$Q_SYSNODE+4	1131
			52	D4	0027C	CLRL	1	1133
	50	0348	C3	D0	0027E	MOVL	MAIL\$Q_SYSNODE+4, R0	1135
60	0344	C3	3A	3A	00283	LOCC	#58, MAIL\$Q_SYSNODE, (R0)	1134
			02	12	00289	BNEQ	15\$	
			51	D4	0028B	CLRL	R1	
			51	D5	0028D	TSTL	R1	1135
			04	13	0028F	BEQL	16\$	

			0344	C3	B7	00291	DECW	MAIL\$Q_SYSNODE	1136
				52	D6	00295	INCL	I	1134
		01		52	D1	00297	CMPL	I, #1	
				E2	1B	0029A	BLEQU	14\$	
			0A88	C4	9F	0029C	PUSHAB	P.AFD	1140
			0344	C3	9F	002A0	PUSHAB	MAIL\$Q_SYSNODE	
			034C	C3	9F	002A4	PUSHAB	MAIL\$Q_SYSNODE2	
	00000000G	00		03	FB	002A8	CALLS	#3, STR\$CONCAT	
		50	08	AE	9E	002AF	MOVAB	GETDVILST, \$\$ITMBLKPTR	1151
		80	00040004	8F	D0	002B3	MOVL	#262148, (\$\$ITMBLKPTR)+	
		80	50	A3	9E	002BA	MOVAB	MAIL\$L_INPCCLASS, (\$\$ITMBLKPTR)+	
				80	D4	002BE	CLRL	(\$\$ITMBLKPTR)+	
		80	000A0004	8F	D0	002C0	MOVL	#655364, (\$\$ITMBLKPTR)+	
		80	54	A3	9E	002C7	MOVAB	MAIL\$L_INPDEVDP, (\$\$ITMBLKPTR)+	
				80	7C	002CB	CLRL	(\$\$ITMBLKPTR)+	
				7E	7C	002CD	CLRL	-(SP)	1154
				7E	7C	002CF	CLRL	-(SP)	
		18		AE	9F	002D1	PUSHAB	GETDVILST	
		48		A3	9F	002D4	PUSHAB	MAIL\$Q_INPTRAN	
				7E	7C	002D7	CLRL	-(SP)	
		68		08	FB	002D9	CALLS	#8, SYSS\$GETDVIW	
		50	08	AE	9E	002DC	MOVAB	GETDVILST, \$\$ITMBLKPTR	1158
		80	00040004	8F	D0	002E0	MOVL	#262148, (\$\$ITMBLKPTR)+	
		80	58	A3	9E	002E7	MOVAB	MAIL\$L_OUTCLASS, (\$\$ITMBLKPTR)+	
				80	D4	002EB	CLRL	(\$\$ITMBLKPTR)+	
		80	000A0004	8F	D0	002ED	MOVL	#655364, (\$\$ITMBLKPTR)+	
		80	5C	A3	9E	002F4	MOVAB	MAIL\$L_OUTDEVDP, (\$\$ITMBLKPTR)+	
				80	7C	002F8	CLRL	(\$\$ITMBLKPTR)+	
				7E	7C	002FA	CLRL	-(SP)	1161
				7E	7C	002FC	CLRL	-(SP)	
		18		AE	9F	002FE	PUSHAB	GETDVILST	
			0A9C	C4	9F	00301	PUSHAB	P.AFF	
				7E	7C	00305	CLRL	-(SP)	
		68		08	FB	00307	CALLS	#8, SYSS\$GETDVIW	
		8F	50	A3	D1	0030A	CMPL	MAIL\$L_INFCLASS, #66	1164
				03	12	00312	BNEQ	18\$	
		63		04	88	00314	BISB2	#4, MAIL\$GL_FLAGS	1165
		8F	58	A3	D1	00317	CMPL	MAIL\$L_OUTCLASS, #66	1167
				0E	12	0031F	BNEQ	19\$	
		63		08	88	00321	BISB2	#8, MAIL\$GL_FLAGS	1169
50	5D	01		04	EF	00324	EXTZV	#4, #1, MAIL\$L_OUTDEVDP+1, R0	1170
63		04		50	F0	0032A	INSV	R0, #4, #1, MAIL\$GL_FLAGS	
		60	5F	A3	9A	0032F	MOVZBL	MAIL\$L_OUTDEVDP+3, MAIL\$L_OUTLINES	1174
		01		01	E0	00334	BBS	#1, MAIL\$GL_FLAGS+1, 20\$	1175
				08	12	00339	BNEQ	21\$	1176
		60	7FFFFFFF	8F	D0	0033B	MOVL	#2147483647, MAIL\$L_OUTLINES	1177
		00000000G	00	00	FB	00343	CALLS	#0, MAIL\$OPEN_UAF	1181
		07		50	E9	0034A	BLBC	STATUS, 22\$	
		00000000G	00	00	FB	0034D	CALLS	#0, MAIL\$OPEN_SYSMAL	1182
		01		50	E8	00354	BLBS	STATUS, 23\$	
				04	00357		RET		
		3C	AE	8F	9A	00358	MOVZBL	#255, DESC	1186
		40	AE	14	A3	D0	MOVL	MAIL\$GL_BIGBUFFER, DESC+4	1187
				24	A3	DD	PUSHL	MAIL\$L_MYPID	1190
				40	AE	9F	PUSHAB	DESC	1188
				44	AE	9F	PUSHAB	DESC	
				26	A3	B5	TSTW	MAIL\$L_MYPID+2	

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

I 1
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2

Page 52
(18)

MA
V0

			07	13	0036E	BEQL	24\$		
			C4	9E	00370	MOVAB	TMPFILE_FAO_CDESC, R0		
			05	11	00375	BRB	25\$		
		50	C4	9E	00377	MOVAB	TMPFILE_FAO_DESC, R0		
			50	DD	0037C	PUSHL	R0		
		69		04	FB	0037E	CALLS	#4, SYSS\$FAO	
			3C	AE	9F	00381	PUSHAB	DESC	1191
			016C	C5	9F	00384	PUSHAB	TMPFILE_DESC	
		6A		02	FB	00388	CALLS	#2, STR\$COPY_DX	
	3C	AE	FF	8F	9A	0038B	MOVZBL	#255, DESC	1192
	40	AE	14	A3	D0	00390	MOVL	MAIL\$GL_BIGBUFFER, DESC+4	1193
			24	A3	DD	00395	PUSHL	MAIL\$MYPID	1196
			40	AE	9F	00398	PUSHAB	DESC	1194
			44	AE	9F	0039B	PUSHAB	DESC	
			26	A3	B5	0039E	TSTW	MAIL\$MYPID+2	
			07	13	003A1	BEQL	26\$		
		50	FF34	C4	9E	003A3	MOVAB	TMPFILE1_FAO_CDESC, R0	
			05	11	003A8	BRB	27\$		
		50	FF78	C4	9E	003AA	MOVAB	TMPFILE1_FAO_DESC, R0	
			50	DD	003AF	PUSHL	R0		
		69		04	FB	003B1	CALLS	#4, SYSS\$FAO	
			3C	AE	9F	003B4	PUSHAB	DESC	1197
			0174	C5	9F	003B7	PUSHAB	TMPFILE1_DESC	
		6A		02	FB	003BB	CALLS	#2, STR\$COPY_DX	
08		63		01	E1	003BE	BBC	#1, MAIL\$GL_FLAGS, 28\$	1201
	00000000V	00		00	FB	003C2	CALLS	#0, NETJOB	1202
					04	003C9	RET		
			014C	C5	9F	003CA	PUSHAB	EXIT_BLOCK	1206
	00000000G	00		01	FB	003CE	CALLS	#1, SYSS\$DCLEXH	
	FB53	CF		00	FB	003D5	CALLS	#0, MAIL_DISPATCH	1212
					04	003DA	RET		1214
					0000	003DB	.WORD	Save nothing	1015
				7E	D4	003DD	CLRL	-(SP)	
				5E	DD	003DF	PUSHL	SP	
		7E	04	AC	7D	003E1	MOVQ	4(AP), -(SP)	
	00000000G	00		03	FB	003E5	CALLS	#3, MAIL\$HANDLE	
				04	003EC	RET			

; Routine Size: 1005 bytes, Routine Base: \$CODE\$ + 0C10

; 1087 1215 1


```
1089 1216 1 ROUTINE CLI_COMMAND =
1090 1217 1 ++
1091 1218 1 FUNCTIONAL DESCRIPTION:
1092 1219 1
1093 1220 1 Callback the CLI for the rest of the command parameters and send
1094 1221 1 the specified file to the specified users. Then return to the CLI.
1095 1222 1
1096 1223 1 INPUTS:
1097 1224 1
1098 1225 1 INPUT_DESC = CLI request descriptor for input filespec.
1099 1226 1
1100 1227 1 OUTPUTS:
1101 1228 1
1102 1229 1 Input file sent to specified users.
1103 1230 1 --
1104 1231 2 BEGIN
1105 1232 2
1106 1233 2 OWN
1107 1234 2 EXITVALUE : $BBLOCK[4] INITIAL(1);
1108 1235 2
1109 1236 2 ROUTINE HANDLER (SIGARGS,MECHARGS) =
1110 1237 3 BEGIN
1111 1238 3 MAP
1112 1239 3 sigargs : REF $BBLOCK,
1113 1240 3 mechargs : REF $BBLOCK;
1114 1241 3
1115 1242 3 BIND
1116 1243 3 signame = sigargs [chf$l_sig_name] : $BBLOCK; !Name of signal
1117 1244 3
1118 1245 3 IF NOT .signame !If its an error signal
1119 1246 5 AND ((.signame [sts$v_severity] and severity is worse than it was
1120 1247 5 GEQU .exitvalue [sts$v_severity])
1121 1248 4 OR .exitvalue [sts$v_severity]) ! or we haven't had any errors
1122 1249 3 THEN exitvalue = .signame; ! then remember it for exiting
1123 1250 3
1124 1251 3 RETURN SS$_RESIGNAL
1125 1252 2 END;
```

.PSECT \$OWNS\$,NOEXE,2

00000001 00320 EXITVALUE:
.LONG 1

.PSECT \$CODE\$,NOWRT,2

			0004 00000	HANDLER:	WORD	Save R2	: 1236
		52 00000000	00 9E 00002	MOVAB	EXITVALUE, R2		: 1243
50	51	04 AC	04 C1 00009	ADDL3	#4, SIGARGS, R1		: 1245
50	62	12	61 E8 0000E	BLBS	(R1), 2\$		: 1247
	63	03	00 EF 00011	EXTZV	#0, #3, EXITVALUE, R0		:
	61	03	00 ED 00016	CMPZV	#0, #3, (R1), R0		:
			03 1E 0001B	BGEQU	1\$		:
		03	50 E9 0001D	BLBC	R0, 2\$		: 1248

62	0918	61	DO 00020 1\$:	MJVL	(R1), EXITVALUE	:	1249
50		8F	3C 00023 2\$:	MOVZWL	#2328, R0	:	1251
		04	00028	RET		:	1252

: Routine Size: 41 bytes, Routine Base: \$CODE\$ + OFFD

```
: 1126      1253  2
: 1127      1254  2 BUILTIN
: 1128      1255  2   FP;
: 1129      1256  2
: 1130      1257  2 LOCAL
: 1131      1258  2     SELF FLAG,
: 1132      1259  2     TT_INPUT,
: 1133      1260  2     STATUS: LONG;
: 1134      1261  2
: 1135      1262  2 BIND
: 1136      1263  2     NAMDVI = TMP_NAM[NAM$T_DVI] : VECTOR[,BYTE];
: 1137      1264  2
: 1138      1265  2     .FP = HANDLER;
: 1139      1266  2
: 1140      1267  2     ! Open the input file and get the subject
: 1141      1268  2
: 1142      P 1269  2 $FAB_INIT(FAB=TMP_FAB,
: 1143      P 1270  2     DEQ=4,
: 1144      P 1271  2     NAM=TMP_NAM,
: 1145      P 1272  2     DNM='TXT',
: 1146      P 1273  2     CTX=MAIL$ OPENIN,
: 1147      P 1274  2     FNS = .INPUT_DESC[DSC$W_LENGTH],
: 1148      P 1275  2     FNA = .INPUT_DESC[DSC$A_POINTER],
: 1149      P 1276  2     RAT = CR,
: 1150      P 1277  2     RFM = VAR,
: 1151      P 1278  2     FOP = SQO,
: 1152      P 1279  2     FAC = (GET,BRO));
: 1153      1280  2
: 1154      P 1281  2 $NAM_INIT(NAM=TMP_NAM,
: 1155      P 1282  2     RSA=TMP_ESNAM_BUF,
: 1156      P 1283  2     RSS=NAM$C_MAXRSS,
: 1157      P 1284  2     ESA=TMP_ESNAM_BUF,
: 1158      P 1285  2     ESS=NAM$C_MAXRSS);
: 1159      1286  2
: 1160      P 1287  2 IF_ERR( $OPEN(FAB=TMP_FAB,ERR=UTIL$REPORT_IO_ERROR);,
: 1161      1288  2     RETURN .STATUS);
: 1162      1289  2
: 1163      P 1290  2 $RAB_INIT(RAB=TMP_RAB,
: 1164      P 1291  2     FAB=TMP_FAB,
: 1165      P 1292  2     CTX=MAIL$ OPENIN,
: 1166      P 1293  2     UBF=TMP_BOFF,
: 1167      1294  2     USZ=MAIL$K_INBUFSZ);
: 1168      1295  2
: 1169      1296  2 TT_INPUT = 0;
: 1170      1297  2 IF .TMP_FAB[$FAB_DEV(TRM)]
: 1171      1298  3   OR ?TMP_NAM[NAM$V_PPF]
: 1172      1299  3   AND CH$EQL(.SYSIN_DESC[DSC$W_LENGTH],.SYSIN_DESC[DSC$A_POINTER],
: 1173      1300  3     .NAMDVI[0],NAMDVI[1],0))
: 1174      1301  2   THEN TT_INPUT = 1;
: 1175      1302  2 IF NOT .TT_INPUT
```



```
: 1176      1303 3 THEN BEGIN
: 1177      1304 3     TMP_RAB[RAB$V_BIO] = TRUE;
: 1178      1305 3     MAIL$G_CNCT[CNCT_B_FILRAT] = .TMP_FAB[FAB$B_RAT];
: 1179      1306 3     MAIL$G_CNCT[CNCT_B_FILRFM] = .TMP_FAB[FAB$B_RFM];
: 1180      1307 3     END;
: 1181      1308 2
: 1182 P 1309 2 IF_ERR($CONNECT(RAB=TMP_RAB,ERR=UTIL$REPORT_IO_ERROR);,
: 1183      1310 2     RETURN .STATUS);
: 1184      1311 2 TMP_RAB[RAB$LCTX] = MAIL$_READERR;
: 1185      1312 2
: 1186 P 1313 2 IF_ERR(SMG$CREATE_VIRTUAL_KEYBOARD(MAIL$L_SMG_KEYBOARD,SYSIN_DESC);,
: 1187      1314 2     RETURN SIGNAL(.STATUS));
: 1188 P 1315 2 IF_ERR(SMG$CREATE_KEY_TABLE(MAIL$L_SMG_KEYTABLE);,
: 1189      1316 2     RETURN SIGNAL(.STATUS));
: 1190      1317 2
: 1191      1318 2 : Get qualifiers
: 1192      1319 2
: 1193      1320 2 CLISGET_VALUE(SD SUBJECT,MAIL$Q_SUBJ);
: 1194      1321 2 SELF_FLAG = CLIS$PRESENT(SD_SELF);
: 1195      1322 2
: 1196      1323 2 : Get the user names to send to
: 1197      1324 2
: 1198      1325 2 WHILE CLISGET_VALUE(SD_TOLIST,DYN_DESC)
: 1199      1326 3 DO BEGIN
: 1200      1327 3     IF .MAIL$Q_TO[DSC$W_LENGTH] NEQ 0
: 1201      1328 3     THEN STR$APPEND(MAIL$Q_TO,COMMA_DESC);
: 1202      1329 3     STR$APPEND(MAIL$Q_TO,DYN_DESC);
: 1203      1330 3     END;
: 1204      1331 2 IF .MAIL$Q_TO[DSC$W_LENGTH] EQL 0 ! If no users, prompt
: 1205      1332 3 THEN BEGIN
: 1206      1333 4     IF NOT (STATUS = MAIL$GET_INPUT(MAIL$Q_TO,MAIL$GETMSG(MAIL$_ISUB20)))
: 1207      1334 3     THEN RETURN .STATUS;
: 1208      1335 3     MAIL$UPCASE_Q(MAIL$Q_TO);
: 1209      1336 2     END;
: 1210      1337 3 IF (.MAIL$L_USR$SYSBUF[VMD V_SELFSEND] OR .SELF_FLAG)
: 1211      1338 3 AND ((.SELF_FLAG NEQ CLIS$_NEGATED) AND (.SELF_FLAG NEQ CLIS$_LOCNEG))
: 1212      1339 3 THEN BEGIN
: 1213      1340 3     STR$APPEND(MAIL$Q_TO,COMMA_DESC);
: 1214      1341 3     STR$APPEND(MAIL$Q_TO,MAIL$Q_USERNAME);
: 1215      1342 2     END;
: 1216      1343 2
: 1217      1344 2 : Add the usernames to the addressee list and send the file to them.
: 1218      1345 2
: 1219      1346 2 MAIL$INI_ADDR(MAIL$G_CNCT); ! Init addressee list
: 1220      1347 2 MAIL$DELETELINK(MAIL$G_CNCT[CNCT_Q_LNKLST]);
: 1221      1348 2 MAIL$GET_CLUNODES(); !Create cluster node list
: 1222      1349 2 STATUS = CHECK_SEND(
: 1223      1350 2     MAIL$ADD_ADDR(MAIL$Q_TO,MAIL$G_CNCT); ! Add the addresses
: 1224      1351 2 IF NOT .STATUS
: 1225      1352 2     THEN RETURN .EXITVALUE;
: 1226      1353 2
: 1227      1354 2 : Send end of users and to-list
: 1228      1355 2
: 1229      1356 2 MAIL$NET_END_USERS(MAIL$G_CNCT);
: 1230      1357 2
: 1231      1358 2 : if the input file is SYS$INPUT or it's TRM (assume IT:), then copy it
: 1232      1359 2 : to a temporary file
```

```
1233 1360 2 !
1234 1361 2 IF .TT INPUT
1235 1362 3 THEN BEGIN
1236 1363 3 $CLOSE(FAB=TMP_FAB);
1237 1364 3
1238 1365 3 ! Open TMP_FAB to a temporary file
1239 1366 3
1240 P 1367 3 $FAB_INIT(FAB=TMP_FAB,
1241 P 1368 3 DEQ=4,
1242 P 1369 3 FNS=.TMPFILE_DESC[DSC$W_LENGTH],
1243 P 1370 3 FNA=.TMPFILE_DESC[DSC$A_POINTER],
1244 P 1371 3 CTX=MAIL$ OPENOUT,
1245 P 1372 3 NAM=TMP_NAM,
1246 P 1373 3 FOP=(SQD,TMD),
1247 P 1374 3 RFM=VAR,
1248 P 1375 3 RAT=CR,
1249 1376 3 FAC=(GET,PUT,BRO));
1250 1377 3
1251 P 1378 3 $NAM_INIT(NAM=TMP_NAM,
1252 P 1379 3 RSA=TMP_ESNAM_BUF,
1253 P 1380 3 RSS=NAM$C_MAXRSS,
1254 P 1381 3 ESA=TMP_ESNAM_BUF,
1255 1382 3 ESS=NAM$C_MAXRSS);
1256 1383 3
1257 P 1384 3 $RAB_INIT(RAB=TMP_RAB,
1258 P 1385 3 FAB=TMP_FAB,
1259 P 1386 3 CTX=MAIL$ OPENOUT,
1260 P 1387 3 UBF=TMP_BUFF,
1261 1388 3 USZ=MAIL$K_INBUFSZ);
1262 1389 3
1263 P 1390 3 IF_ERR($CREATE(FAB=TMP_FAB,ERR=UTIL$REPORT_IO_ERROR);,
1264 1391 3 RETURN .STATUS);
1265 1392 3
1266 P 1393 3 IF_ERR($CONNECT(RAB=TMP_RAB,ERR=UTIL$REPORT_IO_ERROR);,
1267 P 1394 3 $CLOSE(FAB=TMP_FAB);
1268 1395 3 RETURN .STATUS);
1269 1396 3
1270 1397 3 ! If input is from terminal, assign a channel to it to enable CTRL/C AST's
1271 1398 3
1272 1399 3 IF .MAIL$GL_FLAGS[MAIF V TERM]
1273 1400 3 THEN $ASSIGN(CHAN=MAIL$W_TTCHAN,DEVNAM=MAIL$Q_INPTRAN);
1274 1401 3 MAIL$ENABLE_CTRLC();
1275 1402 3 SMG$DELETE_VIRTUAL_KEYBOARD(MAIL$L_SMG_KEYBOARD); !Delete any previous
1276 P 1403 3 IF_ERR(SMG$CREATE_VIRTUAL_KEYBOARD(MAIL$L_SMG_KEYBOARD,INPUT_DESC);,
1277 1404 3 SIGNAC(.STATUS));
1278 1405 4 IF NOT (STATUS = MAIL$ENTER(TMP_RAB))
1279 1406 3 THEN RETURN .STATUS;
1280 1407 3 MAIL$DISABLE_CTRLC();
1281 1408 2 END;
1282 1409 2 !
1283 1410 2 ! Send the message
1284 1411 2
1285 1412 2 MAIL$SEND_ADDR(MAIL$G_CNCT);
1286 1413 2 RETURN .EXITVALUE ! Return the status
1287 1414 1 END;
```

54 58 54 2E 01026 P.AFH: .ASCII \.TXT\

NAMDVI= TMP_NAM+20
\$RMS_PTR= TMP_FAB
\$RMS_PTR= TMP_NAM
\$RMS_PTR= MAIL\$G_CNCT+646
\$RMS_PTR= TMP_FAB
\$RMS_PTR= TMP_NAM
\$RMS_PTR= MAIL\$G_CNCT+646
.EXTRN SYSS\$CREATE, SYSS\$ASSIGN

OFFC 00000 CLI_COMMAND:

			5B	00000000G	00	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	1216
			5A	00000000G	00	9E	00009	MOVAB	STR\$APPEND, R11	
			59	EFEA	CF	9E	00010	MOVAB	UTIL\$REPORT_IO_ERROR, R10	
			58	00000000'	00	9E	00015	MOVAB	SYSDN_DESC, R9	
			57	00000000'	00	9E	0001C	MOVAB	MAIL\$G_CNCT, R8	
			6D	AD	AF	9E	00023	MOVAB	\$RMS_PTR, R7	1265
0050	8F	00	6E		00	2C	00027	MOVCS	HANDLER, (FP)	1279
							67	0002E		
			67	5003	8F	B0	0002F	MOVW	#20483, \$RMS_PTR	
		04	A7	40	8F	9A	00034	MOVZBL	#64, \$RMS_PTR+4	
		14	A7		04	B0	00039	MOVW	#4, \$RMS_PTR+20	
		16	A7	42	8F	90	0003D	MOVW	#66, \$RMS_PTR+22	
		18	A7	007E109A	8F	D0	00042	MOVL	#8261786, \$RMS_PTR+24	
		1E	A7	0202	8F	B0	0004A	MOVW	#514, \$RMS_PTR+30	
		28	A7	A0	A7	9E	00050	MOVAB	TMP_NAM, \$RMS_PTR+40	
		2C	A7	30F8	C7	D0	00055	MOVL	INPUT_DESC+4, \$RMS_PTR+44	
		30	A7	0FFE	C9	9E	0005B	MOVAB	P.AFH, \$RMS_PTR+48	
		34	A7	00F4	C7	90	00061	MOVW	INPUT_DESC, \$RMS_PTR+52	
0060	8F	00	35		04	90	00067	MOVW	#4, \$RMS_PTR+53	
			6E		00	2C	0006B	MOVCS	#0, (SP), #0, #96, \$RMS_PTR	1285
				A0	A7		00072			
			A0	A7	8F	B0	00074	MOVW	#24578, \$RMS_PTR	
			A2	A7	01	8E	0007A	MNEGB	#1, \$RMS_PTR+2	
			A4	A7	C7	9E	0007E	MOVAB	TMP_ESNAM_BUF, \$RMS_PTR+4	
			AA	A7	01	8E	00084	MNEGB	#1, \$RMS_PTR+10	
			AC	A7	C7	9E	00088	MOVAB	TMP_ESNAM_BUF, \$RMS_PTR+12	
				FE2C	8F	BB	0008E	PUSHR	#*MZR7,R10>	1288
				0480	02	FB	00092	CALLS	#2, SYSS\$OPEN	
		00000000G	00		50	E9	00099	BLBC	STATUS, 4\$	
0044	8F	00	6E		00	2C	0009C	MOVCS	#0, (SP), #0, #68, \$RMS_PTR	1294
				0276	C8		000A3			
			0276	C8	8F	B0	000A6	MOVW	#17409, \$RMS_PTR	
			028E	C8	8F	D0	000AD	MOVL	#8261786, \$RMS_PTR+24	
			0296	C8	8F	B0	000B6	MOVW	#512, \$RMS_PTR+32	
			029A	C8	A8	9E	000BD	MOVAB	TMP_BUFF, \$RMS_PTR+36	
			02B2	C8	67	9E	000C3	MOVAB	TMP_FAB, \$RMS_PTR+60	
					54	D4	000C8	CLRL	TT_INPUT	1296
		15	40	A7	02	E0	000CA	BBS	#2, TMP_FAB+64, 1\$	1297
			14	D6	A7	E9	000CF	BLBC	TMP_NAM+54, 2\$	1298
			51	04	A9	D0	000D3	MOVL	SYSDN_DESC+4, R1	1299
			50	84	A7	9A	000D7	MOVZBL	NAMDVI, R0	1300
50		00	61		69	2D	000DB	CMPCS	SYSDN_DESC, (R1), #0, R0, NAMDVI+1	
				B5	A7		000EO			

