


```

SSSSSSSS EEEEEEEEE TTTT TTTT
SSSSSSSS EEEEEEEEE TTTT TTTT
SS        EE          TT
SS        EE          TT
SS        EE          TT
SS        EE          TT
SSSSSS    EEEEEEEEE TT
SSSSSS    EEEEEEEEE TT
          SS        TT
          SS        TT
          SS        TT
          SS        TT
SSSSSSSS EEEEEEEEE TT
SSSSSSSS EEEEEEEEE TT
          ....
          ....
          ....
          ....

```

```

LL        IIIIII    SSSSSSS
LL        IIIIII    SSSSSSS
LL        II        SS
LL        II        SS
LL        II        SS
LL        II        SS
LL        II        SSSSS
LL        II        SSSSS
LL        II        SS
LL        II        SS
LL        II        SS
LL        II        SS
LLLLLLLLL IIIIII    SSSSSSS
LLLLLLLLL IIIIII    SSSSSSS

```

```

1 0001 0 MODULE MAIL$SET( %TITLE 'VAX/VMS MAIL UTILITY SET COMMAND'
2 0002 0 IDENT = 'V04-000' ) =
3 0003 1 BEGIN
4 0004 1
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 *   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY   *
9 0009 1 *   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 *   ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 *   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 *   ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 *   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 *   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 *   OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 *   TRANSFERRED. *
18 0018 1 *
19 0019 1 *   THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 *   AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 *   CORPORATION. *
22 0022 1 *
23 0023 1 *   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 *   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *****
27 0027 1
28 0028 1
29 0029 1
30 0030 1 **
31 0031 1 FACILITY:      VAX/VMS MAIL UTILITY
32 0032 1
33 0033 1 ABSTRACT:      SET command handling.
34 0034 1
35 0035 1 AUTHOR:       Benn Schreiber, 5-Dec-1982
36 0036 1
37 0037 1 MODIFIED BY:
38 0038 1
39 0039 1     V03-010 BLS0255      Benn Schreiber      28-Dec-1983
40 0040 1     New global flags
41 0041 1
42 0042 1     V03-009 BLS0246      Benn Schreiber      8-Nov-1983
43 0043 1     Rework for new VMSMAIL.DAT handling, move new mail
44 0044 1     count to VMSMAIL.
45 0045 1
46 0046 1     V03-008 BLS0229      Benn Schreiber      29-Jul-1983
47 0047 1     Add SET NOMAIL_DIRECTORY command
48 0048 1
49 0049 1     V03-007 BLS0225      Benn Schreiber      17-Jun-1983
50 0050 1     Minimize window of confusion in SET MAIL_DIRECTORY. Enforce
51 0051 1     restriction that personal name must start with alphabetic
52 0052 1     character (informal personal name spec agreement with
53 0053 1     RSX and ARPA-interface mail developers).
54 0054 1
55 0055 1     V03-006 BLS0222      Benn Schreiber      7-May-1983
56 0056 1     Disallow setting the wastebasket folder to MAIL or NEWMAIL
57 0057 1

```

MAIL\$SET
V04-000

VAX/VMS MAIL UTILITY SET COMMAND

M 8
16-Sep-1984 01:20:06
14-Sep-1984 12:42:32

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

58	0058	1	V03-005	BLS0220	Benn Schreiber	30-Apr-1983
59	0059	1		Add SET MAIL_DIRECTORY,SET AUTO_PURGE		
60	0060	1				
61	0061	1	V03-004	BLS0217	Benn Schreiber	12-Apr-1983
62	0062	1		Add SET COPY_SELF		
63	0063	1				
64	0064	1	V03-003	BLS0211	Benn Schreiber	13-Mar-1983
65	0065	1		New commands for ISAM mail files		
66	0066	1				
67	0067	1	V03-002	BLS0202	Benn Schreiber	5-Jan-1982
68	0068	1		Check strings for illegal length/characters		
69	0069	1				

```
71 0070 1 |
72 0071 1 | Declarations
73 0072 1 |
74 0073 1 LIBRARY 'SYSSLIBRARY:LIB';
75 0074 1 REQUIRE 'SRC$:MAILREQ';
76 0220 1 LIBRARY 'LIB$:MAILDEF';
77 0221 1 |
78 0222 1 OWN
79 0223 1     COMMAND_DESC: DYN_DESC_DECL,
80 0224 1     DYN_DESC: DYN_DESC_DECL,
81 0225 1     DYN_DESC_2: DYN_DESC_DECL;
82 0226 1 |
83 0227 1 EXTERNAL
84 0228 1     MAIL$RD_RAB : $BBLOCK,           !MAIL file RAB
85