Address	Disassembly	Comment	Address	Disassembly	Comment
027B 70	54 0A C8 A8	1E	03 12 000E2	BNEQ	2\$
			01 D0 000E4	MOVL	#1, TT_INPUT
			54 E8 000E7	BLBS	TT_INPUT, 3\$
			08 88 000EA	BISB2	#8, TMP_RAB+5
			A7 B0 000EF	MOVW	TMP_FAB+30, MAIL\$G_CNCT+128
			5A DD 000F4	PUSHL	R10
00000000G	00 01	0276	C8 9F 000F6	PUSHAB	TMP_RAB
			02 FB 000FA	CALLS	#2, SYSS\$CONNECT
			50 E8 00101	BLBS	STATUS, 5\$
			04 00104	RET	
028E	C8 007E10B0		8F D0 00105	MOVL	#8261808, TMP_RAB+24
			59 DD 0010E	PUSHL	R9
		80	A8 9F 00110	PUSHAB	MAIL\$L_SMG_KEYBOARD
00000000G	00 0D		02 FB 00113	CALLS	#2, SMG\$CREATE_VIRTUAL_KEYBOARD
			50 E9 0011A	BLBC	STATUS, 6\$
		84	A8 9F 0011D	PUSHAB	MAIL\$L_SMG_KEYTABLE
00000000G	00 0A		01 FB 00120	CALLS	#1, SMG\$CREATE_KEY_TABLE
			50 E8 00127	BLBS	STATUS, 7\$
			50 DD 0012A	PUSHL	STATUS
00000000G	00		01 FB 0012C	CALLS	#1, LIB\$SIGNAL
			04 00133	RET	
		08	A8 9F 00134	PUSHAB	MAIL\$Q_SUBJ
00000000G	00 01D4		C9 9F 00137	PUSHAB	SD_SUBJECT
			02 FB 0013B	CALLS	#2, CLIS\$GET_VALUE
		01B4	C9 9F 00142	PUSHAB	SD_SELF
00000000G	00 52		01 FB 00146	CALLS	#1, CLIS\$PRESENT
			50 D0 0014D	MOVL	R0, SELF_FLAG
		00E4	C7 9F 00150	PUSHAB	DYN_DESC
		01FC	C9 9F 00154	PUSHAB	SD_TOLIST
00000000G	00 17		02 FB 00158	CALLS	#2, CLIS\$GET_VALUE
			50 E9 0015F	BLBC	R0, 10\$
			68 B5 00162	TSTW	MAIL\$Q_TO
			08 13 00164	BEQL	9\$
		E0	A9 9F 00166	PUSHAB	COMMA_DESC
			58 DD 00169	PUSHL	R8
		68	02 FB 0016B	CALLS	#2, STR\$APPEND
		00E4	C7 9F 0016E	PUSHAB	DYN_DESC
			58 DD 00172	PUSHL	R8
		6B	02 FB 00174	CALLS	#2, STR\$APPEND
			D7 11 00177	BRB	8\$
			68 B5 00179	TSTW	MAIL\$Q_TO
			2A 12 0017B	BNEQ	12\$
00000000G	00 007EDDF0		8F DD 0017D	PUSHL	#8314352
			01 FB 00183	CALLS	#1, MAIL\$GETMSG
			50 DD 0018A	PUSHL	R0
			58 DD 0018C	PUSHL	R8
00000000G	00 56		02 FB 0018E	CALLS	#2, MAIL\$GET_INPUT
	03		50 D0 00195	MOVL	R0, STATUS
			56 E8 00198	BLBS	STATUS, 11\$
		0192	31 0019B	BRW	21\$
00000000G	00		58 DD 0019E	PUSHL	R8
			01 FB 001A0	CALLS	#1, MAIL\$UPCASE_Q
		00000000G	00 D0 001A7	MOVL	MAIL\$L_USR\$SYSBUF, R0
		1F	A0 E8 001AE	BLBS	31(R0), 13\$
			52 E9 001B2	BLBC	SELF_FLAG, 14\$
00000000G	8F		52 D1 001B5	CMPL	SELF_FLAG, #CLIS\$NEGATED
			19 13 001BC	BEQL	14\$

		00000000G	8F		52	D1	001BE	CMPL	SELF_FLAG, #CLIS_LOCNEG		
					10	13	001C5	BEQL	14\$		
				E0	A9	9F	001C7	PUSHAB	COMMA_DESC	1340	
			6B		58	DD	001CA	PUSHL	R8		
					02	FB	001CC	CALLS	#2, STR\$APPEND		
				E8	A8	9F	001CF	PUSHAB	MAIL\$Q_USERNAME	1341	
					58	DD	001D2	PUSHL	R8		
			6B		02	FB	001D4	CALLS	#2, STR\$APPEND		
				F0	A8	9F	001D7	PUSHAB	MAIL\$G_CNCT	1346	
		00000000G	00		01	FB	001DA	CALLS	#1, MAIL\$INI_ADDR		
				20	A8	9F	001E1	PUSHAB	MAIL\$G_CNCT+Z8	1347	
		00000000G	00		01	FB	001E4	CALLS	#1, MAIL\$DELETELINK		
		00000000V	00		00	FB	001EB	CALLS	#0, MAIL\$GET_CLUNODES	1348	
				F0	A8	9F	001F2	PUSHAB	MAIL\$G_CNCT	1350	
					58	DD	001F5	PUSHL	R8		
		00000000G	00		02	FB	001F7	CALLS	#2, MAIL\$ADD_ADDR		
					50	DD	001FE	PUSHL	R0		
		00000000V	00		01	FB	00200	CALLS	#1, CHECK_SEND		
			56		50	DD	00207	MOVL	R0, STATUS		
			03		56	E8	0020A	BLBS	STATUS, 15\$	1351	
					0133	31	0020D	BRW	24\$		
				F0	A8	9F	00210	PUSHAB	MAIL\$G_CNCT	1356	
		00000000G	00		01	FB	00213	CALLS	#1, MAIL\$NET_END_USERS		
			03		54	E8	0021A	BLBS	TT INPUT, 16\$	1361	
					0119	31	0021D	BRW	23\$		
					57	DD	00220	PUSHL	R7	1363	
		00000000G	00		01	FB	00222	CALLS	#1, SYSS\$CLOSE		
0050	8F		00		00	2C	00229	MOVCS	#0, (SP), #0, #80, \$RMS_PTR	1376	
					67		00230				
			67	5003	8F	B0	00231	MOVW	#20483, \$RMS_PTR		
		04	A7	50	8F	9A	00236	MOVZBL	#80, \$RMS_PTR+4		
		14	A7		04	B0	0023B	MOVW	#4, \$RMS_PTR+20		
		16	A7	43	8F	90	0023F	MOVW	#67, \$RMS_PTR+22		
		18	A7	007E10A2	8F	D0	00244	MOVL	#8261794, \$RMS_PTR+24		
		1E	A7	0202	8F	B0	0024C	MOVW	#514, \$RMS_PTR+30		
		28	A7	A0	A7	9E	00252	MOVAB	TMP_NAM, \$RMS_PTR+40		
		2C	A7	00D0	C7	D0	00257	MOVL	TMPFILE_DESC+Z, \$RMS_PTR+44		
0060	8F		00		34	A7	0025D	MOVW	TMPFILE_DESC, \$RMS_PTR+52		
					00	2C	00263	MOVCS	#0, (SPT), #0, #96, \$RMS_PTR	1382	
					A0	A7	0026A				
		A0	A7	6002	8F	B0	0026C	MOVW	#24578, \$RMS_PTR		
		A2	A7		01	8E	00272	MNEGB	#1, \$RMS_PTR+2		
		A4	A7	FE2C	C7	9E	00276	MOVAB	TMP_ESNAM_BUF, \$RMS_PTR+4		
		AA	A7		01	8E	0027C	MNEGB	#1, \$RMS_PTR+10		
0044	8F		00		AC	A7	FE2C	MOVAB	TMP_ESNAM_BUF, \$RMS_PTR+12		
					6E	00	00286	MOVCS	#0, (SP), #0, #68, \$RMS_PTR	1388	
					0276	C8	0028D				
		0276	C8	4401	8F	B0	00290	MOVW	#17409, \$RMS_PTR		
		028E	C8	007E10A2	8F	D0	00297	MOVL	#8261794, \$RMS_PTR+24		
		0296	C8	0200	8F	B0	002A0	MOVW	#512, \$RMS_PTR+32		
		029A	C8	76	A8	9E	002A7	MOVAB	TMP_BUFF, \$RMS_PTR+36		
		02B2	C8		67	9E	002AD	MOVAB	TMP_FAB, \$RMS_PTR+60		
				0480	8F	BB	002B2	PUSHR	#*MZR7, R10>	1391	
		00000000G	00		02	FB	002B6	CALLS	#2, SYSS\$CREATE		
			01		50	E8	002BD	BLBS	STATUS, 17\$		
					04	002C0	RET				
					5A	DD	002C1	PUSHL	R10	1395	

			0276	C8	9F	002C3	PUSHAB	TMP_RAB		
	00000000G	00		02	FB	002C7	CALLS	#2, SYSSCONNECT		
		52		50	DD	002CE	MOVL	R0, STATUS		
		0D		52	EB	002D1	BLBS	STATUS, 18\$		
				57	DD	002D4	PUSHL	R7		
	00000000G	00		01	FB	002D6	CALLS	#1, SYSCLOSE		
		50		52	DD	002DD	MOVL	STATUS, R0		
				04	002E0	RET				
OF	FF78	C8		02	E1	002E1	BBC	#2, MAIL\$GL_FLAGS, 19\$	1399	
				7E	7C	002E7	CLRQ	-(SP)	1400	
			E6	A8	9F	002E9	PUSHAB	MAIL\$W_TTCHAN		
			C0	A8	9F	002EC	PUSHAB	MAIL\$Q_INPTRAN		
	00000000G	00		04	FB	002EF	CALLS	#4, SYSSASSIGN		
	EF4C	CF		00	FB	002F6	CALLS	#0, MAIL\$ENABLE_CTRLC	1401	
			80	A8	9F	002FB	PUSHAB	MAIL\$L_SMG_KEYBOARD	1402	
	00000000G	00		01	FB	002FE	CALLS	#1, SMG\$DELETE_VIRTUAL_KEYBOARD		
			00F4	C7	9F	00305	PUSHAB	INPUT_DESC	1404	
			P0	A8	9F	00309	PUSHAB	MAIL\$C_SMG_KEYBOARD		
	00000000G	00		02	FB	0030C	CALLS	#2, SMG\$CREATE_VIRTUAL_KEYBOARD		
		09		50	EB	00313	BLBS	STATUS, 20\$		
				50	DD	00316	PUSHL	STATUS		
	00000000G	00		01	FB	00318	CALLS	#1, LIB\$SIGNAL		
			0276	C8	9F	0031F	PUSHAB	TMP_RAB	1405	
	00000000G	00		01	FB	00323	CALLS	#1, MAIL\$ENTER		
		56		50	DD	0032A	MOVL	R0, STATUS		
		04		56	EB	0032D	BLBS	STATUS, 22\$		
		50		56	DD	00330	MOVL	STATUS, R0	1406	
				04	00333	RET				
	EEE1	CF		00	FB	00334	CALLS	#0, MAIL\$DISABLE_CTRLC	1407	
			F0	A8	9F	00339	PUSHAB	MAIL\$G_CNCT	1412	
	00000000G	00		01	FB	0033C	CALLS	#1, MAIL\$SEND_ADDR		
		50	00FC	C7	DD	00343	MOVL	EXITVALUE, R0	1413	
				04	00348	RET			1414	

; Routine Size: 841 bytes, Routine Base: \$CODE\$ + 102A


```
1289 1415 1 ROUTINE NETJOB =
1290 1416 1
1291 1417 1 ++
1292 1418 1 FUNCTIONAL DESCRIPTION:
1293 1419 1
1294 1420 1 If this is a network job, it doesn't process commands, rather it
1295 1421 1 is just a slave to send mail to some addressees that are at a different
1296 1422 1 network node than the sender. This routine reads the sender's and
1297 1423 1 addressee's names from the master mail job and sends the mail to
1298 1424 1 the addressees. It then just exits.
1299 1425 1
1300 1426 1 INPUTS:
1301 1427 1
1302 1428 1 MAIL$Q_INPTRAN = SYSS$NET logical name translation.
1303 1429 1
1304 1430 1 OUTPUTS:
1305 1431 1
1306 1432 1 Mail message sent to specified addressee's.
1307 1433 1 --
1308 1434 2 BEGIN
1309 1435 2 OWN
1310 1436 2 DESC: DYN_DESC_DECL; !dynamic descriptor
1311 1437 2
1312 1438 2 BUILTIN
1313 1439 2 FP;
1314 1440 2
1315 1441 2 LOCAL
1316 1442 2 PTR : REF VECTOR[,BYTE], ! String pointer
1317 1443 2 COUNT,
1318 1444 2 PURGE_VEC : VECTOR[2, LONG],
1319 1445 2 FORWARD_ERROR, ! Set if forwarding loop error
1320 1446 2 STATUS; ! Status value
1321 1447 2
1322 1448 2 Get value of /PROTOCOL qualifier in case inbound foreign net
1323 1449 2
1324 1450 2 .FP = IGNORE SIGNALS; !If not there, just ignore it
1325 1451 2 IF NOT CL$GET VALUE(SD_PROTOCOL, MAIL$Q_PROTOCOL)
1326 1452 2 THEN STR$FREE1_DX(MAIL$Q_PROTOCOL);
1327 1453 2 .FP = 0;
1328 1454 2
1329 1455 2 Get sender's name
1330 1456 2
1331 1457 2 INIT_DDESC(MAIL$Q_USERNAME); ! Create dynamic descriptor
1332 1458 2 MAIL$GL_FLAGS[MAIF_V_SERVERLOOP] = 1; ! Loop by default
1333 1459 2
1334 1460 2 Begin looping. Will exit loop when master disconnects the link.
1335 1461 2
1336 1462 2 WHILE TRUE
1337 1463 3 DO BEGIN
1338 1464 3 MAIL$G_CNCT[CNCT_B_FILRAT] = FAB$M_CR;
1339 1465 3 MAIL$G_CNCT[CNCT_B_FILRFM] = FAB$C_VAR;
1340 1466 3 MAIL$INI_ADDR(MAIL$G_CNCT); ! Initialize the list
1341 1467 3 MAIL$DELETELINK(MAIL$G_CNCT[CNCT_Q_LNKLST]);
1342 1468 3
1343 1469 3 This first call will also create the logical link
1344 1470 3
1345 1471 3 IF_ERR(MAIL$GET_INPUT(MAIL$Q_USERNAME, LNK_C_IN_SENDER);, RETURN TRUE);
```

```
: 1346 1472 3 |
: 1347 1473 3 | Prevent an accvio, later will check for null username
: 1348 1474 3 |
: 1349 1475 3 | IF .MAIL$Q_USERNAME[DSC$W_LENGTH] EQL 0
: 1350 1476 3 | THEN MAIL$Q_USERNAME[DSC$A_POINTER] = MAIL$Q_USERNAME;
: 1351 1477 3 |
: 1352 1478 3 | Check to see if we have been at this node before. If so, signal
: 1353 1479 3 | the error and return. Note the check is done before the sender
: 1354 1480 3 | node is put on to allow STAR::ME to send to STAR::YOU
: 1355 1481 3 |
: 1356 1482 3 | FORWARD_ERROR = 0;
: 1357 1483 3 | COUNT = .MAIL$Q_USERNAME[DSC$W_LENGTH];
: 1358 1484 3 | PTR = .MAIL$Q_USERNAME[DSC$A_POINTER];
: 1359 1485 3 | IF .MAIL$Q_SYSNODE2[DSC$W_LENGTH] NEQ 0 .Should not happen, but...
: 1360 1486 3 | THEN INCRU I FROM 1 TO 10
: 1361 1487 4 | DO BEGIN
: 1362 1488 4 |     IF CH$FAIL(PTR = CH$FIND_SUB(.COUNT, PTR,
: 1363 1489 4 |         .MAIL$Q_SYSNODE2[DSC$W_LENGTH],
: 1364 1490 4 |         .MAIL$Q_SYSNODE2[DSC$A_POINTER]))
: 1365 1491 4 |     THEN EXITLOOP
: 1366 1492 5 |     ELSE BEGIN
: 1367 1493 7 |         IF ((PTR NEQ .MAIL$Q_USERNAME[DSC$A_POINTER])
: 1368 1494 6 |             AND (.PTR[-1] NEQ '%'))
: 1369 1495 5 |             THEN EXITLOOP;
: 1370 1496 5 |         PTR = .PTR + .MAIL$Q_SYSNODE2[DSC$W_LENGTH];
: 1371 1497 5 |         COUNT = .COUNT - .MAIL$Q_SYSNODE2[DSC$W_LENGTH];
: 1372 1498 5 |         IF .I EQL 10
: 1373 1499 5 |             THEN FORWARD_ERROR = 1;
: 1374 1500 4 |         END;
: 1375 1501 3 |     END;
: 1376 1502 3 |
: 1377 1503 3 | Add sender's username to the node name, creating a full sender's name
: 1378 1504 3 | unless the sender's node name and our node are the same node
: 1379 1505 3 |
: 1380 1506 3 | IF .MAIL$Q_USERNAME[DSC$W_LENGTH] NEQ 0 THEN
: 1381 1507 4 | BEGIN
: 1382 1508 4 | IF NOT CH$FAIL(PTR = CH$FIND_CH(.MAIL$Q_INPTRAN[DSC$W_LENGTH], ! find end of node name
: 1383 1509 4 |     .MAIL$Q_INPTRAN[DSC$A_POINTER],
: 1384 1510 4 |     ':'))
: 1385 1511 5 | THEN BEGIN
: 1386 1512 5 |     MAIL$Q_INPTRAN[DSC$W_LENGTH] = .PTR+2-.MAIL$Q_INPTRAN[DSC$A_POINTER];
: 1387 1513 5 |     IF CH$FAIL(CH$FIND_SOB(.MAIL$Q_USERNAME[DSC$W_LENGTH],
: 1388 1514 5 |         .MAIL$Q_USERNAME[DSC$A_POINTER],
: 1389 1515 5 |         .MAIL$Q_SYSNODE2[DSC$W_LENGTH],
: 1390 1516 5 |         .MAIL$Q_SYSNODE2[DSC$A_POINTER]))
: 1391 1517 5 |     THEN STR$CONCAT(MAIL$Q_USERNAME,MAIL$Q_INPTRAN,MAIL$Q_USERNAME);
: 1392 1518 4 |     END;
: 1393 1519 3 | END;
: 1394 1520 3 |
: 1395 1521 3 | Also fill in MAIL$Q_FULLUSER for NOTIFY
: 1396 1522 3 |
: 1397 1523 3 | STR$COPY_DX(MAIL$Q_FULLUSER,MAIL$Q_USERNAME);
: 1398 1524 3 |
: 1399 1525 3 | Get the addressee names and enter them in addressee list
: 1400 1526 3 |
: 1401 1527 3 | WHILE 1 DO
: 1402 1528 4 | BEGIN
```



```
: 1403 P 1529 4 IF_ERR(MAIL$GET_INPUT(DESC,LNK_C_IN_CHKUSER);, ! Get addressee name
: 1404 1530 4 RETURN TRUE);
: 1405 1531 4 IF .DESC[DSC$W_LENGTH] EQL 0 ! Check for 0-length name
: 1406 1532 5 THEN BEGIN
: 1407 1533 5 SIGNAL(MAIL$_USERSPEC,1,DESC); ! Signal the error
: 1408 1534 5 EXITLOOP;
: 1409 1535 4 END;
: 1410 1536 4 IF CH$RCHAR(.DESC[DSC$A_POINTER]) EQL 0 ! If null name,
: 1411 1537 4 THEN ! Then all done
: 1412 1538 4 EXITLOOP;
: 1413 1539 4
: 1414 1540 4 ! If we detected a forwarding error, then send the
: 1415 1541 4 status back. Otherwise add the addressee to the list
: 1416 1542 4
: 1417 1543 4 IF .FORWARD_ERROR
: 1418 1544 5 THEN BEGIN
: 1419 1545 5 STATUS = MAIL$_FORWLOOP;
: 1420 1546 5 SIGNAL(.STATUS,1,DESC);
: 1421 1547 5 END
: 1422 1548 4 ELSE STATUS = MAIL$ADD_ADDR(DESC,MAIL$G_CNCT);! Add addressee to list
: 1423 1549 4 IF .STATUS ! If success,
: 1424 1550 4 THEN
: 1425 1551 4 SIGNAL(.STATUS); ! Signal the success
: 1426 1552 3 END;
: 1427 1553 3
: 1428 1554 3 ! Send the end-of-users flag and the to-list to any remote nodes
: 1429 1555 3
: 1430 1556 3
: 1431 1557 3 ! Get the To: and Subj: lists. There is a bit of 'behind the back' error
: 1432 1558 3 handling going on here. If the master mail job decides not to send,
: 1433 1559 3 the logical link will be broken and one or the other of the following
: 1434 1560 3 MAIL$GET_INPUT calls will fail, causing this job to exit.
: 1435 1561 3
: 1436 P 1562 3 IF_ERR( MAIL$GET_INPUT(MAIL$Q_TO,LNK_C_IN_TO);,
: 1437 1563 3 RETURN TRUE);
: 1438 1564 3 MAIL$NET_END_USERS(MAIL$G_CNCT);
: 1439 1565 3
: 1440 P 1566 3 IF_ERR( MAIL$GET_INPUT(MAIL$Q_SUBJ,LNK_C_IN_SUBJ);,
: 1441 1567 3 RETURN TRUE);
: 1442 1568 3 IF .MAIL$Q_SUBJ[DSC$W_LENGTH] NEQ 0
: 1443 1569 3 THEN IF CH$FAIL(CH$FIND_NOT_CH(.MAIL$Q_SUBJ[DSC$W_LENGTH],
: 1444 1570 3 .MAIL$Q_SUBJ[DSC$A_POINTER],%C' '))
: 1445 1571 3 THEN STR$FREE1_DX(MAIL$Q_SUBJ);
: 1446 1572 3
: 1447 1573 3 ! Create a temporary file to store the message text
: 1448 1574 3
: 1449 P 1575 3 $FAB_INIT(FAB=TMP_FAB,
: 1450 P 1576 3 NAM=TMP_NAM,
: 1451 P 1577 3 DNM='MAIL.TXT',
: 1452 P 1578 3 DEQ=3,
: 1453 P 1579 3 CTX=MAIL$_OPENOUT,
: 1454 P 1580 3 RAT = CR,
: 1455 P 1581 3 RFM = VAR,
: 1456 P 1582 3 FOP = (TMD),
: 1457 1583 3 FAC = (GET,PUT,BRO));
: 1458 1584 3
: 1459 1585 3 IF .MAIL$G_CNCT[CNCT_V_BLKMODE] !Did sender say block mode?
```

```
: 1460      1586 4 THEN BEGIN
: 1461      1587 4     TMP_FAB[FAB$B_RAT] = .MAIL$G_CNCT[CNCT_B_FILRAT];    !Get record attributes
: 1462      1588 4     TMP_FAB[FAB$B_RFM] = .MAIL$G_CNCT[CNCT_B_FILRFM];    ! and format
: 1463      1589 3     END;
: 1464      1590 3
: 1465      P 1591 3     $NAM_INIT(NAM=TMP_NAM,
: 1466      P 1592 3         RSA=TMP_ESNAM_BUF,
: 1467      P 1593 3         RSS=NAM$C_MAXRSS,
: 1468      P 1594 3         ESA=TMP_ESNAM_BUF,
: 1469      1595 3         ESS=NAM$C_MAXRSS);
: 1470      1596 3
: 1471      1597 3     MAIL$G_CNCT[CNCT_L_FILERR] = 1;
: 1472      P 1598 3     IF_ERRT $CREATE(FAB=TMP_FAB);
: 1473      P 1599 3         MAIL$G_CNCT[CNCT_L_FILERR] = .TMP_FAB[FAB$L_CTX];
: 1474      P 1600 3         MAIL$G_CNCT[CNCT_L_FILESTS] = .TMP_FAB[FAB$[STV]];
: 1475      1601 3         MAIL$G_CNCT[CNCT_L_FILESTV] = .TMP_FAB[FAB$[STV]];
: 1476      1602 3
: 1477      P 1603 3     $RAB_INIT(RAB=TMP_RAB,
: 1478      P 1604 3         FAB=TMP_FAB,
: 1479      P 1605 3         CTX=MAIL$[OPENOUT,
: 1480      P 1606 3         UBF=TMP_BUFF,
: 1481      1607 3         USZ=MAIL$[INBUFFSZ]);
: 1482      1608 3
: 1483      1609 3     IF .MAIL$G_CNCT[CNCT_V_BLKMODE]    !If block mode, set bio in rab
: 1484      1610 3     THEN TMP_RAB[RAB$V_BIO] = TRUE;
: 1485      1611 3
: 1486      1612 3     IF .MAIL$G_CNCT[CNCT_L_FILERR]
: 1487      P 1613 3     THEN IF ERR($CONNECT(RAB=TMP_RAB);
: 1488      P 1614 3         MAIL$G_CNCT[CNCT_L_FILERR] = .TMP_RAB[RAB$L_CTX];
: 1489      P 1615 3         MAIL$G_CNCT[CNCT_L_FILESTS] = .TMP_RAB[RAB$[STV]];
: 1490      1616 3         MAIL$G_CNCT[CNCT_L_FILESTV] = .TMP_RAB[RAB$[STV]];
: 1491      1617 3
: 1492      1618 3     TMP_RAB[RAB$L_CTX] = MAIL$[WRITEERR];
: 1493      1619 3
: 1494      1620 3     ! Get the message text and store in the temporary file
: 1495      1621 3
: 1496      1622 3     MAIL$G_CNCT[CNCT_L_NRECS] = 0;
: 1497      1623 3     IF NOT .MAIL$GL_FLAGS[MAIF_V_ALTP]
: 1498      1624 4     THEN BEGIN
: 1499      1625 4
: 1500      1626 4         ! Shrink working set while waiting for the first record. Usually
: 1501      1627 4         ! this takes a long time (while the person at the other end enters
: 1502      1628 4         ! the message).
: 1503      1629 4
: 1504      1630 4         PURGE_VEC[0] = 0;
: 1505      1631 4         PURGE_VEC[1] = '7FFFFFFFFF';
: 1506      1632 4         $PURGE(INADR=PURGE_VEC);
: 1507      1633 4         STATUS = MAIL$GET_INPUT(DESC);
: 1508      1634 5         UNTIL ((.DESC[DSC$W_LENGTH] EQL 1) AND
: 1509      1635 5             (CH$RCHAR(.DESC[DSC$A_POINTER]) EQL 0))
: 1510      1536 4         OR NOT .STATUS
: 1511      1637 5         DO BEGIN
: 1512      1638 5             TMP_RAB[RAB$W_RSZ] = .DESC[DSC$W_LENGTH];
: 1513      1639 5             TMP_RAB[RAB$L_RBF] = .DESC[DSC$A_POINTER];
: 1514      1640 5             IF .MAIL$G_CNCT[CNCT_L_FILERR]
: 1515      1641 6             THEN IF NOT (IF .TMP_RAB[RAB$V_BIO] !Use $write if block io
: 1516      1642 7                 THEN $WRITE(RAB=TMP_RAB)
```

```
1517 1643 6 ELSE $PUT(RAB=TMP_RAB))! Put record in file
1518 1644 6 THEN BEGIN
1519 1645 6 MAIL$G_CNCT[CNCT_L_FILEERR] = .TMP_RAB[RAB$L_CTX];
1520 1646 6 MAIL$G_CNCT[CNCT_L_FILESTS] = .TMP_RAB[RAB$L_STS];
1521 1647 6 MAIL$G_CNCT[CNCT_L_FILESTV] = .TMP_RAB[RAB$L_STV];
1522 1648 6 END
1523 1649 5 ELSE IF NOT .TMP_RAB[RAB$V_BIO]
1524 1650 5 THEN MAIL$G_CNCT[CNCT_L_NRECS] = .MAIL$G_CNCT[CNCT_L_NRECS] + 1;
1525 1651 5
1526 1652 5 STATUS = MAIL$GET_INPUT(DESC); ! Get next record
1527 1653 4 END;
1528 1654 4 END
1529 1655 3 ELSE STATUS = MAIL$READ_FOREIGN_FILE(TMP_RAB);
1530 1656 3 TMP_RAB[RAB$L_CTX] = MAIL$_READERR; !Now we read
1531 1657 3
1532 1658 3 ! If success, send the message. Errors writing the temp file will be
1533 1659 3 ! reported once per addressee
1534 1660 3
1535 1661 3 IF .STATUS
1536 1662 3 THEN MAIL$SEND_ADDR(MAIL$G_CNCT)
1537 1663 3 ELSE SIGNAL(.STATUS);
1538 1664 3
1539 1665 3 IF NOT .MAIL$GL_FLAGS[MAIF_V_SERVERLOOP]
1540 1666 3 THEN EXITLOOP;
1541 1667 3
1542 1668 2 END; !Looping until master disconnects
1543 1669 2 RETURN TRUE
1544 1670 1 END;
```

```
54 58 54 2E 4C 49 41 4D 01373 P.AFI: .ASCII \MAIL.TXT\
.PSECT $OWNS,NOEXE,2
```

```
02 00# 00324 DESC: .BYTE 0[2]
0E 00326 .BYTE 14, 2
00328 .BLKB 4
```

```
$RMS_PTR= TMP_FAB
$RMS_PTR= TMP_NAM
$RMS_PTR= MAIL$G_CNCT+646
.EXTRN SYSS$PURGWS, SYSS$WRITE
.EXTRN SYSS$PUT
```

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

```
OFFC 00000 NETJOB: .WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 : 1415
5E 0C C2 00002 .SUBL2 #12, SP : 1450
6D F025 CF 9E 00005 .MOVAB IGNORE_SIGNALS, (FP) : 1451
00000000' 00 00 9F 0000A .PUSHAB MAIL$Q_PROTOCOL
EE31 CF 9F 00010 .PUSHAB SD_PROTOCOL
00000000G 00 02 FB 00014 .CALLS #2, CLISGET_VALUE
0D 50 E8 0001B .BLBS R0, 1$
00000000' 00 00 9F 0001E .PUSHAB MAIL$Q_PROTOCOL : 1452
00000000G 00 01 FB 00024 .CALLS #1, STR$FREE1_DX
6D D4 0002B 1$: .CLRL (FP) : 1453
```

	00000000'	00	020E0000	8F	D0	0002D	MOVL	#34471936, MAIL\$Q_USERNAME	1457
			00000000'	00	D4	00038	CLRL	MAIL\$Q_USERNAME+4	
	00000000'	00		04	88	0003E	BISB2	#4, MAIL\$GL_FLAGS+2	1458
	00000000'	00	0202	8F	B0	00045	MOVW	#514, MAIL\$G_CNCT+128	1464
			00000000'	00	9F	0004E	PUSHAB	MAIL\$G_CNCT	1466
	00000000G	00		01	FB	00054	CALLS	#1, MAIL\$INI_ADDR	
			00000000'	00	9F	0005B	PUSHAB	MAIL\$G_CNCT+Z8	1467
	00000000G	00		01	FB	00061	CALLS	#1, MAIL\$DELETELINK	
			00000000'	09	DD	00068	PUSHL	#9	1471
			00000000'	00	9F	0006A	PUSHAB	MAIL\$Q_USERNAME	
	00000000G	00		02	FB	00070	CALLS	#2, MAIL\$GET_INPUT	
		03		50	E8	00077	BLBS	STATUS, 3\$	
			00000000'	0408	31	0007A	BRW	38\$	
			00000000'	00	B5	0007D	TSTW	MAIL\$Q_USERNAME	1475
				08	12	00083	BNEQ	4\$	
	00000000'	00	00000000'	00	9E	00085	MOVAB	MAIL\$Q_USERNAME, MAIL\$Q_USERNAME+4	1476
				6E	D4	00090	CLRL	FORWARD_ERROR	1482
		57	00000000'	00	3C	00092	MOVZWL	MAIL\$Q_USERNAME, R7	1483
		58		57	D0	00099	MOVL	R7, COUNT	
		56	00000000'	00	D0	0009C	MOVL	MAIL\$Q_USERNAME+4, R6	1484
		58		56	D0	000A3	MOVL	R6, PTR	
		54	00000000'	00	3C	000A6	MOVZWL	MAIL\$Q_SYSNODE2, R4	1485
				3C	13	000AD	BEQL	9\$	
		59	00000000'	00	D0	000AF	MOVL	MAIL\$Q_SYSNODE2+4, R9	1490
		55		01	D0	000B6	MOVL	#1, I	
68		69		54	39	000B9	MATCHC	R4, (R9), COUNT, (PTR)	
				03	13	000BE	BEQL	6\$	
		53		54	D0	000C0	MOVL	R4, R3	
		53		54	C2	000C3	SUBL2	R4, R3	
		58		53	D0	000C6	MOVL	R3, PTR	
				20	13	000C9	BEQL	9\$	
		56		58	D1	000CB	CMPL	PTR, R6	1493
				06	13	000CE	BEQL	7\$	
		3A	FF	A8	91	000D0	CMPB	-1(PTR), #58	1494
				15	12	000D4	BNEQ	9\$	
		58		54	C0	000D6	ADDL2	R4, PTR	1496
		58		54	C2	000D9	SUBL2	R4, COUNT	1497
		0A		55	D1	000DC	CMPL	I, #10	1498
				03	12	000DF	BNEQ	8\$	
		6E		01	D0	000E1	MOVL	#1, FORWARD_ERROR	1499
				55	D6	000E4	INCL	I	1486
		0A		55	D1	000E6	CMPL	I, #10	
				CE	1B	000E9	BLEQU	5\$	
				57	D5	000EB	TSTL	R7	1506
				53	13	000ED	BEQL	12\$	
		52	00000000'	00	D0	000EF	MOVL	MAIL\$Q_INPTRAN+4, R2	1509
62	00000000'	00		3A	3A	000F6	LOCC	#58, MAIL\$Q_INPTRAN, (R2)	1508
				02	12	000FE	BNEQ	10\$	
				51	D4	00100	CLRL	R1	
		58		51	D0	00102	MOVL	R1, PTR	
				3B	13	00105	BEQL	12\$	1510
		53		52	C3	00107	SUBL3	R2, PTR, R3	1512
	00000000'	00		02	A1	0010B	ADDW3	#2, R3, MAIL\$Q_INPTRAN	
		50	00000000'	00	D0	00113	MOVL	MAIL\$Q_SYSNODE2+4, R0	1516
66		57		54	39	0011A	MATCHC	R4, (R0), R7, (R6)	
				03	13	0011F	BEQL	11\$	
				54	D0	00121	MOVL	R4, R3	

Address	Hex	Asm	Comment
00000000G	00	00000000'	1517
00000000G	00	00000000'	1523
00000000G	00	00000000'	1530
00000000G	00	00000000'	1531
00000000G	00	00000000'	1533
00000000G	00	00000000'	1532
00000000G	50	00000000'	1536
00000000G	1A	00000000'	1543
00000000G	5A	00000000'	1545
00000000G	00	00000000'	1546
00000000G	00	00000000'	1543
00000000G	00	00000000'	1548
00000000G	00	00000000'	1549
00000000G	5A	00000000'	1551
00000000G	8E	00000000'	1527
00000000G	00	00000000'	1563
00000000G	00	00000000'	1564
00000000G	00	00000000'	1567
00000000G	00	00000000'	1568
00000000G	03	00000000'	1570
00000000G	50	00000000'	1569
00000000G	51	00000000'	1570
00000000G	50	00000000'	1570

.....

1. A

0050	8F	00	00000000G	00	00000000'	00	9F 00222	PUSHAB	MAIL\$Q SUBJ	1571
				6E		01	FB 00228	CALLS	#1, STR\$FREE1_DX	
						00	2C 0022F	MOVC5	#0, (SP), #0, #80, \$RMS_PTR	1583
			00000000'	00	00000000'	00	00236			
			00000000'	00	5003	8F	B0 0023B	MOVW	#20483, \$RMS_PTR	
			00000000'	00		10	D0 00244	MOVL	#16, \$RMS_PTR+4	
			00000000'	00		03	B0 0024B	MOVW	#3, \$RMS_PTR+20	
			00000000'	00	43	8F	90 00252	MOVW	#67, \$RMS_PTR+22	
			00000000'	00	007E10A2	8F	D0 0025A	MOVL	#8261794, \$RMS_PTR+24	
			00000000'	00	0202	8F	B0 00265	MOVW	#514, \$RMS_PTR+30	
			00000000'	00	00000000'	00	9E 0026E	MOVAB	TMP_NAM, \$RMS_PTR+40	
			00000000'	00	FD7B	CF	9E 00279	MOVAB	P.AF, \$RMS_PTR+48	
			00000000'	00		08	90 00282	MOVW	#8, \$RMS_PTR+53	
		0B	00000000'	00		02	E1 00289	BBC	#2, MAIL\$G_CNCT+132, 23\$	1585
			00000000'	00	00000000'	00	B0 00291	MOVW	MAIL\$G_CNCT+128, TMP_FAB+30	1587
0060	8F	00		6E		00	2C 0029C	MOVC5	#0, (SP), #0, #96, \$RMS_PTR	1595
			00000000'	00	00000000'	00	002A3			
			00000000'	00	6002	8F	B0 002A8	MOVW	#24578, \$RMS_PTR	
			00000000'	00		01	8E 002B1	MNEGB	#1, \$RMS_PTR+2	
			00000000'	00	00000000'	00	9E 002B8	MOVAB	TMP_ESNAM_BUF, \$RMS_PTR+4	
			00000000'	00		01	8E 002C3	MNEGB	#1, \$RMS_PTR+10	
			00000000'	00	00000000'	00	9E 002CA	MOVAB	TMP_ESNAM_BUF, \$RMS_PTR+12	
			00000000'	00		01	D0 002D5	MOVL	#1, MAIL\$G_CNCT+72	1597
			00000000G	00	00000000'	00	9F 002DC	PUSHAB	TMP_FAB	1601
				16		01	FB 002E2	CALLS	#1, SYSS\$CREATE	
			00000000'	00	00000000'	50	E8 002E9	BLBS	STATUS, 24\$	
			00000000'	00	00000000'	00	D0 002EC	MOVL	TMP_FAB+24, MAIL\$G_CNCT+72	
0044	8F	00	00000000'	00	00000000'	00	7D 002F7	MOVQ	TMP_FAB+8, MAIL\$G_CNCT+76	
				6E		00	2C 00302	MOVC5	#0, (SP), #0, #68, \$RMS_PTR	1607
			00000000'	00	00000000'	00	00309			
			00000000'	00	4401	8F	B0 0030E	MOVW	#17409, \$RMS_PTR	
			00000000'	00	007E10A2	8F	D0 00317	MOVL	#8261794, \$RMS_PTR+24	
			00000000'	00	0200	8F	B0 00322	MOVW	#512, \$RMS_PTR+32	
			00000000'	00	00000000'	00	9E 0032B	MOVAB	TMP_BUFF, \$RMS_PTR+36	
			00000000'	00	00000000'	00	9E 00336	MOVAB	TMP_FAB, \$RMS_PTR+60	
		07	00000000'	00		02	E1 00341	BBC	#2, MAIL\$G_CNCT+132, 25\$	1609
			00000000'	00		08	88 00349	BISB2	#8, TMP_RAB+5	1610
				26	00000000'	00	E9 00350	BLBC	MAIL\$G_CNCT+72, 26\$	1612
			00000000G	00	00000000'	00	9F 00357	PUSHAB	TMP_RAB	1616
				16		01	FB 0035D	CALLS	#1, SYSS\$CONNECT	
			00000000'	00	00000000'	50	E8 00364	BLBS	STATUS, 26\$	
			00000000'	00	00000000'	00	D0 00367	MOVL	TMP_RAB+24, MAIL\$G_CNCT+72	
			00000000'	00	00000000'	00	7D 00372	MOVQ	TMP_RAB+8, MAIL\$G_CNCT+76	
			00000000'	00	007E10D0	8F	D0 0037D	MOVL	#8261840, TMP_RAB+24	1618
					00000000'	00	D4 00388	CLR	MAIL\$G_CNCT+100	1622
		03	00000000'	00		02	E1 0038E	BBC	#2, MAIL\$GL_FLAGS+1, 27\$	1623
						00AB	31 00396	BRW	34\$	
				04		AE	D4 00399	CLRL	PURGE_VEC	1630
		08	AE 7FFFFFFF	8F	D0 0039C	AE	D0 0039C	MOVL	#2147483647, PURGE_VEC+4	1631
				04		AE	9F 003A4	PUSHAB	PURGE_VEC	1632
			00000000G	00		01	FB 003A7	CALLS	#1, SYSS\$PURGWS	
					00000000'	00	9F 003AE	PUSHAB	DESC	1633
			00000000G	00		01	FB 003B4	CALLS	#1, MAIL\$GET_INPUT	
				5A		50	D0 003BB	MOVL	R0, STATUS	
				50	00000000'	00	3C 003BE	MOV2WL	DESC, R0	1634
				01		50	B1 003C5	CMPL	R0, #1	
						0B	12 003C8	BNEQ	29\$	

51	00000000'	00	D0	003CA	MOVL	DESC+4, R1	1635
		61	95	003D1	TSTB	(R1)	
		7F	13	003D3	BEQL	35\$	
	7C	5A	E9	003D5	BLBC	STATUS, 35\$	1636
00000000'	00	50	B0	003D8	MOVW	R0, TMP_RAB+34	1638
00000000'	00	00	D0	003DF	MOVL	DESC+4, -TMP_RAB+40	1639
	BD	00	E9	003EA	BLBC	MAIL\$G_CNCT+72, 28\$	1640
12 00000000'	00	03	E1	003F1	BBC	#3, TMP_RAB+5, 30\$	1641
	00000000'	00	9F	003F9	PUSHAB	TMP_RAB	1642
00000000G	00	01	FB	003FF	CALLS	#1, -SYSS\$WRITE	
	12	50	E9	00406	BLBC	R0, 31\$	
		28	11	00409	BRB	32\$	
	00000000'	00	9F	0040B	PUSHAB	TMP_RAB	1643
00000000G	00	01	FB	00411	CALLS	#1, -SYSS\$PUT	
	18	50	E8	00418	BLBS	R0, 32\$	
00000000'	00	00	D0	0041B	MOVL	TMP_RAB+24, MAIL\$G_CNCT+72	1645
00000000'	00	00	7D	00426	MOVQ	TMP_RAB+8, MAIL\$G_CNCT+76	1646
		0E	11	00431	BRB	33\$	1641
06 00000000'	00	03	F0	00433	BBS	#3, TMP_RAB+5, 33\$	1649
	00000000'	00	D6	0043B	INCL	MAIL\$G_CNCT+100	1650
		F6A	31	00441	BRW	28\$	1652
	00000000'	00	9F	00444	PUSHAB	TMP_RAB	1655
00000000G	00	01	FB	0044A	CALLS	#1, -MAIL\$READ_FOREIGN_FILE	
	5A	50	D0	00451	MOVL	R0, STATUS	
00000000'	00	8F	D0	00454	MOVL	#8261808, TMP_RAB+24	1656
	0F	5A	E9	0045F	BLBC	STATUS, 36\$	1661
	00000000'	00	9F	00462	PUSHAB	MAIL\$G_CNCT	1662
00000000G	00	01	FB	00468	CALLS	#1, MAIL\$SEND_ADDR	
		09	11	0046F	BRB	37\$	
		5A	DD	00471	PUSHL	STATUS	1663
00000000G	00	01	FB	00473	CALLS	#1, LIB\$SIGNAL	
03 00000000'	00	02	E1	0047A	BBC	#2, MAIL\$GL_FLAGS+2, 38\$	1665
		FBC0	31	00482	BRW	2\$	
	50	01	D0	00485	MOVL	#1, R0	1669
		04	00488	RET			1670

; Routine Size: 1161 bytes, Routine Base: \$CODE\$ + 137B

```
: 1546      1671 1 ROUTINE INTJOB =
: 1547      1672 1
: 1548      1673 1 ++
: 1549      1674 1 FUNCTIONAL DESCRIPTION:
: 1550      1675 1
: 1551      1676 1         Get the sender's username and default directory spec, check for
: 1552      1677 1         new mail, get commands, parse them and dispatch to the command
: 1553      1678 1         action routines.
: 1554      1679 1
: 1555      1680 1 INPUTS:
: 1556      1681 1
: 1557      1682 1         MAIL$Q_INPTRAN = SYSS$INPUT logical name translation descriptor.
: 1558      1683 1         INPCHAR_DESC = SYSS$INPUT device characteristics buffer descriptor
: 1559      1684 1
: 1560      1685 1 OUTPUTS:
: 1561      1686 1
: 1562      1687 1         Commands parsed and executed.
: 1563      1688 1 --
: 1564      1689 2 BEGIN
: 1565      1690 2
: 1566      1691 2 ROUTINE SWALLOW_SIGNALS(SIGARG,MECHARG) =
: 1567      1692 3 BEGIN
: 1568      1693 3
: 1569      1694 3 MAP
: 1570      1695 3     SIGARG : REF $BBLOCK;
: 1571      1696 3
: 1572      1697 3 BIND
: 1573      1698 3     SIGNAME = SIGARG[CHFS$L_SIG_NAME] : $BBLOCK;
: 1574      1699 3
: 1575      1700 4 RETURN (IF .SIGNAME           !If success, then resignal to print
: 1576      1701 4     THEN SS$_RESIGNAL       ! otherwise, ignore error signals
: 1577      1702 4     ELSE SS$_CONTINUE)
: 1578      1703 4
: 1579      1704 2 END;
```

0000 00000 SWALLOW_SIGNALS:

50	04	AC	04	C1	00002	.WORD	Save nothing	: 1691
		06	60	E9	00007	ADDL3	#4, SIGARG, R0	: 1698
		50	8F	3C	0000A	BLBC	(R0), 1\$	: 1700
				04	0000F	MOVZWL	#2328, R0	:
		50	01	D0	00010	RET		:
				04	00013	MOVL	#1, R0	:
						RET		: 1704

; Routine Size: 20 bytes, Routine Base: \$CODE\$ + 1804

```
: 1580      1705 2 ROUTINE NOTICE_SIGNAL (SIGARG,MECHARG,ENBL) =
: 1581      1706 3 BEGIN
: 1582      1707 3 MAP
: 1583      1708 3     SIGARG : REF $BBLOCK,
: 1584      1709 3     ENBL : REF VECTOR[,LONG];
: 1585      1710 3
```



```
: 1586      1711 3 BIND
: 1587      1712 3     SIGNAME = SIGARG[CHFSL_SIG_NAME] : $BBLOCK,
: 1588      1713 3     RETSTS = ENBL[1] : REF-VECTOR[.LONG];
: 1589      1714 3
: 1590      1715 3 IF .SIGNAME[STSSV_FAC_NO] EQL 3
: 1591      1716 3     THEN RETSTS[0] = T;
: 1592      1717 3 RETURN SS$_RESIGNAL
: 1593      1718 2 END;
```

				0000 00000 NOTICE_SIGNAL:					
				04	C1	00002	.WORD	Save nothing	: 1705
				04	C1	00007	ADDL3	#4, SIGARG, R1	: 1712
				04	C1	00007	ADDL3	#4, ENBL, R0	: 1713
03	02	A1	51	04	ED	0000C	CMPZV	#0, #12, 2(R1), #3	: 1715
			50	04	12	00012	BNEQ	1\$	
			OC	01	D0	00014	MOVL	#1, @0(R0)	: 1716
			AC	8F	3C	00018	MOVZWL	#2328, R0	: 1717
			OC	04	0001D		RET		: 1718
			50						
				0918					

: Routine Size: 30 bytes, Routine Base: \$CODE\$ + 1818

```
: 1594      1719 2 BUILTIN
: 1595      1720 2     FP;
: 1596      1721 2
: 1597      1722 2 LOCAL
: 1598      1723 2     NEWCOUNT,
: 1599      1724 2     NOERASE_FLAG,
: 1600      1725 2     ROWS,
: 1601      1726 2     COLS,
: 1602      1727 2     BLI_HANDLER_ADDR,
: 1603      1728 2     TRN[NMLST : $ITMLST DECL(ITEMS=1),
: 1604      1729 2     PARSE_SIGNALLED : VOLATILE,
: 1605      1730 2     NULLDESC : VECTOR[2,LONG],
: 1606      1731 2     STATUS;
: 1607      1732 2
: 1608      1733 2 ENABLE NOTICE_SIGNAL(PARSE_SIGNALLED);
: 1609      1734 2 BLI_HANDLER_ADDR = ..FP;
: 1610      1735 2 .FP = 0;
: 1611      1736 2 NULLDESC[0] = 0;
: 1612      1737 2 NULLDESC[1] = NULLDESC[0];
: 1613      1738 2
: 1614      1739 2 ! Parse any command qualifiers specified in the DCL command line
: 1615      1740 2 ! (currently only one is /EDIT). This is not done unless invoked
: 1616      1741 2 ! with a "real" DCL command (ie. not RUN MAIL, MCR MAIL, or foreign command)
: 1617      1742 2 ! IF /EDIT with no values, default to /EDIT=(SEND,REPLY)
: 1618      1743 2 !
: 1619      1744 2 IF NOT .CLI_SIGNALLED
: 1620      1745 2     AND CLI$PRESENT(SD_EDIT)
: 1621      1746 3 THEN BEGIN
: 1622      1747 3     WHILE CLI$GET_VALUE(SD_EDIT,DYN_DESC)
: 1623      1748 4     DO IF NOT (STATUS = CALL TP$PARSE(DYN_DESC,
: 1624      1749 4     EDITQUAL_STATE,EDITQUAL_KEY))
```

```

: 1625      1750      3      THEN SIGNAL STOP(.STATUS);
: 1626      1751      3      IF .CLI_EDITQUAL EQ 0
: 1627      1752      3      THEN CLI_EDITQUAL = 1^F SEND_EDIT OR 1^F_REPLY_EDIT;
: 1628      1753      3      CLI_EXTRACT = CLISPRESNT(SD_EXTRACT);
: 1629      1754      2      END;
: 1630      1755      2
: 1631      1756      2
: 1632      1757      2      ! Translate logical name MAIL$EDIT to see if there is a specific edit
: 1633      1758      2      command file to be used. If not, callable EDT will be used
: 1634      1759      2
: 1635      P 1760      2      $ITMLST_INIT(ITMLST=TRNLNMLST,
: 1636      P 1761      2      (ITMCOB=LNMS_STRING,BUFADR=.MAIL$Q_EDITNAM_DESC[1],
: 1637      P 1762      2      BUFSTZ=.MAIL$Q_EDITNAM_DESC[0],
: 1638      1763      2      RETLEN=MAIL$Q_EDITNAM_DESC));
: 1639      1764      2
: 1640      P 1765      3      MAIL$GL_FLAGS[MAIF_V_USEREDIT] = (STATUS = STRNLNM(ATTR=XREF(LNMSM_CASE_BLIND),
: 1641      P 1766      3      TABNAM=SD_LNMSFILE_DEV,
: 1642      P 1767      3      LOGNAM=$DESCRIPTOR('MAIL$EDIT'),
: 1643      1768      2      ITMLST=TRNLNMLST)) EQL SS$_NORMAL;
: 1644      1769      2
: 1645      1770      2
: 1646      1771      2      ! If first character is a @ strip it off
: 1647      1772      2
: 1648      1773      2      IF .MAIL$GL_FLAGS[MAIF_V_USEREDIT]
: 1649      1774      2      THEN
: 1650      1775      2      IF CH$EQL(1,CH$PTR(NULL_CHAR),1,CH$PTR(.MAIL$Q_EDITNAM_DESC[1])) THEN
: 1651      1776      3      BEGIN
: 1652      1777      3      MAIL$Q_EDITNAM_DESC[0] = .MAIL$Q_EDITNAM_DESC[0] - 1;
: 1653      1778      3      MAIL$Q_EDITNAM_DESC[1] = .MAIL$Q_EDITNAM_DESC[1] + 1;
: 1654      1779      3      END;
: 1655      1780      2
: 1656      1781      3      IF (NOT .MAIL$GL_FLAGS[MAIF_V_USEREDIT]) OR (.MAIL$Q_EDITNAM_DESC[0] EQL 0)
: 1657      1782      3      THEN BEGIN
: 1658      1783      3      CH$MOVE(.CALLABLE_EDT[0],CALLABLE_EDT[1],.MAIL$Q_EDITNAM_DESC[1]);
: 1659      1784      3      MAIL$Q_EDITNAM_DESC[0] = .CALLABLE_EDT[0];
: 1660      1785      2      END;
: 1661      1786      2
: 1662      1787      2      MAIL$UPCASE(MAIL$Q_EDITNAM_DESC);
: 1663      1788      2      MAIL$GL_FLAGS[MAIF_V_CALEDIT] = CH$EQL(.CALLABLE_EDT[0],
: 1664      1789      2      .MAIL$Q_EDITNAM_DESC[1],
: 1665      1790      2      .CALLABLE_EDT[0],CALLABLE_EDT[1],0);
: 1666      1791      2      IF .MAIL$GL_FLAGS[MAIF_V_CALEDIT]
: 1667      1792      3      THEN BEGIN
: 1668      1793      3      LOCAL
: 1669      1794      3      DESC : VECTOR[2, LONG];
: 1670      1795      3
: 1671      1796      3      MAP
: 1672      1797      3      SD_TPU : VECTOR[, LONG],
: 1673      1798      3      SD_EDT : VECTOR[, LONG];
: 1674      1799      3
: 1675      1800      3      DESC[0] = .MAIL$Q_EDITNAM_DESC[0] - .CALLABLE_EDT[0];
: 1676      1801      3      DESC[1] = .MAIL$Q_EDITNAM_DESC[1] + .CALLABLE_EDT[0];
: 1677      1802      3      IF .DESC[0] EQL 0
: 1678      1803      4      THEN BEGIN
: 1679      1804      4      DESC[0] = .SD_EDT[0];
: 1680      1805      4      DESC[1] = .SD_EDT[1];
: 1681      1806      3      END;
```

```
1682 1807 3      | Check for 'EDT'
1683 1808 3
1684 1809 3
1685 1810 3      | IF CH$EQL(.SD EDT[0],.SD EDT[1],.DESC[0],.DESC[1])
1686 1811 3      | THEN MAIL$GL_FLAGS[MAIF_V_EDTEDT] = 1;
1687 1812 3      | IF CH$EQL(.SD TPO[0],.SD TPO[1],.DESC[0],.DESC[1])
1688 1813 3      | THEN MAIL$GL_FLAGS[MAIF_V_EDTXYZ] = 1;
1689 1814 3      | STR$COPY_DX(MAIL$Q_CALEDTNAM,DESC);
1690 1815 2      | END;
1691 1816 2
1692 1817 2      | Create the virtual display without erasing screen
1693 1818 2
1694 1819 2      | NOERASE_FLAG=1;
1695 1820 2      | COLS=0;
1696 1821 2      | IF .MAIL$GL_FLAGS[MAIF_V_SCOPE] THEN
P 1822 2      | IF _ERR(SMG$CREATE_PASTEBBOARD(MAIL$L_SMG_DISPLAY,$DESCRIPTOR('SYS$OUTPUT'),
P 1823 2      | ROWS,COLS,NOERASE_FLAG);,
1824 2      | RETURN .STATUS);
1700 1825 2
1701 1826 2      | Create the virtual keyboard and key table
1702 1827 2
P 1828 2      | IF _ERR(SMG$CREATE_VIRTUAL_KEYBOARD(MAIL$L_SMG_KEYBOARD,SYSIN_DESC);,
1704 1829 2      | RETURN SIGNAL(.STATUS));
P 1830 2      | IF _ERR(SMG$CREATE_KEY_TABLE(MAIL$L_SMG_KEYTABLE);,
1706 1831 2      | RETURN SIGNAL(.STATUS));
1707 1832 2
1708 1833 2      | Get user's default device and directory names and concatenate them together
1709 1834 2
P 1835 2      | IF _ERR(MAIL$GET_DEVDIR(MAIL$Q_USERNAME,MAIL$Q_USERDEF);,
1711 1836 2      | RETURN .STATUS);
1712 1837 2
1713 1838 2      | MAIL$GET_DEVDIR reads the uaf record
1714 1839 2
1715 1840 2      | MAIL$L_MYUIC = .MAIL$T_UAF_BUFF[UAF$L_UIC];          ! Get my UIC
1716 1841 2      | STR$CONCAT(MAIL$Q_MAILDEF,MAIL$Q_USERDEF,MAIL$Q_MAIL_MAIL);
1717 1842 2      | MAIL$GL_FLAGS[MAIF_V_CAPTIVE] = .MAIL$T_UAF_BUFF[UAF$V_CAPTIVE];
1718 1843 2
1719 1844 2      | Check for new mail
1720 1845 2
1721 1846 2      | IF NOT .MAIL$GL_FLAGS[MAIF_V_BATCH]                !Unless in BATCH mode
1722 1847 3      | THEN BEGIN
1723 1848 3      |   .FP = SWALLOW SIGNALS;
1724 1849 3      |   IF MAIL$GET_NEWMAIL(MAIL$Q_USERNAME,NEWCOUNT)
1725 1850 3      |       AND .NEWCOUNT NEQ 0
1726 1851 4      |       THEN (MAIL$PUT_OUTPUT(NULLDESC);
1727 1852 4      |           MAIL$PUT_OUTPUT(MAIL$GETMSG(MAIL$Q_YOULHAVENEW),
1728 1853 3      |               .NEWCOUNT));
1729 1854 3      |   .FP = 0;
1730 1855 3
1731 1856 3      | Delete any old MAIL.TMP files
1732 1857 3
1733 1858 3      | MAIL$DEL_TMPF();
1734 1859 2      | END;
1735 1860 2
1736 1861 2      | If input is from terminal, assign a channel to it to enable CTRL/C AST's
1737 1862 2
1738 1863 2      | IF .MAIL$GL_FLAGS[MAIF_V_TERM]
```

```
1739 1864 3 THEN BEGIN
1740 1865 3     $ASSIGN(CHAN=MAIL$W_TTCHAN,DEVNAM=MAIL$Q_INPTRAN);
1741 1866 3     DEFINE_DEFAULT_KEYPAD();
1742 1867 3     LOAD_KEY_DEFS();
1743 1868 3     END;
1744 1869 3
1745 1870 3     Get commands and parse them
1746 1871 3
1747 1872 3 DO BEGIN
1748 1873 3     IF .MAIL$GL_FLAGS[MAIF_V_ITEM]
1749 1874 3     THEN MAIL$PUT_OUTPUT(NULLDESC);
1750 1875 3     IF .ERR( MAIL$GET_INPUT(COMMAND_DESC, ! Get the command
1751 1876 3     DESCRIPTOR(%CHAR(LF),'MAIL> '));,
1752 1877 3     IF .STATUS NEQ RMSS_EOF
1753 1878 3     AND .STATUS NEQ SMGS_EOF
1754 1879 3     THEN SIGNAL(.STATUS);
1755 1880 3     EXITLOOP);
1756 1881 3     STR$TRIM(COMMAND_DESC,COMMAND_DESC);
1757 1882 3     MAIL$UPCASE_Q(COMMAND_DESC); . Uppcase the command
1758 1883 3
1759 1884 3     Check string to see if it's a number. If so, prefix with 'READ'
1760 1885 3
1761 1886 3     IF CALL TPARSE(COMMAND_DESC,DECNUM_STATE,DECNUM_KEY)
1762 1887 3     THEN STR$CONCAT(COMMAND_DESC,
1763 1888 3     $DESCRIPTOR('READ '),COMMAND_DESC);
1764 1889 3     FILE_DESC[DSC$W_LENGTH] = 0; ! Init file name desc
1765 1890 3     CLIFLAGS = 0; ! init qualifier flags
1766 1891 3
1767 1892 3     If length is 0, then assume CONTINUE. Else parse command, and if
1768 1893 3     success, dispatch to process the command
1769 1894 3
1770 1895 3     MAIL$ENABLE_CTRL();
1771 1896 3     IF .COMMAND_DESC[DSC$W_LENGTH] EQL 0
1772 1897 3     THEN MAIL$CMD_CONTINUE()
1773 1898 3     ELSE BEGIN
1774 1899 3     !
1775 1900 3     ! .FP = .BLI_HANDLER_ADDR;
1776 1901 3     ! PARSE_SIGNALLED = 0;
1777 1902 3     ! IF (STATUS = CLISDCL_PARSE(COMMAND_DESC,
1778 1903 3     ! MAIL$COMMAND_TABLE,MAIL$GET_INPUT))
1779 1904 3     ! THEN STATUS = CLISDISPATCH();
1780 1905 3     ! .FP = 0;
1781 1906 3     ! IF NOT .STATUS
1782 1907 3     ! AND NOT .PARSE_SIGNALLED
1783 1908 3     ! THEN SIGNAL(.STATUS); !Signal all errors
1784 1909 3     END;
1785 1910 3     MAIL$G_CNCT[CNCT_B_FILPAT] = FAB$M_CR;
1786 1911 3     MAIL$G_CNCT[CNCT_B_FILRFM] = FAB$C_VAR;
1787 1912 3     MAIL$DELETELINK(MAIL$G_CNCT[CNCT_Q_LNKLST]);
1788 1913 3     MAIL$DISABLE_CTRL();
1789 1914 3     MAIL$GL_FLAGS[MAIF_V_ATTACHMENT] = 0;
1790 1915 2 UNTIL .CLIFLAGS[EXIT_FLAG] OR .CLIFLAGS[QUIT_FLAG];
1791 1916 2 IF NOT .CLIFLAGS[QUIT_FLAG] ! If not QUIT command,
1792 1917 2 THEN
1793 1918 2     MAIL$CLOSE_FILE(); ! Close any open file
1794 1919 2     $CLOSE(FAB = TMP_FAB); ! Close open input file
1795 1920 2 RETURN TRUE
```

```
54 49 44 45 24 4C 49 41 4D 01836 P.AFK: .ASCII \MAIL$EDIT\
0183F .BLKB 1
00000009 01840 P.AFJ: .LONG 9
00000000' 01844 .ADDRESS P.AFK
54 55 50 54 55 4F 24 53 59 53 01848 P.AFM: .ASCII \SYS$OUTPUT\
01852 .BLKB 2
0000000A 01854 P.AFL: .LONG 10
00000000' 01858 .ADDRESS P.AFM
0A 0185C P.AFO: .ASCII <10>
20 3E 4C 49 41 4D 0185D .ASCII \MAIL> \
01863 .BLKB 1
00000007 01864 P.AFN: .LONG 7
00000000' 01868 .ADDRESS P.AFO
20 44 41 45 52 0186C P.AFQ: .ASCII \READ \
01871 .BLKB 3
00000005 01874 P.AFP: .LONG 5
00000000' 01878 .ADDRESS P.AFQ
```

```
OFFC 00000 INTJOB: .WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11
5B 00000000G 00 9E 00002 MOVAB MAIL$PUT_OUTPUT, R11
5A 00C00000' 00 9E 00009 MOVAB COMMAND_DESC, R10
59 E8C8 CF 9E 00010 MOVAB SD_EDIT, R9
58 00000000' 00 9E 00015 MOVAB MAIL$GL_FLAGS, R8
5E 38 C2 0001C SUBL2 #56, SP
24 AE D4 0001F CLRL PARSE_SIGNALLED
6D 0341 CF DE 00022 MOVAL 30$, 7(FP)
50 6D D0 00027 MOVL (FP), BLI_HANDLER_ADDR
6D D4 0002A CLRL (FP)
1C AE D4 0002C CLRL NULLDESC
20 AE 1C AE 9E 0002F MOVAB NULLDESC, NULLDESC+4
5A FE60 CA E8 00034 BLBS CLI_SIGNALLED, 4$
59 DD 00039 PUSHL R9
00000000G 00 01 FB 0003B CALLS #1, CLISPRESENT
4E 50 E9 00042 BLBC R0, 4$
08 AA 9F 00045 1$: PUSHAB DYN_DESC
59 DD 00048 PUSHL R9
00000000G 00 02 FB 0004A CALLS #2, CLISGET_VALUE
25 50 E9 00051 BLBC R0, 2$
00000000' 00 9F 00054 PUSHAB EDITQUAL_KEY
00000000' 00 9F 0005A PUSHAB EDITQUAL_STATE
08 AA 9F 00060 PUSHAB DYN_DESC
EA23 CF 03 FB 00063 CALLS #3, CALL_TPARSE
57 50 D0 00068 MOVL R0, STATUS
D7 57 E8 0006B BLBS STATUS, 1$
57 DD 0006E PUSHL STATUS
00000000G 00 01 FB 00070 CALLS #1, LIB$STOP
CC 11 00077 BKR 1$
FE58 CA D5 00079 2$: TSTL CLI_EDITQUAL
05 12 0007D BNEQ 3$
FE58 CA 03 D0 0007F MOVL #3, CLI_EDITQUAL
CC A9 9F 00084 3$: PUSHAB SD_EXTRACT
```

			00000000G	00	01	FB	00087	CALLS	#1, CLISPRESENT			
			FESC	CA	50	90	0008E	MOV	R0, CLI_EXTRACT			
				50	28	AE	9E	00093	4\$:	MOVAB	TRNLNMLST, \$\$ITMBLKPTR	1763
				80	64	A8	B0	00097		MOVW	MAIL\$Q_EDITNAM_DESC, (\$\$ITMBLKPTR)+	
				80		02	B0	0009B		MOVW	#2, (\$\$ITMBLKPTR)+	
				80	68	A8	D0	0009E		MOVL	MAIL\$Q_EDITNAM_DESC+4, (\$\$ITMBLKPTR)+	
				80	64	A8	9E	000A2		MOVAB	MAIL\$Q_EDITNAM_DESC, (\$\$ITMBLKPTR)+	
						80	D4	000A6		CLRL	(\$\$ITMBLKPTR)+	
					28	AE	9F	000A8		PUSHAB	TRNLNMLST	1768
						7E	D4	000AB		CLRL	-(SP)	
					16E8	C9	9F	000AD		PUSHAB	P.AFJ	
					28	A9	9F	000B1		PUSHAB	SD_LNMSFILE_DEV	
			10	AE	02000000	8F	D0	000B4		MOVL	#33554432, T6(SP)	
					10	AE	9F	000BC		PUSHAB	16(SP)	
			00000000G	00	05	FB	000BF	CALLS	#5, SYS\$TRNLNM			
				57	50	D0	000C6	MOVL	R0, STATUS			
					50	D4	000C9	CLRL	R0			
				01	57	D1	000CB	CMPL	STATUS, #1			
					02	12	000CE	BNEQ	5\$			
					50	D6	000D0	INCL	R0			
01	A8			05	50	F0	000D2	5\$:	INSV	R0, #5, #1, MAIL\$GL_FLAGS+1		
	51			01	05	EF	000D8		EXTZV	#5, #1, MAIL\$GL_FLAGS+1, R1	1773	
				19	51	E9	000DE		BLBC	R1, 7\$		
				50	68	A8	D0	000E1	MOVL	MAIL\$Q_EDITNAM_DESC+4, R0	1775	
				60	FEA8	C9	91	000E5	CMPL	NULL_CHAR, (R0)		
						06	12	000EA	BNEQ	6\$		
					64	A8	D7	000EC	DECL	MAIL\$Q_EDITNAM_DESC	1777	
					68	A8	D6	000EF	INCL	MAIL\$Q_EDITNAM_DESC+4	1778	
				05	51	E9	000F2	6\$:	BLBC	R1, 7\$	1781	
					64	A8	D5	000F5	TSTL	MAIL\$Q_EDITNAM_DESC		
					11	12	000F8	BNEQ	8\$			
				56	94	A9	9A	000FA	7\$:	MOVZBL	CALLABLE_EDT, R6	1783
				50	68	A8	D0	000FE		MOVL	MAIL\$Q_EDITNAM_DESC+4, R0	
			60	95	A9	56	28	00102	MOVZBL	R6, CALLABLE_EDT+1, (R0)		
				64	A8	56	D0	00107	MOVZBL	R6, MAIL\$Q_EDITNAM_DESC	1784	
					64	A8	9F	0010B	8\$:	PUSHAB	MAIL\$Q_EDITNAM_DESC	1787
			00000000G	00	01	FB	0010E	CALLS	#1, MAIL\$UPCASE			
				54	94	A9	9A	00115	MOVZBL	CALLABLE_EDT, R4	1788	
				56	68	A8	D0	00119	MOVL	MAIL\$Q_EDITNAM_DESC+4, R6	1789	
						55	D4	0011D	CLRL	R5	1790	
				95	A9	54	29	0011F	CMPL	R4, (R6), CALLABLE_EDT+1		
						02	12	00124	BNEQ	9\$		
						55	D6	00126	INCL	R5		
01	A8			07	55	F0	00128	9\$:	INSV	R5, #7, #1, MAIL\$GL_FLAGS+1		
					01	A8	95	0012E	TSTB	MAIL\$GL_FLAGS+1	1791	
						4B	18	00131	BGEQ	13\$		
				14	AE	54	C3	00133	SUBL3	R4, MAIL\$Q_EDITNAM_DESC, DESC	1800	
				18	AE	54	C1	00139	ADDL3	R4, R6, DESC+4	1801	
						14	AE	D5	DESC		1802	
						05	12	00141	BNEQ	10\$		
				14	AE	F4	A9	7D	MOVQ	SD_EDT, DESC	1804	
						50	F8	A9	D0	SD_EDT+4, R0	1810	
14	AE			60	F4	A9	2D	0014C	10\$:	CMPL	SD_EDT, (R0), #0, DESC, @DESC+4	
					18	BE		00153				
						04	12	00155	BNEQ	11\$		
				02	A8	01	88	00157	BISB2	#1, MAIL\$GL_FLAGS+2	1811	
					50	C9	D0	0015B	11\$:	MOVL	SD_TPU+4, R0	1812

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

H 3
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2

Page 77
(21)

MA
VO

14	AE	00	60	00E4	C9	2D	00160	CMPC5	SD_TPU, (R0), #0, DESC, @DESC+4		
				18	BE		00163				
		02	A8		04	12	0016A	BNEQ	12\$		
				14	02	88	0016C	BISB2	#2, MAIL\$GL_FLAGS+2	1813	
				0354	AE	9F	00170	PUSHAB	DESC	1814	
		00000000G	00		C8	9F	00173	PUSHAB	MAIL\$Q CALEDTNAM		
		04	AE		02	FB	00177	CALLS	#2, STR\$COPY DX		
1A			68		01	7D	0017E	MOVQ	#1, NOERASE_FLAG	1819	
				04	04	E1	00182	BBC	#4, MAIL\$GL_FLAGS, 14\$	1821	
				04	AE	9F	00186	PUSHAB	NOERASE_FLAG	1824	
				0C	AE	9F	00189	PUSHAB	COLS		
				14	AE	9F	0018C	PUSHAB	ROWS		
				16FC	C9	9F	0018F	PUSHAB	P.AFL		
				04	A8	9F	00193	PUSHAB	MAIL\$L SMG DISPLAY		
		00000000G	00		05	FB	00196	CALLS	#5, SMG\$CREATE_PASTEBOARD		
			36		50	E9	0019D	BLBC	STATUS, 17\$		
				FED0	C9	9F	001A0	PUSHAB	SYSIN DESC	1829	
				08	A8	9F	001A4	PUSHAB	MAIL\$C SMG KEYBOARD		
		00000000G	00		02	FB	001A7	CALLS	#2, SMG\$CREATE_VIRTUAL_KEYBOARD		
			0D		50	E9	001AE	BLBC	STATUS, 15\$		
				0C	A8	9F	001B1	PUSHAB	MAIL\$L SMG KEYTABLE	1831	
		00000000G	00		01	FB	001B4	CALLS	#1, SMG\$CREATE_KEY_TABLE		
			0A		50	E8	001BB	BLBS	STATUS, 16\$		
		00000000G	00		50	DD	001BE	PUSHL	STATUS		
					01	FB	001C0	CALLS	#1, LIB\$SIGNAL		
				036C	C8	9F	001C8	RET			
				70	A8	9F	001CC	PUSHAB	MAIL\$Q_USERDEF	1836	
		00000000G	00		02	FB	001CF	PUSHAB	MAIL\$Q_USERNAME		
			01		50	E8	001D6	CALLS	#2, MAIL\$GET_DEVDIR		
					04	001D9		BLBS	STATUS, 18\$		
			50	00000000G	00	DD	001DA	RET			
		20	A8		AD	DD	001E1	MOVL	MAIL\$T_UAF_BUFF, R0	1840	
				00000000G	00	9F	001E6	MOVL	36(R0), MAIL\$L_MYUIC		
				036C	C8	9F	001EC	PUSHAB	MAIL\$Q_MAIL_MAI	1841	
				0364	C8	9F	001F0	PUSHAB	MAIL\$Q_USERDEF		
		00000000G	00		03	FB	001F4	PUSHAB	MAIL\$Q_MAILDEF		
			50	00000000G	00	DD	001FB	CALLS	#3, STR\$CONCAT		
			01		03	EF	00202	MOVL	MAIL\$T_UAF_BUFF, R0	1842	
			06		51	F0	00209	EXTZV	#3, #1, 468(R0), R1		
01	51	01D4	01		01	E0	0020F	INSV	R1, #6, #1, MAIL\$GL_FLAGS+1		
	AB		6D		01	E0	0020F	BBS	#1, MAIL\$GL_FLAGS+1, 20\$	1846	
				FD70	CF	9E	00214	MOVAB	SWALLOW SIGNALS, (FP)	1848	
				10	AE	9F	00219	PUSHAB	NEWCOUNT	1849	
				70	A8	9F	0021C	PUSHAB	MAIL\$Q_USERNAME		
		00000000G	00		02	FB	0021F	CALLS	#2, MAIL\$GET_NEWMAIL		
			20		50	E9	00226	BLBC	R0, 19\$		
				10	AE	D5	00229	TSTL	NEWCOUNT	1850	
					1B	13	0022C	BEQL	19\$		
				1C	AE	9F	0022E	PUSHAB	NULLDESC	1851	
			6B		01	FB	00231	CALLS	#1, MAIL\$PUT_OUTPUT		
				10	AE	DD	00234	PUSHL	NEWCOUNT	1853	
				007EDE08	8F	DD	00237	PUSHL	#8314376	1852	
		00000000G	00		01	FB	0023D	CALLS	#1, MAIL\$GETMSG		
					50	DD	00244	PUSHL	R0		
			6B		02	FB	00246	CALLS	#2, MAIL\$PUT_OUTPUT		
		00000000G	00		6D	D4	00249	CLRL	(FP)	1854	
					00	FB	0024B	CALLS	#0, MAIL\$DEL_TMPF	1858	

23	68	02	E1	00252	20\$:	BBC	#2, MAIL\$GL_FLAGS, 22\$	1863	
		7E	7C	00256		CLRQ	-(SP)	1865	
		6E	A8	9F	00258	PUSHAB	MAIL\$W_TTCHAN		
		48	A8	9F	00258	PUSHAB	MAIL\$Q-INPTRAN		
00000000G	00	04	FB	0025E		CALLS	#4, SYSS\$ASSIGN		
EEAE	CF	00	FB	00265		CALLS	#0, DEFINE_DEFAULT_KEYPAD	1866	
EEDC	CF	00	FB	0026A		CALLS	#0, LOAD_KEY_DEFS	1867	
06	68	02	E1	0026F	21\$:	BBC	#2, MAIL\$GL_FLAGS, 22\$	1873	
		1C	AE	9F	00273	PUSHAB	NULLDESC	1874	
	6B	01	FB	00276		CALLS	#1, MAIL\$PUT_OUTPUT		
		170C	C9	9F	00279	PUSHAB	P.AFN	1880	
00000000G	00	5A	DD	0027D	22\$:	PUSHL	R10		
	1E	02	FB	0027F		CALLS	#2, MAIL\$GET_INPUT		
0001827A	8F	50	E8	00286		BLBS	STATUS, 24\$		
		50	D1	00289		CMPL	STATUS, #98938		
00000000G	8F	12	13	00290		BEQL	23\$		
		50	D1	00292		CMPL	STATUS, #SMGS_EOF		
		09	13	00299		BEQL	23\$		
		50	DD	0029B		PUSHL	STATUS		
00000000G	00	01	FB	0029D		CALLS	#1, LIB\$SIGNAL		
		00A4	31	002A4	23\$:	BRW	28\$		
		5A	DD	002A7	24\$:	PUSHL	R10	1881	
		5A	DD	002A9		PUSHL	R10		
00000000G	00	02	FB	002AB		CALLS	#2, STR\$TRIM		
		5A	DD	002B2		PUSHL	R10	1882	
00000000G	00	01	FB	002B4		CALLS	#1, MAIL\$UPCASE_Q		
	00000000G	00	9F	002BB		PUSHAB	DECNUM_KEY	1886	
	00000000G	00	9F	002C1		PUSHAB	DECNUM_STATE		
		5A	DD	002C7		PUSHL	R10		
E7BD	CF	03	FB	002C9		CALLS	#3, CALL_TPARSE		
	0F	50	E9	002CE		BLBC	R0, 25\$		
		5A	DD	002D1		PUSHL	R10	1887	
		171C	C9	9F	002D3	PUSHAB	P.AFP	1888	
		5A	DD	002D7		PUSHL	R10	1887	
00000000G	00	03	FB	002D9		CALLS	#3, STR\$CONCAT		
		E8	AA	B4	002E0	25\$:	CLRW	FILE_DESC	1889
		FE50	CA	D4	002E3		CLRL	CLIFLAGS	1890
E709	CF	00	FB	002E7		CALLS	#0, MAIL\$ENABLE_CTRL	1895	
		6A	B5	002EC		TSTW	COMMAND_DESC	1896	
		09	12	002EE		BNEQ	26\$		
00000000G	00	00	FB	002F0		CALLS	#0, MAIL\$CMD_CONTINUE	1897	
		28	11	002F7		BRB	27\$		
		24	AE	D4	002F9	26\$:	CLRL	PARSE_SIGNALLED	1900
	00000000G	00	9F	002FC		PUSHAB	MAIL\$GET_INPUT	1901	
	00000000G	00	9F	00302		PUSHAB	MAIL\$COMMAND_TABLE		
		5A	DD	00308		PUSHL	R10		
00000000G	00	03	FB	0030A		CALLS	#3, CLISDCL_PARSE		
	57	50	D0	00311		MOVL	R0, STATUS		
	0A	57	E9	00314		BIBC	STATUS, 27\$		
00000000G	00	00	FB	00317		CALLS	#0, CLISDISPATCH	1903	
	57	50	D0	0031E		MOVL	R0, STATUS		
00F8	C8	0202	8F	B0	00321	27\$:	MOVW	#514, MAIL\$G_CNCT+128	1909
		00A8	C8	9F	00328		PUSHAB	MAIL\$G_CNCT+48	1911
00000000G	00	01	FB	0032C		CALLS	#1, MAIL\$DELETELINK		
E690	CF	00	FB	00333		CALLS	#0, MAIL\$DISABLE_CTRL	1912	
	02	08	8A	00338		BICB2	#8, MAIL\$GL_FLAGS+2	1913	
09	FE50	CA	03	E0	0033C	BBS	#3, CLIFLAGS, 28\$	1915	

MAILSMAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

J 3
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2

Page 79
(21)

MAIL
V04

03	FE50	CA		04	E0	00342	BBS	#4, CLIFLAGS, 28\$	:
				FF24	31	00348	BRW	21\$	:
07	FE50	CA		04	E0	00348	BBS	#4, CLIFLAGS, 29\$	: 1916
	00000000G	00		00	FB	00351	CALLS	#0, MAIL\$CLOSE_FILE	: 1918
			FF24	CA	9F	00358	PUSHAB	TMP_FAB	: 1919
	00000000G	00		01	FB	0035C	CALLS	#1, -SYSS\$CLOSE	:
		50		01	DO	00363	MOVL	#1, R0	: 1920
					04	00366	RET		: 1921
					0000	00367	.WORD	Save nothing	: 1689
		50	08	AC	DO	00369	MOVL	8(AP), R0	:
		50	04	A0	DO	0036D	MOVL	4(R0), R0	:
			EC	A0	9F	00371	PUSHAB	PARSE_SIGNALLED	:
				01	DD	00374	PUSHL	#1	:
				SE	DD	00376	PUSHL	SP	:
		7E	04	AC	7D	00378	MOVQ	4(AP), -(SP)	:
	FC1B	CF		03	FB	0037C	CALLS	#3, NOTICE_SIGNAL	:
				04		00381	RET		:

; Routine Size: 898 bytes, Routine Base: \$CODE\$ + 187C

```
: 1798 1922 1 GLOBAL ROUTINE MAIL$UP_PRIV (PRIVMASK1,PRIVMASK2) :NOVALUE =
: 1799 1923 1 ++
: 1800 1924 1 FUNCTIONAL DESCRIPTION:
: 1801 1925 1
: 1802 1926 1 Ensure privs specified by privmask are enabled
: 1803 1927 1
: 1804 1928 1 --
: 1805 1929 2 BEGIN
: 1806 1930 3 ENHANCED_PRIVS[0] = (.PRIVMASK1 AND .MAIL$Q PROCPRIV<0,32,0>)
: 1807 1931 3 XOR .PRIVMASK1;
: 1808 1932 3 ENHANCED_PRIVS[1] = (.PRIVMASK2 AND .MAIL$Q PROCPRIV<4,32,0>)
: 1809 1933 3 XOR .PRIVMASK2;
: 1810 1934 4 IF ((.ENHANCED_PRIVS[0] NEQ 0)
: 1811 1935 3 OR (.ENHANCED_PRIVS[1] NEQ 0))
: 1812 1936 2 THEN $SETPRV(ENBFLG=1,PRVADR=ENHANCED_PRIVS);
: 1813 1937 2 RETURN;
: 1814 1938 1 END;
```

				000C 00000	.ENTRY MAIL\$UP_PRIV, Save R2,R3	: 1922
		53	00000000'	00 9E 00002	MOVAB MAIL\$Q PROCPRIV, R3	
		52	00000000'	00 9E 00009	MOVAB ENHANCED_PRIVS, R2	
		50		63 D2 00010	MCOML MAIL\$Q PROCPRIV, R0	: 1930
	50	04	AC	50 CB 00013	BICL3 R0, PRIVMASK1, R0	
	62		50	04 CD 00018	XORL3 PRIVMASK1, R0, ENHANCED_PRIVS	: 1931
50	63		20	04 EF 0001D	EXTZV #4, #32, MAIL\$Q PROCPRIV, R0	: 1932
			50	50 D2 00022	MCOML R0, R0	
	50	08	AC	50 CB 00025	BICL3 R0, PRIVMASK2, R0	
04	A2		50	08 AC CD 0002A	XORL3 PRIVMASK2, R0, ENHANCED_PRIVS+4	: 1933
				62 D5 00030	TSTL ENHANCED_PRIVS	: 1934
				05 12 00032	BNEQ 1\$	
			04	A2 D5 00034	TSTL ENHANCED_PRIVS+4	: 1935
				0D 13 00037	BEQL 2\$	
				7E 7C 00039 1\$:	CLRQ -(SP)	: 1936
				52 DD 0003B	PUSHL R2	
				01 DD 0003D	PUSHL #1	
	00000000G 00			04 FB 0003F	CALLS #4, SYS\$SETPRV	
				04 00046 2\$:	RET	: 1938

: Routine Size: 71 bytes, Routine Base: \$CODE\$ + 1BFE

```
: 1816 1939 1 GLOBAL ROUTINE MAIL$DOWN_PRIV : NOVALUE =
: 1817 1940 1 ++
: 1818 1941 1 FUNCTIONAL DESCRIPTION:
: 1819 1942 1
: 1820 1943 1 Disable privileges enabled by MAIL$UP_PRIV
: 1821 1944 1
: 1822 1945 1 --
: 1823 1946 2 BEGIN
: 1824 1947 2
: 1825 1948 4 IF ((.ENHANCED_PRIVS[0] NEQ 0)
: 1826 1949 3 OR (.ENHANCED_PRIVS[1] NEQ 0))
: 1827 1950 2 THEN $SETPRV(ENBFLG=0,PRVADR=ENHANCED_PRIVS);
: 1828 1951 2 ENHANCED_PRIVS[0] = ENHANCED_PRIVS[0] = 0,
: 1829 1952 2 RETURN;
: 1830 1953 1 END;
```

52 00000000'	00	0004 00000	9E 00002	.ENTRY MAIL\$DWN PRIV, Save R2	1939
	62	D5 00009	D5 00009	MOVAB ENHANCED_PRIVS, R2	1948
	05	12 0000B	12 0000B	TSTL ENHANCED_PRIVS	1949
04	A2	D5 0000D	D5 0000D	BNEQ 1\$	1950
	0D	13 00010	13 00010	TSTL ENHANCED_PRIVS+4	
	7E	7C 00012	7C 00012	BEQL 2\$	
	52	DD 00014	DD 00014	CLRQ -(SP)	
	7E	D4 00016	D4 00016	PUSHL R2	
00000000G 00	04	FB 00018	FB 00018	CLRL -(SP)	
	62	D4 0001F	D4 0001F	CALLS #4, SYS\$SETPRV	1951
	04	00021	00021	CLRL ENHANCED_PRIVS	1953
				RET	

; Routine Size: 34 bytes, Routine Base: \$CODE\$ + 1C45

```
: 1832 1954 1 GLOBAL ROUTINE MAIL$FORW_PRIV_CHECK =
: 1833 1955 2 BEGIN
: 1834 1956 2 ++
: 1835 1957 2 FUNCTIONAL DESCRIPTION:
: 1836 1958 2
: 1837 1959 2 This routine checks that the current process has SYSNAM
: 1838 1960 2 privilege, so that it can do a SET/SHOW FORWARD on another name.
: 1839 1961 2
: 1840 1962 2 --
: 1841 1963 2 IF .MAIL$Q_PROCPRIV[PRV$V_SYSNAM]
: 1842 1964 2 THEN RETURN TRUE
: 1843 1965 2 ELSE BEGIN
: 1844 1966 3 SIGNAL((SS$ NOSYSNAM-ST$K_SEVERE+ST$K_ERROR) OR MAIL$V_FACILITY);
: 1845 1967 3 RETURN ST$R_ERROR
: 1846 1968 2 END;
: 1847 1969 1 END;
```

04 00000000'	00	0000 00000	.ENTRY MAIL\$FORW PRIV CHECK, Save nothing	: 1954
	50	02 E1 00002	BBC #2, MAIL\$Q_PROCPRIV, 1\$	: 1963
		01 D0 0000A	MOVL #1, R0	: 1965
		04 0000D	RET	: 1966
00000000G	00	8F DD 0000E 1\$:	PUSHL #8267794	: 1967
	50	01 FB 00014	CALLS #1, LIB\$SIGNAL	: 1969
		02 D0 0001B	MOVL #2, R0	
		04 0001E	RET	

: Routine Size: 31 bytes, Routine Base: \$CODE\$ + 1C67

```
: 1849      1970 1 GLOBAL ROUTINE MAIL$CMD_DEFINE =  
: 1850      1971 1 |++  
: 1851      1972 1 | FUNCTIONAL DESCRIPTION:  
: 1852      1973 1 |  
: 1853      1974 1 |         Process the DEFINE key command  
: 1854      1975 1 |  
: 1855      1976 2 BEGIN  
: 1856      1977 2  
: 1857 P 1978 2 IF_ERR(SMG$DEFINE_KEY(MAIL$L_SMG_KEYTABLE,COMMAND_DESC);,  
: 1858      1979 2         SIGNAL(MAIL$_PARSEFAIL,1,COMMAND_DESC,.STATUS));  
: 1859      1980 2 RETURN TRUE  
: 1860      1981 1 END;
```

		0004 00000	.ENTRY MAIL\$CMD_DEFINE, Save R2	
	52 00000000'	00 9E 00002	MOYAB COMMAND_DESC, R2	
		52 DD 00009	PUSHL R2	
	00000000'	00 9F 0000B	PUSHAB MAIL\$L_SMG_KEYTABLE	
00000000G	00	02 FB 00011	CALLS #2, SMG\$DEFINE_KEY	
	13	50 E8 00018	BLBS STATUS, 1\$	
		50 DD 0001B	PUSHL STATUS	
		52 DD 0001D	PUSHL R2	
		01 DD 0001F	PUSHL #1	
	007E1248	8F DD 00021	PUSHL #8262216	
00000000G	00	04 FB 00027	CALLS #4, LIB\$SIGNAL	
	50	01 00 0002E 1\$:	MOVL #1, R0	
		04 00031	RET	

: 1970

: 1979

: 1980
: 1981

: Routine Size: 50 bytes, Routine Base: \$CODE\$ + 1C86

```
1862 1 GLOBAL ROUTINE MAIL$GET_CLUNODES =
1863 1
1864 1 Translate MAIL$SYSTEM_FLAGS. If enabled, and in a cluster, then
1865 1 make a concatenated list of cluster nodes to avoid setting up decnet
1866 1 links. nodes will be separated by '/'.
1867 1
1868 2 BEGIN
1869 2
1870 2 LOCAL
1871 2 CSIDADR,
1872 2 SYITMLST : $ITMLST_DECL(ITEMS=2),
1873 2 MEMBER_FLAG : VOLATILE,
1874 2 NDESC : VECTOR[2,LONG];
1875 2
1876 2 IF .MAIL$GL_SYSFLAGS[SYSF_V_CLU_MAIL]
1877 3 THEN BEGIN
1878 3 $ITMLST_INIT(ITMLST=SYITMLST,
1879 3 (ITMCOD=SYIS_CLUSTER_MEMBER,BUFADR=MEMBER_FLAG));
1880 3
1881 3 CSIDADR = 0;
1882 3 $GETSYIW(EFN=1,CSIDADR=CSIDADR,ITMLST=SYITMLST);
1883 3 IF .MEMBER_FLAG
1884 4 THEN BEGIN
1885 4 $ITMLST_INIT(ITMLST=SYITMLST,
1886 4 (ITMCOD=SYIS_NODENAME,
1887 4 BUFADR=.MAIL$GL_BIGBUFFER,BUFSIZ=64,RETLEN=NDESC));
1888 4
1889 4 STR$FREE1_DX(MAIL$Q_CLUNODES);
1890 4 NDESC[0] = 0;
1891 4 NDESC[1] = .MAIL$GL_BIGBUFFER;
1892 4 CSIDADR = -1;
1893 5 WHILE $GETSYIW(EFN=1,CSIDADR=CSIDADR,ITMLST=SYITMLST)
1894 4 DO STR$CONCAT(MAIL$Q_CLUNODES,MAIL$Q_CLUNODES,
1895 4 SLASH_DESC,NDESC);
1896 3 END;
1897 2 END;
1898 2
1899 2 RETURN TRUE
1900 1 END;
```

```
000C 00000
53 00000000G 00 9E 00002
52 00000000' 00 9E 00009
5E 2C C2 00010
79 E8 A2 E9 00013
50 10 AE 9E 00017
80 10CF0004 8F D0 0001B
80 CC AE 9E 00022
80 80 7C 00028
6E D4 00028
7E 7C 0002A
7E D4 0002C
1C AE 9F 0002E
```

.EXTRN SYSS\$GETSYIW

```
.ENTRY MAIL$GET_CLUNODES, Save R2,R3
MOVAB SYSS$GETSYIW, R3
MOVAB MAIL$Q_CLUNODES, R2
SUBL2 #44, SP
BLBC MAIL$GL_SYSFLAGS, 2$
MOVAB SYITMLST, $$ITMBLKPTR
MOVL #282001412, ($$ITMBLKPTR)+
MOVAB MEMBER_FLAG, ($$ITMBLKPTR)+
CLRQ ($$ITMBLKPTR)+
CLRL CSIDADR
CLRQ -(SP)
CLRL -(SP)
PUSHAB SYITMLST
```

```
: 1982
:
:
: 1996
: 1999
:
:
: 2001
: 2002
:
```


		14	7E D4 00031	CLRL	-(SP)	
			AE 9F 00033	PUSHAB	CSIDADR	
			01 DD 00036	PUSHL	#1	
63			07 FB 00038	CALLS	#7, SYSSGETSYIW	
51	0C		AE E9 0003B	BLBC	MEMBER FLAG, 2\$	2003
50	10		AE 9E 0003F	MOVAB	SYITMLST, \$\$ITMBLKPTR	2007
80	10D90040		8F D0 00043	MOVL	#282656832, (\$\$ITMBLKPTR)+	
80	E4		A2 D0 0004A	MOVL	MAIL\$GL BIGBUFFER, (\$\$ITMBLKPTR)+	
80	04		AE 9E 0004E	MOVAB	NDESC, 7\$\$ITMBLKPTR)+	
			80 D4 00052	CLRL	(\$\$ITMBLKPTR)+	
			52 DD 00054	PUSHL	R2	2009
00000000G	00		01 FB 00056	CALLS	#1, STR\$FREE1_DX	
		04	AE D4 0005D	CLRL	NDESC	2010
08	AE	E4	A2 D0 00060	MOVL	MAIL\$GL BIGBUFFER, NDESC+4	2011
	6E		01 CE 00065	MNEGL	#1, CSIDADR	2012
			7E 7C 00068 1\$:	CLRL	-(SP)	2013
			7E D4 0006A	CLRL	-(SP)	
		1C	AE 9F 0006C	PUSHAB	SYITMLST	
			7E D4 0006F	CLRL	-(SP)	
		14	AE 9F 00071	PUSHAB	CSIDADR	
			01 DD 00074	PUSHL	#1	
63			07 FB 00076	CALLS	#7, SYSSGETSYIW	
14			50 E9 00079	BLBC	R0, 2\$	
		04	AE 9F 0007C	PUSHAB	NDESC	2014
		E2D9	CF 9F 0007F	PUSHAB	SLASH_DESC	
			52 DD 00083	PUSHL	R2	
			52 DD 00085	PUSHL	R2	
00000000G	00		04 FB 00087	CALLS	#4, STR\$CONCAT	
			D8 11 0008E	BRB	1\$	
	50		01 D0 00090 2\$:	MOVL	#1, R0	2019
			04 00093	RET		2020

; Routine Size: 148 bytes, Routine Base: \$CODE\$ + 1CB8

```
: 1902      2021 1 GLOBAL ROUTINE MAIL$CHK_CLUNODES(NODE_DESC) =
: 1903      2022 1
: 1904      2023 1 | Check if NODE_DESC is one of the nodes in this cluster,
: 1905      2024 1 | if we are in a cluster.
: 1906      2025 1
: 1907      2026 2 BEGIN
: 1908      2027 2 MAP
: 1909      2028 2     NODE_DESC : REF $BBLOCK;
: 1910      2029 2
: 1911      2030 2 OWN
: 1912      2031 2     TDESC : DYN_DESC_DECL;
: 1913      2032 2
: 1914      2033 2 LOCAL
: 1915      2034 2     STATUS;
: 1916      2035 2
: 1917      2036 2 STATUS = FALSE;
: 1918      2037 2 IF .MAIL$Q_CLUNODES[DSC$W_LENGTH] NEQ 0
: 1919      2038 3 THEN BEGIN
: 1920      2039 3     STR$CONCAT(TDESC,SLASH_DESC,.NODE_DESC);
: 1921      2040 3     STATUS = NOT CH$FAIL(CH$FIND_SUB(.MAIL$Q_CLUNODES[DSC$W_LENGTH],
: 1922      2041 3         .MAIL$Q_CLUNODES[DSC$A_POINTER],
: 1923      2042 3         .TDESC[DSC$W_LENGTH],
: 1924      2043 3         .TDESC[DSC$A_POINTER]));
: 1925      2044 2     END;
: 1926      2045 2
: 1927      2046 2 RETURN .STATUS
: 1928      2047 1 END;
```

.PSECT \$OWNS,NOEXE,2

```
02 00# 0032C TDESC: .BYTE 0[2]
0E 0032E .BYTE 14, 2
00330 .BLKB 4
```

.PSECT \$CODE\$,NOWRT,2

			00FC 00000	.ENTRY MAIL\$CHK_CLUNODES, Save R2,R3,R4,R5,R6,R7	2021
57	00000000	00	9E 00002	MOVAB MAIL\$Q_CLUNODES, R7	
56	00000000	00	9E 00009	MOVAB TDESC, R6	
		54	D4 00010	CLRL STATUS	2036
		67	B5 00012	TSTW MAIL\$Q_CLUNODES	2037
		33	13 00014	BEQL 3\$	
	04	AC	DD 00016	PUSHL NODE_DESC	2039
	E2AB	CF	9F 00019	PUSHAB SLASH_DESC	
		56	DD 0001D	PUSHL R6	
	00000000G	00	03 FB 0001F	CALLS #3, STR\$CONCAT	
		51	04 A7 D0 00026	MOVL MAIL\$Q_CLUNODES+4, R1	2041
		55	66 3C 0002A	MOVZWL TDESC, R5	2042
		50	04 A6 D0 0002D	MOVL TDESC+4, R0	2043
61		60	55 39 00031	MATCHC R5, (R0), MAIL\$Q_CLUNODES, (R1)	
			03 13 00036	BEQL 1\$	
		53	55 D0 00038	MOVL R5, R3	
		53	55 C2 0003B 1\$:	SUBL2 R5, R3	

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

E 4
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.JRC]MAIL.B32;2

Page 87
(27)

	50	04	0003E	CLRL	R0	:
	53	D5	00040	TSTL	R3	:
	02	12	00042	BNEQ	2\$	:
	50	D6	00044	INCL	R0	:
54	50	D2	00046	MCOML	R0, STATUS	: 2040
50	54	D0	00049	MOVL	STATUS, R0	: 2046
	04	0004C	RET			: 2047

; Routine Size: 77 bytes, Routine Base: \$CODE\$ + 1D4C

```
1930 2048 1 GLOBAL ROUTINE MAIL$CMD_FORWARD =
1931 2049 1 **
1932 2050 1 FUNCTIONAL DESCRIPTION:
1933 2051 1
1934 2052 1 This is the FORWARD command action routine and is called to send
1935 2053 1 a copy of the current message to another user or users.
1936 2054 1
1937 2055 1 INPUTS:
1938 2056 1
1939 2057 1 Current message context.
1940 2058 1
1941 2059 1 OUTPUTS:
1942 2060 1
1943 2061 1 A copy of the current message is sent to the specified user(s).
1944 2062 1 --
1945 2063 2 BEGIN
1946 2064 2 LOCAL
1947 2065 2 EDIFAB : $FAB_DECL,
1948 2066 2 EDIRAB : $RAB_DECL,
1949 2067 2 NAMBUF : $BBLOCK[NAM$C_MAXRSS],
1950 2068 2 EDINAM : $NAM_DECL,
1951 2069 2 OUTDESC : VECTOR[2, LONG],
1952 2070 2 MOREFLAG,
1953 2071 2 USERAB : REF $BBLOCK,
1954 2072 2 USEFAB : REF $BBLOCK,
1955 2073 2 USENAM : REF $BBLOCK,
1956 2074 2 STATUS; ! Routine status
1957 2075 2
1958 2076 2 ! Rewind current message
1959 2077 2
1960 P 2078 2 IF_ERR( (IF .MAIL$GL_FLAGS[MAIF_V_ISAM]
1961 P 2079 2 THEN MAIL$REWIND_1_MSG()
1962 P 2080 2 ELSE MAIL$REWIND_MSG());, ! Rewind current message
1963 P 2081 2 SIGNAL (.STATUS);, ! If failure, signal it
1964 2082 2 RETURN TRUE); ! and quit
1965 2083 2
1966 2084 2 $XABPRO_INIT(XAB=TMP_XAB);
1967 2085 2 TMP_XAB[XAB$W_PRO] = TMP_PRO;
1968 2086 2 PARSE_EDIT_QUAL(.CLI_EDITQUAL[F_FORW_EDIT]);
1969 2087 2 IF .C[IFLAGS[EDIT_FLAG]
1970 2088 3 THEN BEGIN
1971 P 2089 3 $FAB_INIT(FAB=EDIFAB,
1972 P 2090 3 FNS=.TMP$FILE1_DESC[DSC$W_LENGTH],
1973 P 2091 3 FNA=.TMP$FILE1_DESC[DSC$A_POINTER],
1974 P 2092 3 RAT=CR,
1975 P 2093 3 RFM=VAR,
1976 P 2094 3 DEQ=3,
1977 P 2095 3 CTX=MAIL$OPENOUT,
1978 P 2096 3 XAB=TMP_XAB,
1979 P 2097 3 FAC=PUT,
1980 P 2098 3 FOP=SQO,
1981 2099 3 NAM=EDINAM);
1982 2100 3
1983 P 2101 3 $NAM_INIT(NAM=EDINAM,
1984 P 2102 3 RSS=NAM$C_MAXRSS,
1985 P 2103 3 RSA=NAMBUF,
1986 P 2104 3 ESS=NAM$C_MAXRSS,
```

```

: 1987      2105      3      ESA=NAMBUF);
: 1988      2106      3
: 1989      P 2107      3      $RAB INIT(RAB=EDIRAB,
: 1990      P 2108      3      FAB=EDIFAB,
: 1991      P 2109      3      CTX=MAIL$ OPENOUT,
: 1992      P 2110      3      UBF=TMP BOFF,
: 1993      2111      3      USZ=MAIL$K_INBUFFSZ);
: 1994      2112      3
: 1995      2113      3      USERAB = EDIRAB;
: 1996      2114      3      USEFAB = EDIFAB;
: 1997      2115      3      USENAM = EDINAM;
: 1998      2116      3      END
: 1999      2117      3      ELSE BEGIN
: 2000      2118      3
: 2001      2119      3      ! Create a temporary file to store the message
: 2002      2120      3
: 2003      2121      3      $CLOSE(FAB=TMP FAB);
: 2004      P 2122      3      $FAB INIT(FAB=TMP_FAB,
: 2005      P 2123      3      DEQ=3,
: 2006      P 2124      3      NAM=TMP NAM,
: 2007      P 2125      3      FNS = .TMPFILE_DESC[DSC$W_LENGTH],
: 2008      P 2126      3      FNA = .TMPFILE_DESC[DSC$A_POINTER],
: 2009      P 2127      3      CTX = MAIL$ OPENOUT,
: 2010      P 2128      3      XAB = TMP_XAB,
: 2011      P 2129      3      RAT = CR,
: 2012      P 2130      3      RFM = VAR,
: 2013      P 2131      3      FOP = (TMD,SQO),
: 2014      2132      3      FAC = (GET,PUT,BRO));
: 2015      2133      3
: 2016      P 2134      3      $NAM INIT(NAM=TMP NAM,
: 2017      P 2135      3      RSS=NAM$C_MAXRSS,
: 2018      P 2136      3      RSA=TMP_ESNAM_BUF,
: 2019      P 2137      3      ESA=TMP_ESNAM_BUF,
: 2020      2138      3      ESS=NAM$C_MAXRSS);
: 2021      2139      3
: 2022      P 2140      3      $RAB INIT(RAB=TMP_RAB,
: 2023      P 2141      3      FAB=TMP FAB,
: 2024      P 2142      3      CTX=MAIL$ OPENOUT,
: 2025      P 2143      3      UBF=TMP BOFF,
: 2026      2144      3      USZ=MAIL$K_INBUFFSZ);
: 2027      2145      3
: 2028      2146      3      USEFAB = TMP_FAB;
: 2029      2147      3      USERAB = TMP_RAB;
: 2030      2148      3      USENAM = TMP_NAM;
: 2031      2149      2      END;
: 2032      2150      2
: 2033      P 2151      2      IF_ERR( $CREATE(FAB = .USEFAB,ERR=UTIL$REPORT_IO_ERROR);,! Create a temp file
: 2034      2152      2      RETURN TRUE);
: 2035      P 2153      2      IF_ERR($CONNECT(RAB = .USERAB,ERR=UTIL$REPORT_IO_ERROR);,! Connect
: 2036      P 2154      2      EDIFAB[FAB$V_DLT] = 1;
: 2037      P 2155      2      $CLOSE(FAB=.USEFAB);
: 2038      2156      2      RETURN TRUE);
: 2039      2157      2      USERAB[RAB$L_CTX] = MAIL$ WRITEERR;
: 2040      2158      2      STATUS = MAIL$EXTRACT_MESSAGE(CLIP$PRESENT(SD_HEADER),.USERAB);
: 2041      2159      2      IF NOT .STATUS
: 2042      2160      3      THEN BEGIN
: 2043      2161      3      USEFAB[FAB$V_DLT] = 1;

```

```
: 2044      2162      3      $CLOSE(FAB = .USEFAB);
: 2045      2163      3      RETURN TRUE;
: 2046      2164      3      END;
: 2047      2165      2      :
: 2048      2166      2      : Get the name of the new addressee(s) and send the message.
: 2049      2167      2      :
: 2050      2168      2      MAIL$GET_CLUNODES();          ! Get list of cluster nodes
: 2051      2169      2      MAIL$GET_INPUT(MAIL$Q_TO,MAIL$GETMSG(MAIL$_ISUB20));
: 2052      2170      2      IF .MAIL$Q_TO[DESC$W_LENGTH] EQL 0      ! If nothing entered, quit
: 2053      2171      2      THEN (USEFAB[FAB$V_DLT] = 1; $CLOSE(FAB = .USEFAB); RETURN TRUE);
: 2054      2172      2      MAIL$UPCASE_Q(MAIL$Q_TO);          ! Upcase the addressee names
: 2055      2173      2      MAIL$INI_ADDR(MAIL$G_CNCT);          ! Init addressee list
: 2056      2174      2      IF CHECK_SEND(MAIL$ADD_ADDR(MAIL$Q_TO,MAIL$G_CNCT))! Add addressee(s) to list
: 2057      2175      3      THEN BEGIN
: 2058      2176      3          MAIL$NET_END_USERS(MAIL$G_CNCT);
: 2059      2177      3          IF MAIL$GET_INPUT(MAIL$Q_SUBJ,
: 2060      2178      3              MAIL$GETMSG(MAIL$_ISUB30)) EQL RMSS_EOF! Get the subject
: 2061      2179      3              THEN (USEFAB[FAB$V_DLT] = 1; $CLOSE(FAB=.USEFAB); RETURN TRUE);
: 2062      2180      3          IF .CLIFLAGS[EDIT_FLAG]
: 2063      2181      4          THEN BEGIN
: 2064      2182      4              $CLOSE(FAB=.USEFAB);
: 2065      2183      4              GET_SEND_MSG(TMPFILE1_DESC);
: 2066      2184      4              USEFAB[FAB$V_DLT] = 1;
: 2067      2185      4              $OPEN(FAB=.USEFAB);
: 2068      2186      4              END
: 2069      2187      3          ELSE MAIL$SEND_ADDR(MAIL$G_CNCT);
: 2070      2188      2          END;
: 2071      2189      2      USEFAB[FAB$V_DLT] = 1;
: 2072      2190      2      $CLOSE(FAB = .USEFAB);          ! Close the temp file
: 2073      2191      2      RETURN TRUE
: 2074      2192      1      END;
```

```
$RMS_PTR=      TMP_XAB
$RMS_PTR=      TMP_FAB
$RMS_PTR=      TMP_NAM
$RMS_PTR=      MAIL$G_CNCT+646
```

	OFFC	00000		.ENTRY	MAIL\$CMD FORWARD, Save R2,R3,R4,R5,R6,R7,-	
					R8,R9,R10,R11	2048
	5B	00000000G	00 9E 00002	MOVAB	MAIL\$GETMSG, R11	
	5A	00000000G	00 9E 00009	MOVAB	UTIL\$REPORT_IO_ERROR, R10	
	59	00000000G	00 9E 00010	MOVAB	SYSSCLOSE, R9	
	58	E581	CF 9E 00017	MOVAB	PARSE EDIT_QUAL, R8	
	57	00000000'	00 9E 0001C	MOVAB	MAIL\$Q_TO, R7	
	56	00000000'	00 9E 00023	MOVAB	TMP FAB, R6	
	5E	FE04	CE 9E 0002A	MOVAB	-508(SP), SP	
	09	FF78	C7 E9 0002F	BLBC	MAIL\$GL_FLAGS, 1\$	2082
00000000G	00		00 FB 00034	CALLS	#0, MAIL\$REWIND_I_MSG	
			07 11 0003B	BRB	2\$	
00000000G	00		00 FB 0003D	CALLS	#0, MAIL\$REWIND_MSG	
	0C		50 E8 00044	BLBS	STATUS, 3\$	
			50 DD 00047	PUSHL	STATUS	
00000000G	00		01 FB 00049	CALLS	#1, LIB\$SIGNAL	
			0251 31 00050	BRW	14\$	

0058	8F	00	6E	00	2C	00053	3\$:	MOVCS	#0, (SP), #0, #88, \$RMS_PTR	2084
			50	A6	8F	80	0005A			
			58	A6	8F	80	0005C	MOVW	#22547, \$RMS_PTR	
	7E	FF34	C6	01	02	EF	00068	MOVW	#-188, TMP_XAB+8	2085
				68	01	FB	0006F	EXTZV	#2, #1, CLT_EDITQUAL, -(SP)	2086
				03	C6	EB	00072	CALLS	#1, PARSE_EDIT_QUAL	
					009A	31	00077	BLBS	CLIFLAGS, -4\$	2087
0050	8F	00	6E	00	2C	0007A	4\$:	BRW	5\$	
								MOVCS	#0, (SP), #0, #80, \$RMS_PTR	2099
				B0	AD	8F	80	00081		
				B4	AD	8F	9A	00083	MOVW	#20483, \$RMS_PTR
				C4	AD	8F	9A	00089	MOVZBL	#64, \$RMS_PTR+4
				C6	AD	03	80	0008E	MOVW	#3, \$RMS_PTR+20
				C8	AD	01	90	00092	MOVW	#3, \$RMS_PTR+20
				CE	AD	8F	D0	00096	MOVW	#1, \$RMS_PTR+22
				D4	AD	8F	D0	0009E	MOVL	#8261794, \$RMS_PTR+24
				D8	AD	A6	9E	000A4	MOVW	#514, \$RMS_PTR+30
				DC	AD	AF	9E	000A9	MOVAB	TMP_XAB, \$RMS_PTR+36
				E4	AD	C6	D0	000AE	MOVAB	EDINAM, \$RMS_PTR+40
0060	8F	00	6E	00	2C	000BA		MOVL	TMPIFILE1_DESC+4, \$RMS_PTR+44	
								MOVW	TMPIFILE1_DESC, \$RMS_PTR+52	
								MOVCS	#0, (SP), #0, #96, \$RMS_PTR	2105
				08	AE	000C1				
				0A	AE	8F	80	000C3	MOVW	#24578, \$RMS_PTR
				0C	AE	01	8E	000C9	MNEGB	#1, \$RMS_PTR+2
				12	AE	AE	9E	000CD	MOVAB	NAMBUF, \$RMS_PTR+4
				14	AE	01	8E	000D2	MNEGB	#1, \$RMS_PTR+10
0044	8F	00	6E	00	2C	000DB		MOVAB	NAMBUF, \$RMS_PTR+12	
								MOVCS	#0, (SP), #0, #68, \$RMS_PTR	2111
				FF6C	CD	000E2				
				84	AD	8F	80	000E5	MOVW	#17409, \$RMS_PTR
				8C	AD	8F	D0	000EC	MOVL	#8261794, \$RMS_PTR+24
				90	AD	8F	80	000F4	MOVW	#512, \$RMS_PTR+32
				A8	AD	A7	9E	000FA	MOVAB	TMP_BUFF, \$RMS_PTR+36
					AD	AD	9E	000FF	MOVAB	EDIFAB, \$RMS_PTR+60
				53	FF6C	CD	9E	00104	MOVAB	EDIRAB, USERAB
				52	B0	AD	9E	00109	MOVAB	EDIFAB, USEFAB
				50	08	AE	9E	0010D	MOVAB	EDINAM, USENAM
					009F	31	00111	BRW	6\$	2113
					56	DD	00114	PUSHL	R6	2114
					01	FB	00116	CALLS	#1, SYS\$CLOSE	2115
0050	8F	00	6E	00	2C	00119		MOVCS	#0, (SP), #0, #80, \$RMS_PTR	2087
										2121
				66	8F	80	00120			
				04	A6	8F	9A	00121	MOVW	#20483, \$RMS_PTR
				14	A6	8F	9A	00126	MOVZBL	#80, \$RMS_PTR+4
				16	A6	03	80	00128	MOVW	#3, \$RMS_PTR+20
				18	A6	8F	90	0012F	MOVW	#3, \$RMS_PTR+20
				1E	A6	8F	D0	00134	MOVW	#67, \$RMS_PTR+22
				24	A6	8F	D0	0013C	MOVL	#8261794, \$RMS_PTR+24
				28	A6	8F	D0	0013C	MOVW	#514, \$RMS_PTR+30
				2C	A6	A6	9E	00142	MOVAB	TMP_XAB, \$RMS_PTR+36
				34	A6	A6	9E	00147	MOVAB	TMP_NAM, \$RMS_PTR+40
0060	8F	00	6E	00	2C	00158		MOVL	TMPIFILE1_DESC+4, \$RMS_PTR+44	
								MOVW	TMPIFILE1_DESC, \$RMS_PTR+52	
								MOVCS	#0, (SP), #0, #96, \$RMS_PTR	2138
				A0	A6	8F	80	0015F		
				A2	A6	01	8E	00161	MOVW	#24578, \$RMS_PTR
				A4	A6	01	8E	00167	MNEGB	#1, \$RMS_PTR+2
					FE2C	C6	9E	0016B	MOVAB	TMP_ESNAM_BUF, \$RMS_PTR+4

0044	8F	00	AA AC	A6 A6 6E	FE2C	01 C6 00	8E 9E 2C	00171 00175 00178	MNEGB MOVAB MOVCS	#1, \$RMS_PTR+10 TMP_ESNAM_BUF, \$RMS_PTR+12 #0, (SP), #0, #68, \$RMS_PTR	2144
					0276	C7	8F	00182			
			0276	C7	4401	8F	80	00185	MOVW	#17409, \$RMS_PTR	
			028E	C7	007E10A2	8F	80	0018C	MOVL	#8261794, \$RMS_PTR+24	
			0296	C7	0200	8F	80	00195	MOVW	#512, \$RMS_PTR+32	
			029A	C7	76	A7	9E	0019C	MOVAB	TMP_BUFF, \$RMS_PTR+36	
			02B2	C7		66	9E	001A2	MOVAB	TMP_FAB, \$RMS_PTR+60	
				52		66	9E	001A7	MOVAB	TMP_FAB, USEFAB	2146
				53	0276	C7	9E	001AA	MOVAB	TMP_RAB, USERAB	2147
				50	A0	A6	9E	001AF	MOVAB	TMP_NAM, USENAM	2148
					04C4	8F	BB	001B3	6\$: PUSHR	#^M<R2,R10>	2152
		00000000G		00		02	FB	001B7	CA_LS	#2, SYSS\$CREATE	
				03		50	E8	001BE	BLBS	STATUS, 7\$	
						00E0	31	001C1	BRW	14\$	
					0408	8F	BB	001C4	7\$: PUSHR	#^M<R3,R10>	2156
		00000000G		00		02	FB	001C8	CALLS	#2, SYSS\$CONNECT	
				08		50	E8	001CF	BLBS	STATUS, 8\$	
			B5	AD	80	8F	88	001D2	BISB2	#128, EDIFAB+5	
						00C5	31	001D7	BRW	13\$	
			18	A3	007E10D0	8F	80	001DA	8\$: MOVL	#8261840, 24(USERAB)	2157
						53	DD	001E2	PUSHL	USERAB	2158
					E193	CF	9F	001E4	PUSHAB	SD_HEADER	
		00000000G		00		01	FB	001E8	CALLS	#1, CLIS\$PRESENT	
						50	DD	001EF	PUSHL	R0	
		00000000G		00		02	FB	001F1	CALLS	#2, MAIL\$EXTRACT_MESSAGE	
				45		50	E9	001F8	BLBC	STATUS, 9\$	2159
			1983	C8		00	FB	001FB	CALLS	#0, MAIL\$GET_CLUNODES	2168
					007EDDF0	8F	DD	00200	PUSHL	#8314352	2169
				6B		01	FB	00206	CALLS	#1, MAIL\$GETMSG	
						50	DD	00209	PUSHL	R0	
						57	DD	0020B	PUSHL	R7	
		00000000G		00		02	FB	0020D	CALLS	#2, MAIL\$GET_INPUT	
						67	B5	00214	TSTW	MAIL\$Q_TO	2170
						76	13	00216	BEQL	10\$	
						57	DD	00218	PUSHL	R7	2172
		00000000G		00		01	FB	0021A	CALLS	#1, MAIL\$UPCASE_Q	
					F0	A7	9F	00221	PUSHAB	MAIL\$G_CNCT	2173
		00000000G		00		01	FB	00224	CALLS	#1, MAIL\$INI_ADDR	
					F0	A7	9F	0022B	PUSHAB	MAIL\$G_CNCT	2174
						57	DD	0022E	PUSHL	R7	
		00000000G		00		02	FB	00230	CALLS	#2, MAIL\$ADD_ADDR	
						50	DD	00237	PUSHL	R0	
		00000000V		00		01	FB	00239	CALLS	#1, CHECK_SEND	
				57		50	E9	00240	9\$: BLBC	R0, 12\$	
					F0	A7	9F	00243	PUSHAB	MAIL\$G_CNCT	2176
		00000000G		00		01	FB	00246	CALLS	#1, MAIL\$NET_END_USERS	
					007EDDF8	8F	DD	0024D	PUSHL	#8314360	2178
				6B		01	FB	00253	CALLS	#1, MAIL\$GETMSG	
						50	DD	00256	PUSHL	R0	
					08	A7	9F	00258	PUSHAB	MAIL\$Q_SUBJ	2177
		00000000G		00		02	FB	0025B	CALLS	#2, MAIL\$GET_INPUT	
				8F		50	D1	00262	CMPL	R0, #98938	2178
		0001827A				2F	13	00269	BEQL	12\$	
				20	FF2C	C6	E9	0026B	BLBC	CLIFLAGS, 11\$	2180
						52	DD	00270	PUSHL	USEFAB	2182

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

K 4
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2

Page 93
(28)

69		01	FB	00272	CALLS	#1, SYSSCLOSE	:	
	00D4	C6	9F	00275	PUSHAB	TMPFILE1 DESC	:	2183
00000000V	00	01	FB	00279	CALLS	#1, GET SEND MSG	:	
05	A2	80	8F	88 00280	BISB2	#128, 5(USEFAB)	:	2184
		52	DD	00285	PUSHL	USEFAB	:	2185
00000000G	00	01	FB	00287	CALLS	#1, SYSSOPEN	:	
		0A	11	0028E 10\$:	BRB	12\$	:	2180
		F0	A7	9F 00290 11\$:	PUSHAB	MAIL\$G CNCT	:	2187
00000000G	00	01	FB	00293	CALLS	#1, MAIL\$SEND ADDR	:	
05	A2	80	8F	88 0029A 12\$:	BISB2	#128, 5(USEFAB)	:	2189
		52	DD	0029F 13\$:	PUSHL	USEFAB	:	2190
69		01	FB	002A1	CALLS	#1, SYSSCLOSE	:	
50		01	D0	002A4 14\$:	MOVL	#1, R0	:	2191
		04	002A7		RET		:	2192

; Routine Size: 680 bytes, Routine Base: \$CODE\$ + 1099

```
: 2076      2193 1 GLOBAL ROUTINE MAIL$LOOKUP_FILE(FILE_DESC) =
: 2077      2194 1  ++
: 2078      2195 1  FUNCTIONAL DESCRIPTION:
: 2079      2196 1
: 2080      2197 1      Opens the specified file to ensure it is present and readable.  OPENIN
: 2081      2198 1      error signalled if not
: 2082      2199 1
: 2083      2200 1  INPUTS:
: 2084      2201 1
: 2085      2202 1      FILE_DESC is the address of a dynamic string of the file to open.  It
: 2086      2203 1      is updated to have the resultant name if the open is successful
: 2087      2204 1
: 2088      2205 1  --
: 2089      2206 2 BEGIN
: 2090      2207 2
: 2091      2208 2 MAP
: 2092      2209 2     FILE_DESC : REF $BLOCK;
: 2093      2210 2
: 2094      2211 2 LOCAL
: 2095      2212 2     IFAB : $FAB_DECL,
: 2096      2213 2     INAM : $NAM_DECL,
: 2097      2214 2     NAMRSS : $BLOCK[NAM$C_MAXRSS],
: 2098      2215 2     STATUS;
: 2099      2216 2
: 2100      2217 2 IF .FILE_DESC[DSC$W_LENGTH] EQL 0
: 2101      2218 2 THEN RETURN TRUE;
: 2102      2219 2
: 2103      P 2220 2 $FAB_INIT(FAB=IFAB,
: 2104      P 2221 2     NAM=INAM,
: 2105      P 2222 2     FNS=.FILE_DESC[DSC$W_LENGTH],
: 2106      P 2223 2     FNA=.FILE_DESC[DSC$A_POINTER],
: 2107      P 2224 2     DNM='.TXT',
: 2108      2225 2     CTX=MAIL$_OPENIN);
: 2109      2226 2
: 2110      P 2227 2 $NAM_INIT(NAM=INAM,
: 2111      P 2228 2     RSS=NAM$C_MAXRSS,
: 2112      P 2229 2     RSA=NAMRSS,
: 2113      P 2230 2     ESS=NAM$C_MAXRSS,
: 2114      2231 2     ESA=NAMRSS);
: 2115      2232 2
: 2116      2233 2 STATUS = $OPEN(FAB=IFAB,ERR=UTIL$REPORT_IO_ERROR);
: 2117      2234 2 IF .STATUS
: 2118      2235 3 THEN BEGIN
: 2119      2236 3     MAIL$G_CNCT[CNCT_B_FILRAT] = .IFAB[FAB$B_RAT];
: 2120      2237 3     MAIL$G_CNCT[CNCT_B_FILRFM] = .IFAB[FAB$B_RFM];
: 2121      2238 2     END;
: 2122      2239 2 $CLOSE(FAB=IFAB);
: 2123      2240 2 STR$COPY_DX(.FILE_DESC,UTIL$GETFILENAME(IFAB));
: 2124      2241 2 RETURN .STATUS
: 2125      2242 1 END;
```


MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

M 4
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2

Page 95
(29)

0050	8F	00	5E	FE50	CE	007C	00000	.ENTRY	MAIL\$LOOKUP FILE, Save R2,R3,R4,R5,R6	: 2193
			56	04	AC	9E	00002	MOVAB	-432(SP), SP	: 2217
					66	B5	0000B	MOVL	FILE_DESC, R6	: 2218
					04	12	0000D	TSTW	(R6)	: 2225
			50		01	D0	0000F	BNEQ	1\$	: 2231
						04	00012	MOVL	#1, R0	: 2242
			6E		00	2C	00013	RET		: 2243
				B0	AD		0001A	MOVCS	#0, (SP), #0, #80, \$RMS_PTR	: 2244
				5003	8F	B0	0001C	MOVW	#20483, \$RMS_PTR	: 2245
					02	90	00022	MOVAB	#2, \$RMS_PTR+22	: 2246
				007E109A	8F	D0	00026	MOVL	#8261786, \$RMS_PTR+24	: 2247
					02	90	0002E	MOVAB	#2, \$RMS_PTR+3T	: 2248
					CD	9E	00032	MOVAB	INAM, \$RMS_PTR+40	: 2249
					A6	D0	00038	MOVL	4(R6), \$RMS_PTR+44	: 2250
					AF	9E	0003D	MOVAB	P.AFR, \$RMS_PTR+48	: 2251
					66	90	00042	MOVAB	(R6), \$RMS_PTR+52	: 2252
					04	90	00046	MOVAB	#4, \$RMS_PTR+53	: 2253
0060	8F	00	6E		00	2C	0004A	MOVCS	#0, (SP), #0, #96, \$RMS_PTR	: 2254
				FF50	CD		00051	MOVW	#24578, \$RMS_PTR	: 2255
				6002	8F	B0	00054	MNEGB	#1, \$RMS_PTR+2	: 2256
					01	8E	0005B	MOVAB	NAMRSS, \$RMS_PTR+4	: 2257
					6E	9E	00060	MNEGB	#1, \$RMS_PTR+10	: 2258
					01	8E	00065	MOVAB	NAMRSS, \$RMS_PTR+12	: 2259
					6E	9E	0006A	PUSHAB	UTIL\$REPORT_IO_ERROR	: 2260
					00	9F	0006F	PUSHAB	IFAB	: 2261
					AD	9F	00075	CALLS	#2, SYSS\$OPEN	: 2262
					02	FB	00078	MOVL	R0, STATUS	: 2263
					50	D0	0007F	BLBC	STATUS, 2\$	: 2264
					52	E9	00082	MOVW	IFAB+30, MAIL\$G_CNCT+128	: 2265
					AD	B0	00085	PUSHAB	IFAB	: 2266
					01	FB	00090	CALLS	#1, SYSS\$CLOSE	: 2267
					AD	9F	00097	PUSHAB	IFAB	: 2268
					01	FB	0009A	CALLS	#1, UTIL\$GETFILENAME	: 2269
					50	DD	000A1	PUSHL	R0	: 2270
					56	DD	000A3	PUSHL	R6	: 2271
					02	FB	000A5	CALLS	#2, STR\$COPY_DX	: 2272
					52	D0	000AC	MOVL	STATUS, R0	: 2273
					04	000AF		RET		: 2274

; Routine Size: 176 bytes, Routine Base: \$CODE\$ + 2045

```
2127 2243 1 ROUTINE SEND_SELF_IF (PERM_FLAG) =
2128 2244 1 ++
2129 2245 1 FUNCTIONAL DESCRIPTION:
2130 2246 1
2131 2247 1 This routine adds the sender's name to the end of the 'to' list
2132 2248 1 if (.perm_flag and not /noself) or /self.
2133 2249 1
2134 2250 1 --
2135 2251 2 BEGIN
2136 2252 2 LOCAL
2137 2253 2 STATUS;
2138 2254 2
2139 2255 2 STATUS = CLIS$PRESENT(SD_SELF);
2140 2256 2 IF .STATUS EQL CLIS$NEGATED
2141 2257 2 OR .STATUS EQL CLIS$LOCNEG
2142 2258 2 THEN RETURN .STATUS;
2143 2259 2
2144 2260 2 IF .STATUS
2145 2261 2 OR .PERM_FLAG
2146 2262 3 THEN BEGIN
2147 2263 3 IF .MAIL$Q_TO[DSC$W_LENGTH] NEQ 0
2148 2264 3 THEN STR$APPEND(MAIL$Q_TO, COMMA_DESC);
2149 2265 3 STR$APPEND(MAIL$Q_TO, MAIL$Q_USERNAME);
2150 2266 2 END;
2151 2267 3 RETURN (.STATUS OR .PERM_FLAG)
2152 2268 1 END;
```

001C 00000 SEND_SELF_IF:				
			WORD Save R2,R3,R4	2243
54	00000000G	00	9E 00002	MOVAB STR\$APPEND, R4
53	00000000'	00	9E 00009	MOVAB MAIL\$Q_TO, R3
	EOD3	CF	9F 00010	PUSHAB SD_SELF
00000000G	00	01	FB 00014	CALLS #1, CLIS\$PRESENT
	52	50	D0 0001B	MOVL R0, STATUS
00000000G	8F	52	D1 0001E	CMPL STATUS, #CLIS\$NEGATED
		09	13 00025	BEQL 1\$
00000000G	8F	52	D1 00027	CMPL STATUS, #CLIS\$LOCNEG
		04	12 0002E	BNEQ 2\$
	50	52	D0 00030	MOVL STATUS, R0
		04	00033	RET
	04	52	E8 00034	BLBS STATUS, 3\$
	15	AC	E9 00037	BLBC PERM_FLAG, 5\$
		63	B5 0003B	TSTW MAIL\$Q_TO
		09	13 0003D	BEQL 4\$
		CF	9F 0003F	PUSHAB COMMA_DESC
	DED0	53	DD 00043	PUSHL R3
	64	02	FB 00045	CALLS #2, STR\$APPEND
		A3	9F 00048	PUSHAB MAIL\$Q_USERNAME
		53	DD 0004B	PUSHL R3
	64	02	FB 0004D	CALLS #2, STR\$APPEND
50	52	04	AC C9 00050	BISL3 PERM_FLAG, STATUS, R0
		04	00055	RET

MAILMAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

B 5
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2 Page 97
(30)

; Routine Size: 86 bytes, Routine Base: \$CODE\$ + 20F5

MA
VO

```
2154 2269 1 GLOBAL ROUTINE MAIL$CMD_REPLY =
2155 2270 1 ++
2156 2271 1 FUNCTIONAL DESCRIPTION:
2157 2272 1
2158 2273 1 This routine is the REPLY command action routine and is called
2159 2274 1 to send a message to the sender of the current message.
2160 2275 1
2161 2276 1 INPUTS:
2162 2277 1
2163 2278 1 Current message context.
2164 2279 1
2165 2280 1 OUTPUTS:
2166 2281 1
2167 2282 1 Message sent to sender of current message.
2168 2283 1 --
2169 2284 2 BEGIN
2170 2285 2
2171 2286 2 LOCAL
2172 2287 2 EXTRACT_FLAG,
2173 2288 2 EXTRAB : $FAB_DECL,
2174 2289 2 EXTRAB : $RAB_DECL,
2175 2290 2 EXTNAME : $NAM_DECL,
2176 2291 2 NAMBUF : $BBLOCK[NAM$C_MAXRSS],
2177 2292 2 OUTDESC : VECTOR[2, LONG],
2178 2293 2 MORE_FLAG,
2179 2294 2 PUT_STATUS,
2180 2295 2 SEND_STATUS : VOLATILE;
2181 2296 2
2182 2297 2 ROUTINE REPLY_HANDLER (SIGARG, MECHARG, ENBL) =
2183 2298 3 BEGIN
2184 2299 3 MAP
2185 2300 3 SIGARG : REF $BBLOCK,
2186 2301 3 ENBL : REF VECTOR[, LONG];
2187 2302 3
2188 2303 3 BIND
2189 2304 3 RETSTS = ENBL[1] : REF VECTOR[, LONG];
2190 2305 3
2191 2306 3 RETSTS[0] = .SIGARG[CH$$_SIG_NAME];
2192 2307 3 RETURN $$$_RESIGNAL
2193 2308 2 END;
```

```
0000 00000 REPLY_HANDLER:
51      0C      AC      04      C1 00002      .WORD      Save nothing      : 2297
          50      04      AC      D0 00007      ADDL3      #4, ENBL, R1      : 2304
          00      B1      04      A0 D0 0000B      MOVL      SIGARG, R0      : 2306
          50      0918 8F      3C 00010      MOVL      4(R0), a0(R1)      :
          04      00015      RET      #2328, R0      : 2307
          : 2308
```

; Routine Size: 22 bytes, Routine Base: \$CODE\$ + 214B

; 2194 2309 2

```
: 2195      2310 2 ENABLE REPLY_HANDLER (SEND_STATUS);
: 2196      2311 2 SEND_STATUS = 1;
: 2197      2312 2 PUT STATUS = PARSE_EDIT_QUAL(.CLI_EDITQUAL[F_REPLY_EDIT]);
: 2198      2313 2 CLIFLAGS[LAST_FLAG] = (CLISPRESENT(SD_LAST));
: 2199      2314 2 EXTRACT_FLAG = CLISPRESENT(SD_EXTRACT);
: 2200      2315 2 IF .CLI_EXTRACT
: 2201      2316 3   AND ((.EXTRACT_FLAG NEQ CLIS_NEGATED) AND (.EXTRACT_FLAG NEQ CLIS_LOCNEG))
: 2202      2317 3   THEN EXTRACT_FLAG = TRUE;
: 2203      2318 2
: 2204      2319 2   Disable /extract if explicit /noedit
: 2205      2320 2
: 2206      2321 2 IF .PUT_STATUS EQL CLIS_LOCNEG
: 2207      2322 2   OR .PUT_STATUS EQL CLIS_NEGATED
: 2208      2323 2   THEN EXTRACT_FLAG = FALSE;
: 2209      2324 2 IF CLISGET_VALUE(SD_REPLF,COMMAND_DESC)
: 2210      2325 3 THEN BEGIN
: 2211      2326 3   EXTRACT_FLAG = FALSE;
: 2212      P 2327 3   IF_ERR(MAIL$LOOKUP_FILE(COMMAND_DESC);,
: 2213      2328 3     RETURN TRUE);
: 2214      2329 3
: 2215      2330 2   END;
: 2216      2331 2   /EXTRACT implies /edit
: 2217      2332 2
: 2218      2333 2 IF .EXTRACT_FLAG
: 2219      2334 2   THEN CLIFLAGS[EDIT_FLAG] = 1;                      !Force editing
: 2220      2335 2
: 2221      2336 2   Get the sender and subject of the current message. Use the sender's name
: 2222      2337 2   as the addressee of the new message.
: 2223      2338 2
: 2224      2339 2 MAIL$GET_CLUNODES();                                !Get list of cluster nodes
: 2225      P 2340 2 IF_ERR( TIF .MAIL$GL_FLAGS[MAIF_V_ISAM]
: 2226      P 2341 2   THEN MAIL$INFO_1 MSG(0,MAIL$Q_SUBJ,0,0,0,MAIL$Q_TO)
: 2227      P 2342 2   ELSE MAIL$INFO_1 MSG(MAIL$Q_TO,
: 2228      P 2343 2     MAIL$Q_SUBJ));
: 2229      2344 2   SIGNAL(.STATUS); RETURN TRUE;
: 2230      2345 2 SEND SELF IF(.MAIL$USRSYSBUF[VMD_V_SELFREPLY]);
: 2231      2346 2 MAIL$INI_ADDR(MAIL$G_CNCT);
: 2232      P 2347 2 IF_ERR( MAIL$ADD_ADDR(MAIL$Q_TO,MAIL$G_CNCT);,
: 2233      P 2348 2   IF NOT .MAIL$GL_FLAGS[MAIF_V_ISAM]
: 2234      P 2349 2   THEN MAIL$REWIND_MSG();
: 2235      2350 2   RETURN TRUE;
: 2236      2351 2 MAIL$PUT_OUTPUT(MAIL$GETMSG(MAIL$_MAIL10),MAIL$Q_TO);
: 2237      2352 2 MAIL$NET-END USERS(MAIL$G_CNCT);
: 2238      2353 2 IF .MAIL$GL_FLAGS[MAIF_V_CTRLCLF]
: 2239      2354 2   THEN RETURN SIGNAL(MAIL$_SENDABORT);
: 2240      2355 2
: 2241      2356 2   If a subject was specified in the original message, add a 'RE:' and use it
: 2242      2357 2   as the subject of the new message. Otherwise, just prompt for a subject.
: 2243      2358 2
: 2244      2359 2 IF .MAIL$Q_SUBJ[DSC$W_LENGTH] NEQ 0
: 2245      2360 3 THEN BEGIN
: 2246      2361 3   IF (.MAIL$Q_SUBJ[DSC$A_POINTER]) NEQ 'RE: '    ! If not already 'RE:'
: 2247      2362 3   THEN STR$CONCAT(MAIL$Q_SUBJ,
: 2248      2363 3     DESCRIPTOR('RE: '),
: 2249      2364 3     MAIL$Q_SUBJ);
: 2250      2365 3 MAIL$FILTER_OUTPUT(MAIL$F$O_GETMSG(MAIL$_MAIL20,MAIL$Q_SUBJ));
: 2251      2366 3 END
```

```
2252 2367 3 ELSE BEGIN
2253 2368 3     IF MAIL$GET_INPUT(MAIL$Q_SUBJ,MAIL$GETMSG(MAIL$_ISUB30)) EQL RMSS_EOF
2254 2369 3     THEN RETURN TRUE;
2255 2370 3     END;
2256 2371 2 IF .MAIL$GL_FLAGS[MAIF V CTRL CFL]
2257 2372 2 THEN RETURN SIGNAL(MAIL$_SENDABORT);
2258 2373 2
2259 2374 2 ! If /EXTRACT, extract the message out to a temporary file
2260 2375 2
2261 2376 2 IF .EXTRACT_FLAG
2262 2377 2 THEN BEGIN
2263 P 2378 3     $FAB_INIT(FAB=EXTFAB,
2264 P 2379 3         FNS=.TMPFILE1_DESC[DSC$_LENGTH],
2265 P 2380 3         FNA=.TMPFILE1_DESC[DSC$_POINTER],
2266 P 2381 3         RAT=CR,
2267 P 2382 3         RFM=VAR,
2268 P 2383 3         DEQ=3,
2269 P 2384 3         CTX=MAIL$_OPENOUT,
2270 P 2385 3         XAB=TMP_XAB,
2271 P 2386 3         FAC=PUT,
2272 P 2387 3         FOP=SQO,
2273 2388 3         NAM=EXTNAM);
2274 2389 3
2275 2390 3     $XABPRO_INIT(XAB=TMP_XAB);
2276 2391 3     TMP_XAB[XAB$_PRO] = TMP_PRO;
2277 P 2392 3     $NAM_INIT(NAM=EXTNAM,
2278 P 2393 3         RSS=NAM$_MAXRSS,
2279 P 2394 3         RSA=NAM$_BUF,
2280 P 2395 3         ESS=NAM$_MAXRSS,
2281 2396 3         ESA=NAM$_BUF);
2282 2397 3
2283 P 2398 3     $RAB_INIT(RAB=EXTRAB,
2284 P 2399 3         FAB=EXTFAB,
2285 P 2400 3         CTX=MAIL$_OPENOUT,
2286 P 2401 3         UBF=TMP_BUFF,
2287 2402 3         USZ=MAIL$_INBUFSZ);
2288 2403 3
2289 P 2404 3     IF _ERR($CREATE(FAB=EXTFAB,ERR=UTIL$REPORT_IO_ERROR);,
2290 2405 3         RETURN TRUE);
2291 2406 3
2292 P 2407 3     IF _ERR($CONNECT(RAB=EXTRAB,ERR=UTIL$REPORT_IO_ERROR);,
2293 P 2408 3         EXTRAB[FAB$_DLT] = 1;
2294 P 2409 3         $CLOSE(FAB=EXTFAB);
2295 2410 3         RETURN TRUE);
2296 2411 3
2297 2412 3     ! Copy the message text to the temporary file
2298 2413 3
2299 2414 3     EXTRAB[RAB$_CTX] = MAIL$_WRITEERR;
2300 2415 3     PUT STATUS = MAIL$EXTRACT_MESSAGE(1,EXTRAB);
2301 2416 3     IF NOT .PUT_STATUS
2302 2417 4     THEN BEGIN
2303 2418 4         EXTRAB[FAB$_DLT] = 1;
2304 2419 4         $CLOSE(FAB=EXTFAB);
2305 2420 4         RETURN TRUE;
2306 2421 3     END;
2307 2422 3     $CLOSE(FAB=EXTFAB);
2308 2423 3     STR$COPY_DX(COMMAND_DESC,TMPFILE1_DESC);
```



```
2309 2424 2 END;
2310 2425 2
2311 2426 2 Get the text of the new message and send it
2312 2427 2
2313 2428 2 GET_SEND_MSG(COMMAND_DESC);
2314 2429 2
2315 2430 2 If /EXTRACT, delete the temp file
2316 2431 2
2317 2432 2 IF .EXTRACT_FLAG
2318 2433 2 THEN BEGIN
2319 2434 3 EXT$FAB[FAB$V_DLT] = 1;
2320 2435 3 EXT$FAB[FAB$V_NAM] = 1;
2321 2436 3 EXT$FAB[FAB$L_CTX] = MAIL$ OPENIN;
2322 2437 3 $OPEN(FAB=EXT$FAB,ERR=UTIL$REPORT IO_ERROR);
2323 2438 3 EXT$FAB[FAB$L_CTX] = MAIL$ CLOSEIN;
2324 2439 3 $CLOSE(FAB=EXT$FAB,ERR=UTIL$REPORT IO_ERROR);
2325 2440 2 END;
2326 2441 2
2327 2442 2 If successfully sent, flag message as replied
2328 2443 2
2329 2444 2 IF .SEND_STATUS
2330 2445 2 AND .MAIL$GL_FLAGS[MAIF_V_ISAM]
2331 2446 2 THEN BEGIN
2332 2447 3 BIND
2333 2448 3 MMH = MAIL$L_CURMSG[MMB_T_HDREC] + $BYTEOFFSET(MFR_T_DATA) : $BBLOCK;
2334 2449 3
2335 2450 3 MMH[MMH V REPLIED] = TRUE;
2336 2451 3 MAIL$UPDATE_I_MSG();
2337 2452 2 END;
2338 2453 2 RETURN TRUE
2339 2454 1 END;
```

```
20 3A 45 52 02161 P.AFT: .ASCII \RE: \
02165 .BLKB 3
00000004 02168 P.AFS: .LONG 4
000000C0 0216C .ADDRESS P.AFT
```

```
$RMS_PTR= TMP_XAB
```

```
OFFC 00000 .ENTRY MAIL$CMD REPLY, Save R2,R3,R4,R5,R6,R7,R8,- : 2269
5B 00000000G 00 9E 00002 MOVAB UTIL$REPORT IO_ERROR, R11
5A E188 CF 9E 00009 MOVAB PARSE_EDIT_QUAL, R10
59 00000000' 00 9E 0000E MOVAB COMMAND_DESC, R9
58 00000000' 00 9E 00015 MOVAB MAIL$Q SUBJ, R8
5E FE04 CE 9E 0001C MOVAB -508(SP), SP
7E D4 00021 CLRL SEND_STATUS : 2284
6D 02FC CF DE 00023 MOVAL 23$, -(FP)
6E 01 D0 00028 MOVL #1, SEND_STATUS : 2311
01 01 EF 0002B EXTZV #1, #1, CLI EDITQUAL, -(SP) : 2312
6A 01 FB 00032 CALLS #1, PARSE_EDIT_QUAL
57 50 D0 00035 MOVL R0, PUT_STATUS
E05C CF 9F 00038 PUSHAB SD_LAST : 2313
00000000G 00 01 FB 0003C CALLS #1, CLIP$PRESENT
```

FE50	C9	01	01	DF66	50	FO	00043	INSV	RO, #1, #1, CLIFLAGS	
					CF	9F	0004A	PUSHAB	SD_EXTRACT	2314
		00000000G	00		01	FB	0004E	CALLS	#1, CLISPRESENT	
			56		50	DO	00055	MOVL	RO, EXTRACT_FLAG	
			15	FE5C	C9	E9	00058	BLBC	CL1_EXTRACT, 1\$	2315
		00000000G	8F		56	D1	0005D	CMPL	EXTRACT_FLAG, #CLIS_NEGATED	2316
					0C	13	00064	BEQL	1\$	
		00000000G	8F		56	D1	00066	CMPL	EXTRACT_FLAG, #CLIS_LOCNEG	
			56		03	13	0006D	BEQL	1\$	
		00000000G	8F		01	DO	0006F	MOVL	#1, EXTRACT_FLAG	2317
					57	D1	00072	CMPL	PUT_STATUS, #CLIS_LOCNEG	2321
		00000000G	8F		09	13	00079	BEQL	2\$	
					57	D1	0007B	CMPL	PUT_STATUS, #CLIS_NEGATED	2322
					02	12	00082	BNEQ	3\$	
					56	D4	00084	CLRL	EXTRACT_FLAG	2323
					59	DD	00086	PUSHL	R9	2324
		00000000G	00	DFD4	CF	9F	00088	PUSHAB	SD_REPLF	
			0C		02	FB	0008C	CALLS	#2, CLISGET_VALUE	
					5C	E9	00093	BLBC	RO, 4\$	
					56	D4	00096	CLRL	EXTRACT_FLAG	2326
					59	DD	00098	PUSHL	R9	2328
		1D10	CA		01	FB	0009A	CALLS	#1, MAIL\$LOOKUP_FILE	
			78		50	E9	0009F	BLBC	STATUS, 9\$	
			05		56	E9	000A2	BLBC	EXTRACT_FLAG, 5\$	2333
		FE50	C9		01	88	000A5	BISB2	#1, CLIFLAGS	2334
		1983	CA		00	FB	000AA	CALLS	#0, MAIL\$GET_CLUNODES	2339
			14	FF70	C8	E9	000AF	BLBC	MAIL\$GL_FLAGS, 6\$	2344
				F8	A8	9F	000B4	PUSHAB	MAIL\$Q_TO	
					7E	7C	000B7	CLRQ	-(SP)	
					7E	D4	000B9	CLRL	-(SP)	
					58	DD	000BB	PUSHL	R8	
					7E	D4	000BD	CLRL	-(SP)	
		00000000G	00		06	FB	000BF	CALLS	#6, MAIL\$INFO_I_MSG	
					0C	11	000C6	BRB	7\$	
					58	DD	000C8	PUSHL	R8	6\$:
		00000000G	00	F8	A8	9F	000CA	PUSHAB	MAIL\$Q TO	
			0B		02	FB	000CD	CALLS	#2, MAIL\$INFO_MSG	
					50	E8	000D4	BLBS	STATUS, 8\$	7\$:
		00000000G	00		50	DD	000D7	PUSHL	STATUS	
					01	FB	000D9	CALLS	#1, LIB\$SIGNAL	
					38	11	000E0	BRB	9\$	
			50	00000000G	00	DO	000E2	MOVL	MAIL\$L_USRSYSBUF, R0	2345
			01		01	EF	000E9	EXTZV	#1, #1, 31(R0), -(SP)	
			CA		01	FB	000EF	CALLS	#1, SEND_SELF_IF	
				E8	A8	9F	000F4	PUSHAB	MAIL\$G_CNCT	2346
		00000000G	00		01	FB	000F7	CALLS	#1, MAIL\$INI_ADDR	
				E8	A8	9F	000FE	PUSHAB	MAIL\$G_CNCT	2350
				F8	A8	9F	00101	PUSHAB	MAIL\$Q_TO	
		00000000G	00		02	FB	00104	CALLS	#2, MAIL\$ADD_ADDR	
			0F		50	E8	0010B	BLBS	STATUS, 10\$	
			07	FF70	C8	E8	0010E	BLBS	MAIL\$GL_FLAGS, 9\$	
		00000000G	00		00	FB	00113	CALLS	#0, MAIL\$REWIND_MSG	
					0202	31	0011A	BRW	22\$	
				F8	A8	9F	0011D	PUSHAB	MAIL\$Q TO	2351
				007EDE10	8F	DD	00120	PUSHL	#8314384	
		00000000G	00		01	FB	00126	CALLS	#1, MAIL\$GETMSG	
					50	DD	0012D	PUSHL	R0	

00000000G	00		02	FB	0012F	CALLS	#2, MAIL\$PUT_OUTPUT		
		E8	A8	9F	00136	PUSHAB	MAIL\$G_CNCT	2352	
00000000G	00		01	FB	00139	CALLS	#1, MAIL\$NET_END_USERS		
	63	FF71	C8	E8	00140	BLBS	MAIL\$GL_FLAGS+1, 14\$	2353	
			68	B5	00145	TSTW	MAIL\$Q_SUBJ	2359	
			36	13	00147	BEQL	12\$		
203A4552	50	04	A8	D0	00149	MOVL	MAIL\$Q_SUBJ+4, R0	2361	
	8F		60	D1	0014D	CMPL	(R0), #540689746		
			0F	13	00154	BEQL	11\$		
			58	DD	00156	PUSHL	R8	2362	
		FE9C	CF	9F	00158	PUSHAB	P.AFS	2363	
			58	DD	0015C	PUSHL	R8	2362	
00000000G	00		03	FB	0015E	CALLS	#3, STR\$CONCAT		
		007EDE18	58	DD	00165	PUSHL	R8	2365	
			8F	DD	00167	PUSHL	#8314392		
00000000G	00		02	FB	0016D	CALLS	#2, MAIL\$FAO_GETMSG		
			50	DD	00174	PUSHL	R0		
00000000G	00		01	FB	00176	CALLS	#1, MAIL\$FILTER_OUTPUT		
			24	11	0017D	BRB	13\$	2359	
		007EDDF8	8F	DD	0017F	PUSHL	#8314360	2368	
00000000G	00		01	FB	00185	CALLS	#1, MAIL\$GETMSG		
			50	DD	0018C	PUSHL	R0		
			58	DD	0018E	PUSHL	R8		
00000000G	00		02	FB	00190	CALLS	#2, MAIL\$GET_INPUT		
0001827A	8F		50	D1	00197	CMPL	R0, #98938		
			03	12	0019E	BNEQ	13\$		
			017C	31	001A0	BRW	22\$		
	0E	FF71	C8	E9	001A3	BLBC	MAIL\$GL_FLAGS+1, 15\$	2371	
		007E805A	8F	DD	001A8	PUSHL	#8290392	2372	
00000000G	00		01	FB	001AE	CALLS	#1, LIB\$SIGNAL		
				04	001B5	RET			
	03		56	E8	001B6	BLBS	EXTRACT_FLAG, 16\$	2376	
			0109	31	001B9	BRW	20\$		
0050	8F	00	00	2C	001BC	MOVCS	#0, (SP), #0, #80, \$RMS_PTR	2388	
				AD	001C3				
	B0	AD	5003	8F	B0	MOVW	#20483, \$RMS_PTR		
	B4	AD	40	8F	9A	MOVZBL	#64, \$RMS_PTR+4		
	C4	AD		03	B0	MOVW	#3, \$RMS_PTR+20		
	C6	AD		01	90	MOVW	#1, \$RMS_PTR+22		
	C8	AD	007E10A2	8F	D0	MOVL	#8261794, \$RMS_PTR+24		
	CE	AD	0202	8F	B0	MOVW	#514, \$RMS_PTR+30		
	D4	AD	FF74	C9	9E	MOVAB	TMP_XAB, \$RMS_PTR+36		
	D8	AD	FF0C	CD	9E	MOVAB	EXTNAM, \$RMS_PTR+40		
	DC	AD	FC	A9	D0	MOVL	TMPFILE1_DESC+4, \$RMS_PTR+44		
	E4	AD	F8	A9	90	MOVW	TMPFILE1_DESC, \$RMS_PTR+52		
0058	8F	00	6E	00	2C	MOVCS	#0, (SP), #0, #88, \$RMS_PTR	2390	
				C9	00203				
	FF74	C9	5813	8F	B0	MOVW	#22547, \$RMS_PTR		
	FF7C	C9	FF44	8F	B0	MOVW	#-188, TMP_XAB+8	2391	
0060	8F	00	6E	00	2C	MOVCS	#0, (SP), #0, #96, \$RMS_PTR	2396	
				CD	0021B				
	FF0C	CD	6002	8F	B0	MOVW	#24578, \$RMS_PTR		
	FF0E	CD		01	8E	MNEGB	#1, \$RMS_PTR+2		
	FF10	CD	0C	AE	9E	MOVAB	NAMBUF, \$RMS_PTR+4		
	FF16	CD		01	8E	MNEGB	#1, \$RMS_PTR+10		
	FF18	CD	0C	AE	9E	MOVAB	NAMBUF, \$RMS_PTR+12		
0044	8F	00	6E	00	2C	MOVCS	#0, (SP), #0, #68, \$RMS_PTR	2402	

FF6C	CD	4401	8F	80	00242	MOVW	#17409, \$RMS_PTR	
84	AD	007E10A2	8F	D0	0024C	MOVL	#8261794, \$RMS_PTR+24	
8C	AD	0200	8F	B0	00254	MOVW	#512, \$RMS_PTR+32	
90	AD	6E	A8	9E	0025A	MOVAB	TMP_BUFF, \$RMS_PTR+36	
A8	AD	B0	AD	9E	0025F	MOVAB	EXTFAB, \$RMS_PTR+60	
			5B	DD	00264	PUSHL	R11	2405
			AD	9F	00266	PUSHAB	EXTFAB	
00000000G	00		02	FB	00269	CALLS	#2, SYSS\$CREATE	
	3A		50	E9	00270	BLBC	STATUS, 18\$	
		FF6C	5B	DD	00273	PUSHL	R11	2410
00000000G	00		CD	9F	00275	PUSHAB	EXTRAB	
	1B		02	FB	00279	CALLS	#2, SYSS\$CONNECT	
84	AD	007E10D0	50	E9	00280	BLBC	STATUS, 17\$	
		FF6C	8F	D0	00283	MOVL	#8261840, EXTRAB+24	2414
			CD	9F	0028B	PUSHAB	EXTRAB	2415
			01	DD	0028F	PUSHL	#1	
00000000G	00		02	FB	00291	CALLS	#2, MAIL\$EXTRACT_MESSAGE	
	57		50	D0	00298	MOVL	R0, PUT_STATUS	
	11		57	E8	0029B	BLBS	PUT_STATUS, 19\$	2416
B5	AD	80	8F	88	0029E	BISB2	#128, EXTRAB+5	2418
		B0	AD	9F	002A3	PUSHAB	EXTFAB	2419
00000000G	00		01	FB	002A6	CALLS	#1, SYSS\$CLOSE	
			70	11	002AD	BRB	22\$	2420
		B0	AD	9F	002AF	PUSHAB	EXTFAB	2422
00000000G	00		01	FB	002B2	CALLS	#1, SYSS\$CLOSE	
		F8	A9	9F	002B9	PUSHAB	TMPFILE1_DESC	2423
			59	DD	002BC	PUSHL	R9	
00000000G	00		02	FB	002BE	CALLS	#2, STR\$COPY_DX	
			59	DD	002C5	PUSHL	R9	2428
00000000V	00		01	FB	002C7	CALLS	#1, GET_SEND_MSG	
	30		56	E9	002CE	BLBC	EXTRACT_FLAG, 21\$	2432
B5	AD	00010080	8F	C8	002D1	BISL2	#65664, EXTRAB+7	2435
C8	AD	007E109A	8F	D0	002D9	MOVL	#8261786, EXTRAB+24	2436
			5B	DD	002E1	PUSHL	R11	2437
		B0	AD	9F	002E3	PUSHAB	EXTFAB	
00000000G	00		02	FB	002E6	CALLS	#2, SYSS\$OPEN	
C8	AD	007E1050	8F	D0	002ED	MOVL	#8261712, EXTRAB+24	2438
			5B	DD	002F5	PUSHL	R11	2439
		B0	AD	9F	002F7	PUSHAB	EXTFAB	
00000000G	00		02	FB	002FA	CALLS	#2, SYSS\$CLOSE	
	1B		6E	E9	00301	BLBC	SEND_STATUS, 22\$	2444
	16	FF70	C8	E9	00304	BLBC	MAIL\$GL_FLAGS, 22\$	2445
50 00C00000G	00	00000077	8F	C1	00309	ADDL3	#119, MAIL\$CURMSG, R0	2448
	60		02	88	00315	BISB2	#2, (R0)	2450
00000000G	00		00	FB	00318	CALLS	#0, MAIL\$UPDATE_1_MSG	2451
	50		01	D0	0031F	MOVL	#1, R0	2453
				04	00322	RET		2454
			0000	00323	23\$:	.WORD	Save nothing	2284
	50	08	AC	D0	00325	MOVL	8(AP), R0	
	50	04	A0	D0	00329	MOVL	4(R0), R0	
		FE00	C0	9F	0032D	PUSHAB	SEND_STATUS	
			01	DD	00331	PUSHL	#1	
			5E	DD	00333	PUSHL	SP	
FC9D	7E	04	AC	7D	00335	MOVQ	4(AP), -(SP)	
	CF		03	FB	00339	CALLS	#3, REPLY_HANDLER	
				04	0033E	RET		

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

J 5
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2 Page 105
(31)

; Routine Size: 831 bytes, Routine Base: \$CODE\$ + 2170

|; 2340 2455 1

```
2342 1 GLOBAL ROUTINE MAIL$CMD_SEND =
2343 1 ++
2344 1 FUNCTIONAL DESCRIPTION:
2345 1
2346 1 This routine is the SEND command action routine and is called to
2347 1 send a message to a user or users.
2348 1
2349 1 INPUTS:
2350 1
2351 1
2352 1 OUTPUTS:
2353 1
2354 1 Specified file, or SYS$INPUT if none specified, is sent the
2355 1 the addressee(s).
2356 1 --
2357 2 BEGIN
2358 2
2359 2 PARSE EDIT QUAL(.CLI_EDITQUAL[F SEND_EDIT]);
2360 2 CLIFLAGS[LAST_FLAG] = CLIPRESENT(SD_LAST);
2361 2 CLISGET VALUE(SD_SEND1,COMMAND_DESC);
2362 P 2 IF_ERR(MAIL$LOOKUP_FILE(COMMAND_DESC);,
2363 2 RETURN TRUE);
2364 2 MAIL$GL_FLAGS[MAIF_V_ATTACHMENT] = CLIPRESENT(SD_ATTACHMENT);
2365 2 IF .MAIL$GL_FLAGS[MAIF_V_ATTACHMENT]
2366 3 THEN BEGIN
2367 3 STR$COPY_DX(MAIL$Q_ATTDESC,COMMAND_DESC);
2368 3 STR$FREE_DX(COMMAND_DESC);
2369 3 END;
2370 2
2371 2 Get the name of the addressee(s) and add them to the addressee list
2372 2
2373 2 MAIL$GET_CLUNODES(); !Get list of cluster nodes
2374 2 MAIL$GET_INPUT(MAIL$Q_TO,MAIL$GETMSG(MAIL$_ISUB20));
2375 2 IF .MAIL$Q_TO[DSCSW_LENGTH] EQL 0 ! If nothing entered, quit
2376 2 THEN RETURN TRUE;
2377 2 MAIL$UPCASE Q(MAIL$Q_TO);
2378 2 MAIL$INI_ADDR(MAIL$G_CNCT);
2379 2 IF NOT CHECK_SEND(MAIL$ADD_ADDR(MAIL$Q_TO,MAIL$G_CNCT))
2380 2 THEN RETURN TRUE;
2381 2 IF SEND_SELF IF(.MAIL$Q_USR$SYSBUF[VMD V SELFSEND])
2382 2 THEN IF NOT CHECK_SEND(MAIL$ADD_ADDR(MAIL$Q_USERNAME,MAIL$G_CNCT))
2383 2 THEN RETURN TRUE;
2384 2 MAIL$NET_END_USERS(MAIL$G_CNCT);
2385 2
2386 2 Get the subject and text of the message and send it
2387 2
2388 2 IF NOT CLISGET_VALUE(SD_SUBJECT,MAIL$Q_SUBJ)
2389 2 THEN IF NOT MAIL$GET_INPUT(MAIL$Q_SUBJ,MAIL$GETMSG(MAIL$_ISUB30))
2390 2 THEN RETURN TRUE;
2391 2
2392 2 GET_SEND_MSG(COMMAND_DESC);
2393 2
2394 2 RETURN TRUE
2395 1 END;
```

				07FC 00000	.ENTRY	MAIL\$CMD_SEND, Save R2,R3,R4,R5,R6,R7,R8,-		
				5A 00000000V	00 9E 00002	MOVAB	R9,R10	2456
				59 00000000G	00 9E 00009	MOVAB	CHECK_SEND, R10	
				58 00000000G	00 9E 00010	MOVAB	MAIL\$ADD_ADDR, R9	
				57 00000000G	00 9E 00017	MOVAB	MAIL\$GET_INPUT, R8	
				56 00000000G	00 9E 0001E	MOVAB	MAIL\$GETMSG, R7	
				55 00000000G	00 9E 00025	MOVAB	CLISGET_VALUE, R6	
				54 00000000'	00 9E 0002C	MOVAB	CLISPRESENT, R5	
				53 DE4F	CF 9E 00033	MOVAB	COMMAND_DESC, R4	
				52 00000000'	00 9E 00038	MOVAB	PARSE_EDIT_QUAL, R3	
7E	FE58	C4		01	00 EF 0003F	EXTZV	MAIL\$Q_TO, R2	
				63	01 FB 00046	CALLS	#0, #1, CLI_EDITQUAL, -(SP)	2473
					CF 9F 00049	PUSHAB	#1, PARSE_EDIT_QUAL	
				65	01 FB 0004D	CALLS	SD_LAST	2474
FE50	C4		01	01	50 FO 00050	INSV	#1, CLISPRESENT	
					54 DD 00057	PUSHL	R0, #1, #1, CLIFLAGS	
					CF 9F 00059	PUSHAB	R4	2475
				66	02 FB 0005D	CALLS	SD_SEND1	
					54 DD 00060	PUSHL	#2, CLISGET_VALUE	
				1D10	01 FB 00062	CALLS	R4	2477
				62	50 E9 00067	BLBC	#1, MAIL\$LOOKUP_FILE	
					CF 9F 0006A	PUSHAB	STATUS, 2\$	
				65	01 FB 0006E	CALLS	SD_ATTACHMENT	2478
FF7A	C2		01	03	50 FO 00071	INSV	#1, CLISPRESENT	
			15	FF7A	03 E1 00078	BBC	R0, #3, #1, MAIL\$GL_FLAGS+2	
					54 DD 0007E	PUSHL	#3, MAIL\$GL_FLAGS+2, 1\$	2479
					A2 9F 00080	PUSHAB	R4	2481
				00000000G	02 FB 00083	CALLS	MAIL\$Q_ATTDESC	
					54 DD 0008A	PUSHL	#2, STR\$COPY_DX	
				0C000000G	01 FB 0008C	CALLS	R4	2482
				1983	00 FB 00093	CALLS	#1, STR\$FREE1_DX	
					8F DD 00098	PUSHL	#0, MAIL\$GET_CLUNODES	2487
				67	01 FB 0009E	CALLS	#8314352	2488
					50 DD 000A1	PUSHL	#1, MAIL\$GETMSG	
					52 DD 000A3	PUSHL	R0	
				68	02 FB 000A5	CALLS	R2	
					62 B5 000A8	CALLS	#2, MAIL\$GET_INPUT	
					7D 13 000AA	TSTW	MAIL\$Q_TO	2489
					52 DD 000AC	BEQL	5\$	
				00000000G	01 FB 000AE	PUSHL	R2	2491
					A2 9F 000B5	CALLS	#1, MAIL\$UPCASE_Q	
				00000000G	01 FB 000B8	PUSHAB	MAIL\$G_CNCT	2492
					A2 9F 000BF	CALLS	#1, MAIL\$INI_ADDR	
					52 DD 000C2	PUSHAB	MAIL\$G_CNCT	2493
				69	02 FB 000C4	PUSHL	R2	
					50 DD 000C7	CALLS	#2, MAIL\$ADD_ADDR	
				6A	01 FB 000C9	PUSHL	R0	
				5A	50 E9 000CC	CALLS	#1, CHECK_SEND	
				50 00000000G	00 D0 000CF	BLBC	R0, 5\$	
7E	1F	A0		01	00 EF 000D6	MOVL	MAIL\$L_USRSYSBUF, R0	2495
				1DC0	01 FB 000DC	EXTZV	#0, #1, 31(R0), -(SP)	
					50 E9 000E1	CALLS	#1, SEND_SELF_IF	
					A2 9F 000E4	BLBC	R0, 3\$	
					E8 A2 9F 000E7	PUSHAB	MAIL\$G_CNCT	2496
						PUSHAB	MAIL\$Q_USERNAME	

69		02	FB	000EA	CALLS	#2, MAIL\$ADD_ADDR	:	
		50	DD	000ED	PUSHL	R0	:	
6A		01	FB	000EF	CALLS	#1, CHECK_SEND	:	
34		50	E9	000F2	BLBC	R0, 5\$	:	
	F0	A2	9F	000F5	PUSHAB	MAIL\$G_CNCT	:	2498
00000000G	00	01	FB	000F8	CALLS	#1, MAIL\$NET_END_USERS	:	
	08	A2	9F	000FF	PUSHAB	MAIL\$Q_SUBJ	:	2502
	DC47	CF	9F	00102	PUSHAB	SD SUBJECT	:	
66		02	FB	00106	CALLS	#2, CLISGET_VALUE	:	
14		50	E9	00109	BLBS	R0, 4\$	:	
	007EDDF8	8F	DD	0010C	PUSHL	#8314360	:	2503
67		01	FB	00112	CALLS	#1, MAIL\$GETMSG	:	
		50	DD	00115	PUSHL	R0	:	
	08	A2	9F	00117	PUSHAB	MAIL\$Q_SUBJ	:	
68		02	FB	0011A	CALLS	#2, MAIL\$GET_INPUT	:	
09		50	E9	0011D	BLBC	R0, 5\$	:	
		54	DD	00120	PUSHL	R4	:	2506
00000000V	00	01	FB	00122	CALLS	#1, GET_SEND_MSG	:	
	50	01	DD	00129	MOVL	#1, R0	:	2508
		04	0012C	RET			:	2509

: Routine Size. 301 bytes, Routine Base: \$CODE\$ + 24AF

: 2396 2510 1


```
2398 2511 1 ROUTINE GET_SENMSG(FILE_DESC) =
2399 2512 1 ++
2400 2513 1 FUNCTIONAL DESCRIPTION:
2401 2514 1
2402 2515 1 This routine is called to complete the sending of a message file.
2403 2516 1
2404 2517 1 INPUTS:
2405 2518 1
2406 2519 1 FILE_DESC = Message file name descriptor.
2407 2520 1
2408 2521 1 EDIT_FLAG = 1 if message file is to be edited before sending.
2409 2522 1 = 0 if message file is not to be edited before sending.
2410 2523 1 LAST_FLAG = 1 if last message sent is to be sent again.
2411 2524 1 = 0 if new message is to be sent.
2412 2525 1
2413 2526 1 OUTPUTS:
2414 2527 1
2415 2528 1 Message file is sent to all addressee(s) in addressee list.
2416 2529 1 --
2417 2530 2 BEGIN
2418 2531 2 MAP
2419 2532 2 FILE_DESC: REF BLOCK[,BYTE]; ! File name descriptor
2420 2533 2 LOCAL
2421 2534 2 STATUS; ! Routine status
2422 2535 2
2423 2536 2
2424 2537 2 If the /LAST qualifier was specified and the last file is still open,
2425 2538 2 send it.
2426 2539 2
2427 2540 2 IF .CLIFLAGS[LAST_FLAG] AND .TMP_FAB[FAB$W_IFI] NEQ 0 ! If last file specified
2428 2541 3 THEN BEGIN
2429 2542 3 MAIL$SEND_ADDR(MAIL$G_CNCT); ! Send the previous file
2430 2543 4 IF (IF .MAIL$GL_FLAGS[MAIF_V_ISAM]
2431 2544 4 THEN MAIL$REWIND_I-MSG()
2432 2545 4 ELSE MAIL$REWIND_MSG())
2433 2546 3 THEN
2434 2547 3 MAIL$PUT_OUTPUT(MAIL$GETMSG(MAIL$_PRESRET),0);
2435 2548 3 RETURN TRUE
2436 2549 2 END;
2437 2550 2
2438 2551 2 Close any previously sent file and, depending on whether /EDIT was
2439 2552 2 specified, either call an editor or just prompt for the input message text.
2440 2553 2
2441 2554 2 $CLOSE(FAB=TMP_FAB); ! Close any other file
2442 P 2555 2 $FAB_INIT(FAB=TMP_FAB,
2443 P 2556 2 DEQ=3,
2444 P 2557 2 FNS=.TMPFILE_DESC[DSC$W_LENGTH],
2445 P 2558 2 FNA=.TMPFILE_DESC[DSC$A_POINTER],
2446 P 2559 2 DNM='TXT',
2447 P 2560 2 CTX=MAIL$OPENOUT,
2448 P 2561 2 XAB=TMP_XAB,
2449 P 2562 2 NAM=TMP_NAM,
2450 P 2563 2 FOP=(SQD,TMD),
2451 P 2564 2 RAT=CR,
2452 P 2565 2 RFM=VAR,
2453 2566 2 FAC=(GET,PUT,BRO));
2454 2567 2
```

```
: 2455      2568 2 $XABPRO_INIT(XAB=TMP_XAB);
: 2456      2569 2 TMP_XAB[XAB$W_PRO] = TMP_PRO;
: 2457      2570 2
: 2458      P 2571 2 $NAM_INIT(NAM=TMP_NAM,
: 2459      P 2572 2     RSA=TMP_ESNAM_BUF,
: 2460      P 2573 2     RSS=NAM$C_MAXRSS,
: 2461      P 2574 2     ESA=TMP_ESNAM_BUF,
: 2462      2575 2     ESS=NAM$C_MAXRSS);
: 2463      2576 2
: 2464      P 2577 2 $RAB_INIT(RAB=TMP_RAB,
: 2465      P 2578 2     FAB=TMP_FAB,
: 2466      P 2579 2     CTX=MAIL$OPENOUT,
: 2467      P 2580 2     UBF=TMP_BUFF,
: 2468      2581 2     USZ=MAIL$K_INBUFFSZ);
: 2469      2582 2
: 2470      2583 2 IF .CLIFLAGS[EDIT_FLAG]
: 2471      2584 2 THEN BEGIN
: 2472      2585 3 STATUS = MAIL$EDIT(.FILE_DESC,TMPFILE_DESC);    ! Call the editor
: 2473      2586 3 IF .STATUS
: 2474      2587 4 THEN BEGIN
: 2475      2588 4     !
: 2476      2589 4     ! Ensure we have delete access to the file.
: 2477      2590 4     !
: 2478      2591 5     IF $OPEN(FAB=TMP_FAB,ERR=UTIL$REPORT_IO_ERROR)
: 2479      2592 5     THEN BEGIN
: 2480      2593 5         TMP_XAB[XAB$W_PRO] = TMP_PRO;
: 2481      2594 5         TMP_XAB[XAB$L_UIC] = 0;
: 2482      2595 5         $CLOSE(FAB=TMP_FAB,ERR=UTIL$REPORT_IO_ERROR);
: 2483      2596 5         TMP_FAB[FAB$V_NAM] = 1;
: 2484      2597 4     END;
: 2485      2598 3     END;
: 2486      2599 3     TMP_FAB[FAB$V_DLT] = 1;                                ! Set delete on close
: 2487      2600 3     IF .STATUS
: 2488      2601 3     THEN STATUS = $OPEN(FAB = TMP_FAB,ERR=UTIL$REPORT_IO_ERROR);
: 2489      2602 3     IF .STATUS
: 2490      2603 3     THEN STATUS = $CONNECT(RAB = TMP_RAB,ERR=UTIL$REPORT_IO_ERROR);
: 2491      2604 3     END
: 2492      2605 2 ELSE
: 2493      2606 2 IF .FILE_DESC[DSC$W_LENGTH] EQL 0                        ! If no file specified
: 2494      2607 3 THEN BEGIN
: 2495      2608 3     STATUS = $CREATE(FAB = TMP_FAB,ERR=UTIL$REPORT_IO_ERROR);
: 2496      2609 3     IF .STATUS
: 2497      2610 3     THEN STATUS = $CONNECT(RAB = TMP_RAB,ERR=UTIL$REPORT_IO_ERROR);
: 2498      2611 3     IF .STATUS THEN STATUS = MAIL$ENTER(TMP_RAB);
: 2499      2612 3     END
: 2500      2613 3 ELSE BEGIN
: 2501      2614 3     TMP_FAB[FAB$B_FNS] = .FILE_DESC[DSC$W_LENGTH];
: 2502      2615 3     TMP_FAB[FAB$L_FNA] = .FILE_DESC[DSC$A_POINTER];
: 2503      2616 3     TMP_FAB[FAB$V_TMD] = 0;                                ! Not a temp file
: 2504      2617 3     TMP_FAB[FAB$V_PUT] = 0;                                ! and no PUT's
: 2505      2618 3     TMP_FAB[FAB$L_CTX] = MAIL$OPENIN;
: 2506      2619 3     TMP_RAB[RAB$L_CTX] = MAIL$OPENIN;
: 2507      2620 3     STATUS = $OPEN(FAB = TMP_FAB,ERR=UTIL$REPORT_IO_ERROR);
: 2508      2621 3     IF .STATUS
: 2509      2622 3     THEN STATUS = $CONNECT(RAB = TMP_RAB,ERR=UTIL$REPORT_IO_ERROR);
: 2510      2623 2     END;
: 2511      2624 2 !
```



```
: 2512 2625 2 : If message text successfully obtained, then send it. Otherwise signal
: 2513 2626 2 : the error.
: 2514 2627 2 :
: 2515 2628 2 IF .STATUS
: 2516 2629 2 THEN
: 2517 2630 2 MAIL$SEND_ADDR(MAIL$G_CNCT)
: 2518 2631 2 ELSE
: 2519 2632 2 IF .STATUS EQL MAIL$_SENDABORT
: 2520 2633 2 THEN
: 2521 2634 2 SIGNAL(MAIL$_SENDABORT);
: 2522 2635 2 :
: 2523 2636 2 : If user was previously reading a message, rewind it and display help message.
: 2524 2637 2 :
: 2525 2638 3 IF (IF .MAIL$GL_FLAGS[MAIF_V ISAM]
: 2526 2639 3 THEN MAIL$REWIND_I_MSG()
: 2527 2640 3 ELSE MAIL$REWIND_MSG())
: 2528 2641 2 THEN
: 2529 2642 2 MAIL$PUT_OUTPUT(MAIL$GETMSG(MAIL$_PRESRET),0);
: 2530 2643 2
: 2531 2644 2 RETURN TRUE
: 2532 2645 1 END;
```

54 58 54 2E 025DC P.AFU: .ASCII \.TXT\

```
$RMS_PTR=
$RMS_PTR=
$RMS_PTR=
$RMS_PTR=
TMP_FAB
TMP_XAB
TMP_NAM
MAIL$G_CNCT+646
```

```
OFFC 00000 GET_SEND_MSG:
: 2511 Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11
: 2512 MOVAB MAIL$REWIND_MSG, R11
: 2513 MOVAB MAIL$REWIND_I_MSG, R10
: 2514 MOVAB MAIL$SEND_ADDR, R9
: 2515 MOVAB UTIL$REPORT_IO_ERROR, R8
: 2516 MOVAB $RMS_PTR, R7
: 2517 MOVAB TMP_FAB, R6
: 2518 BBC #1, CLIFLAGS, 4$
: 2519 TSTW TMP_FAB+2
: 2520 BEQL 4$
: 2521 PUSHAB MAIL$G_CNCT
: 2522 CALLS #1, MAIL$SEND_ADDR
: 2523 BLBC MAIL$GL_FLAGS, 1$
: 2524 CALLS #0, MAIL$REWIND_I_MSG
: 2525 BRB 2$
: 2526 CALLS #0, MAIL$REWIND_MSG
: 2527 BLBS R0, 3$
: 2528 BRW 15$
: 2529 BRW 14$
: 2530 PUSHL R6
: 2531 CALLS #1, SYSS$CLOSE
: 2532 MOVCS #0, (SP), #0, #80, $RMS_PTR
: 2533 MOVW #20483, $RMS_PTR
```

5B	00000000G	00	9E	00002	
5A	00000000G	00	9E	00009	
59	00000000G	00	9E	00010	
58	00000000G	00	9E	00017	
57	00000000'	00	9E	0001E	
56	00000000'	00	9E	00025	
22	FF2C	C6	01	E1	0002C
		02	A6	B5	00032
			1D	13	00035
		FD7A	C7	9F	00037
69		01	FB	0003B	
05	FD02	C7	E9	0003E	
6A		00	FB	00043	
		03	11	00046	
68		00	FB	0004B	1\$:
03		50	E8	0004B	2\$:
		01B9	31	0004E	
		019E	31	00051	3\$:
		56	DD	00054	4\$:
0050	8F	00	01	FB	00056
		6E	00	2C	0005D
			66		00064
		66	5003	8F	B0 00065

0058	8F	00	04 A6 50	8F 9A 0006A	MOVZBL	#80, \$RMS_PTR+4	
			14 A5	03 B0 0006F	MOVW	#3, \$RMS_PTR+20	
			16 A6 43	8F 90 00073	MOVW	#67, \$RMS_PTR+22	
			18 A6 007E10A2	8F D0 00078	MOVL	#8261794, \$RMS_PTR+24	
			1E A6 0202	8F B0 00080	MOVW	#514, \$RMS_PTR+30	
			24 A6 50	A6 9E 00086	MOVAB	TMP_XAB, \$RMS_PTR+36	
			28 A6 A0	A6 9E 0008B	MOVAB	TMP_NAM, \$RMS_PTR+40	
			2C A6 00D0	C6 D0 00090	MOVL	TMPFILE_DESC+4, \$RMS_PTR+44	
			30 A6 FF62	CF 9E 0C096	MOVAB	P.AFU, \$RMS_PTR+48	
			34 A6 00CC	C6 90 0009C	MOVW	TMPFILE_DESC, \$RMS_PTR+52	
			35 A6 5E	G4 90 000A2	MOVW	#4, \$RMS_PTR+53	
				00 2C 000A6	MOVCS	#0, (SP), #0, #88, \$RMS_PTR	2568
				A6 000AD			
			50 A6 5813	8F B0 000AF	MOVW	#22547, \$RMS_PTR	
0060	8F	00	58 A6 FF44	8F B0 000B5	MOVW	#-188, TMP_XAB+8	2569
				00 2C 000BB	MOVCS	#0, (SP), #0, #96, \$RMS_PTR	2575
				A6 000C2			
			A0 A6 6002	8F B0 000C4	MOVW	#24578, \$RMS_PTR	
			A2 A6	01 8E 000CA	MNEGB	#1, \$RMS_PTR+2	
			A4 A6 FE2C	C6 9E 000CE	MOVAB	TMP_ESNAM_BUF, \$RMS_PTR+4	
			AA A6	01 8E 000D4	MNEGB	#1, \$RMS_PTR+10	
0044	8F	00	AC A6 FE2C	C6 9E 000D8	MOVAB	TMP_ESNAM_BUF, \$RMS_PTR+12	
				00 2C 000DE	MOVCS	#0, (SP), #0, #68, \$RMS_PTR	2581
				67 000E5			
			67 4401	8F B0 000E6	MOVW	#17409, \$RMS_PTR	
			18 A7 007E10A2	8F D0 000EB	MOVL	#8261794, \$RMS_PTR+24	
			20 A7 0200	8F B0 000F3	MOVW	#512, \$RMS_PTR+32	
			24 A7 FE00	C7 9E 000F9	MOVAB	TMP_BUFF, \$RMS_PTR+36	
			3C A7	66 9E 000FF	MOVAB	TMP_FAB, \$RMS_PTR+60	
				AC D0 00103	MOVL	FILE_DESC, R2	2585
			52 04	C6 E9 00107	BLBC	CLIFCAGS, 6\$	2583
			43 FF2C	C6 9F 0010C	PUSHAB	TMPFILE_DESC	2585
				52 DD 00110	PUSHL	R2	
				02 FB 00112	CALLS	#2, MAIL\$EDIT	
			00000000G 00	50 D0 00119	MOVL	R0, STATUS	
			53	53 E9 0011C	BLBC	STATUS, 5\$	2586
			26	8F BB 0011F	PUSHR	#*M<R6,R8>	2591
				02 FB 00123	CALLS	#2, SYSS\$OPEN	
			00000000G 00	50 E9 0012A	BLBC	R0, 5\$	
			18 A6 FF44	8F B0 0012D	MOVW	#-188, TMP_XAB+8	2593
				A6 D4 00133	CLRL	TMP_XAB+12	2594
				8F BB 00136	PUSHR	#*M<R6,R8>	2595
				02 FB 0013A	CALLS	#2, SYSS\$CLOSE	
			00000000G 00	01 88 00141	BISB2	#1, TMP_FAB+7	2596
			07 A6	8F 88 00145	BISB2	#128, TMP_FAB+5	2599
			05 A6 80	53 E9 0014A	BLBC	STATUS, 10\$	2600
				52 11 0014D	BRB	8\$	2601
				62 B5 0014F	TSTW	(R2)	2606
				2D 12 00151	BNEQ	7\$	
				8F BB 00153	PUSHR	#*M<R6,R8>	2608
			00000000G 00	02 FB 00157	CALLS	#2, SYSS\$CREATE	
			53	50 D0 0015E	MOVL	R0, STATUS	
			68	53 E9 00161	BLBC	STATUS, 10\$	2609
				8F BB 00154	PJSHR	#*M<R7,R8>	2610
			00000000G 00	02 FB 00168	CALLS	#2, SYSS\$CONNECT	
			53	50 D0 0016F	MOVL	R0, STATUS	
			57	53 E9 00172	BLBC	STATUS, 10\$	2611

00000000G	00		57	DD	00175	PUSHL	R7	:
			01	FB	00177	CALLS	#1, MAIL\$ENTER	:
			3D	11	0017E	BRB	9\$	:
34	A6		62	90	00180	7\$:	MOVB (R2), TMP FAB+52	2614
2C	A6	04	A2	D0	00184		MOVL 4(R2), TMP FAB+44	2615
04	A6		10	8A	00189		BICB2 #16, TMP FAB+4	2616
16	A6		01	8A	0018D		BICB2 #1, TMP FAB+22	2617
18	A6	007E109A	8F	D0	00191		MOVL #8261786, TMP FAB+24	2618
18	A7	007E109A	8F	D0	00199		MOVL #8261786, TMP_RAB+24	2619
		0140	8F	BB	001A1	8\$:	PUSHR #^M<R6,R8>	2620
00000000G	00		02	FB	001A5		CALLS #2, SY\$OPEN	:
	53		50	D0	001AC		MOVL R0, STATUS	:
	1A		53	E9	001AF		BLBC STATUS, 10\$	2621
		0180	8F	BB	001B2		PUSHR #^M<R7,R8>	2622
00000000G	00		02	FB	001B6		CALLS #2, SY\$CONNECT	:
	53		50	D0	001BD	9\$:	MOVL R0, STATUS	:
	09		53	E9	001C0		BLBC STATUS, 10\$	2628
		FD7A	C7	9F	001C3		PUSHAB MAIL\$G CNCT	2630
	69		01	FB	001C7		CALLS #1, MAIL\$SEND_ADDR	:
007E805A	8F		16	11	001CA		BRB 11\$	:
			53	D1	001CC	10\$:	CMPL STATUS, #8290394	2632
		007E805A	0D	12	001D3		BNEQ 11\$	:
00000000G	00		8F	DD	001D5		PUSHL #8290394	2634
	05	FD02	01	FB	001DB		CALLS #1, LIB\$SIGNAL	:
	6A		C7	E9	001E2	11\$:	BLBC MAIL\$GL_FLAGS, 12\$	2638
			00	FB	001E7		CALLS #0, MAIL\$REWIND_I_MSG	2639
			03	11	001EA		BRB 13\$	:
	6B		00	FB	001EC	12\$:	CALLS #0, MAIL\$REWIND_MSG	2640
	18		50	E9	001EF	13\$:	BLBC R0, 15\$	:
		007EDE00	7E	D4	001F2	14\$:	CLRL -(SP)	2642
00000000G	00		8F	DD	001F4		PUSHL #8314368	:
			01	FB	001FA		CALLS #1, MAIL\$GETMSG	:
00000000G	00		50	DD	00201		PUSHL R0	:
			02	FB	00203		CALLS #2, MAIL\$PUT_OUTPUT	:
	50		01	D0	0020A	15\$:	MOVL #1, R0	2644
			04	0020D			RET	2645

; Routine Size: 526 bytes, Routine Base: \$CODE\$ + 25E0

; 2533 2646 1

```
2535 2647 1 ROUTINE CHECK_SEND(STATUS) =
2536 2648 1 ++
2537 2649 1 FUNCTIONAL DESCRIPTION:
2538 2650 1
2539 2651 1 This routine is called to check if the ADD_ADDR was completely
2540 2652 1 successful and if it wasn't, the user is asked if mail should be sent
2541 2653 1 to the users that were found.
2542 2654 1
2543 2655 1 INPUTS:
2544 2656 1
2545 2657 1 STATUS = Status from MAIL$ADD_ADDR.
2546 2658 1
2547 2659 1 OUTPUTS:
2548 2660 1
2549 2661 1 Return status = success if mail should be sent.
2550 2662 1 = failure if mail should not be sent.
2551 2663 1 --
2552 2664 2 BEGIN
2553 2665 2 MAP
2554 2666 2 STATUS: BLOCK[,BYTE]; ! Completion status
2555 2667 2
2556 2668 3 RETURN (
2557 2669 3 IF .STATUS[STSSV_SEVERITY] ! If complete success, ok
2558 2670 3 THEN
2559 2671 3 TRUE
2560 2672 4 ELSE BEGIN
2561 2673 4 MAIL$DELEMLINK(MAIL$G_CNCT[CNCT_Q_LNKLST]); ! Delete empty links
2562 2674 5 IF (.STATUS[STSSV_SEVERITY] NEQ STSSK_WARNING)
2563 2675 5 OR (.MAIL$G_CNCT[CNCT_L_NADDRS] EQL 0)
2564 2676 4 THEN
2565 2677 4 STSSK ERROR ! If not partial success, error
2566 2678 4 ELSE MAIL$CONFIRM(DYN_DESC 2, ! Else, ask user what to do
2567 2679 4 MAIL$GETMSG(MAIL$_WANTOSEND))
2568 2680 4 END
2569 2681 3 )
2570 2682 1 END;
```

```
0000 00000 CHECK_SEND:
04 04 AC E9 00002 .WORD Save nothing : 2647
50 01 D0 00006 BLBC STATUS, 1$ : 2669
04 00009 RET #1, R0
00000000G 00 00 9F 0000A 1$: PUSHAB MAIL$G_CNCT+48 : 2673
07 04 AC 93 00010 CALLS #1, MAIL$DELEMLINK : 2674
08 12 0001B BITB STATUS, #7
00000000' 00 00 D5 0001D BNEQ 2$ : 2675
04 12 00023 TSTL MAIL$G_CNCT+56
50 02 D0 00025 2$: BNEQ 3$
04 00028 RET #2, R0 : 2674
00000000G 00 00 7E DE 20 8F DD 00029 3$: PUSHL #8314400 : 2679
01 FB 0002F CALLS #1, MAIL$GETMSG
50 DD 00036 PUSHL R0
```

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

G 6
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2

Page 115
(34)

00000000G 00 00000000' 00 9F 00038
02 FB 0003E
04 00045

PUSHAB DYN_DESC 2
CALLS #2,-MAIL\$CONFIRM
RET

: 2678
:
: 2682

; Routine Size: 70 bytes, Routine Base: \$CODE\$ + 27EE

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

H 6
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2

Page 116
(35)

```
: 2572      2683 1 ROUTINE EXIT_ROUTINE =  
: 2573      2684 1 |++  
: 2574      2685 1 | FUNCTIONAL DESCRIPTION:  
: 2575      2686 1 |  
: 2576      2687 1 |           This routine is called when a $EXIT service is executed to clean  
: 2577      2688 1 |           up.  
: 2578      2689 1 |--  
: 2579      2690 2 BEGIN  
: 2580      2691 2 RETURN MAIL$DELETELINK(MAIL$G_CNCT[CNCT_Q_LNK_LST])  
: 2581      2692 1 END;
```

```
0000 00000 EXIT_ROUTINE:  
00000000G 00 00000000' 00 9F 00002 .WORD Save nothing  
01 FB 00008 PUSHAB MAIL$G_CNCT+48  
04 0000F CALLS #1, MAIL$DELETELINK  
RET
```

```
: 2683  
: 2691  
: 2692
```

; Routine Size: 16 bytes, Routine Base: \$CODE\$ + 2834

MAIL\$MAIL
V04-001

VAX/VMS PERSONAL MAIL UTILITY

1 6
16-Sep-1984 01:01:06
14-Sep-1984 12:42:20

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]MAIL.B32;2 Page 117
(36)

: 2583
: 2584
2693 1 END
2694 0 ELUDOM

.EXTRN LIB\$SIGNAL, LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	820	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$GLOBALS	884	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODES	10308	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
_LIB\$KEY0\$	8	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1)
_LIB\$STATES	84	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1)
_LIB\$KEY1\$	20	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1)
_ABS	0	NOVEC, NOWRT, NORD, NOEXE, NOSHR, LCL, ABS, CON, NOPIC, ALIGN(0)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
-\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	217	1	1000	00:01.4
-\$255\$DUA28:[SYSLIB]TPAMAC.L32;1	42	25	59	14	00:00.1
-\$255\$DUA28:[MAIL.OBJ]MAILDEF.L32;1	457	57	12	26	00:00.4

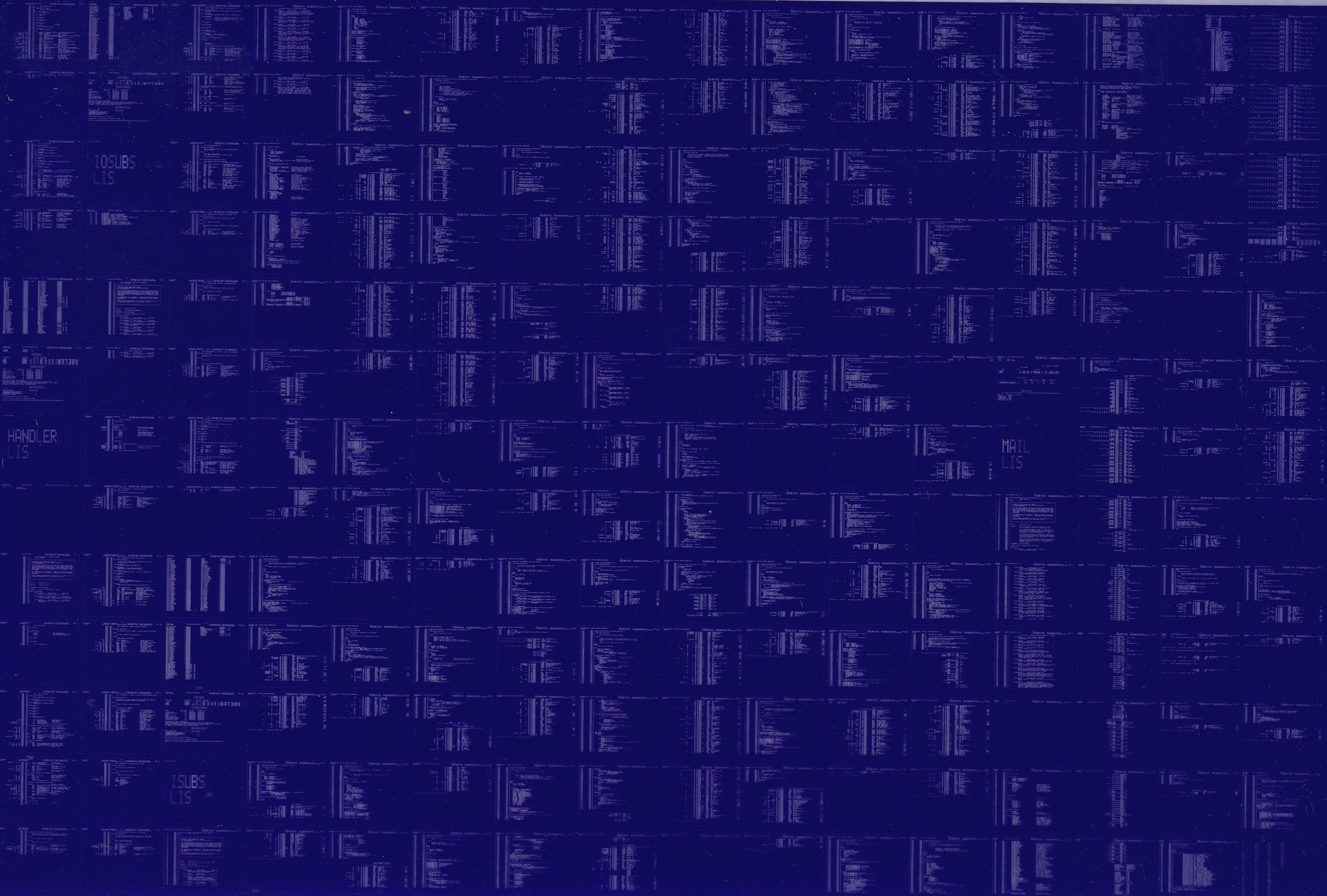
COMMAND QUALIFIERS

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

: Size: 8043 code + 4081 data bytes
: Run Time: 01:47.7
: Elapsed Time: 06:54.4
: Lines/CPU Min: 1500
: Lexemes/CPU-Min: 51859
: Memory Used: 545 pages
: Compilation Complete

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0230 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

MAILMSG
LIS

MAILCDS
LIS

MSGSUBS
LIS

NETSUBS
LIS

NOTIFY
LIS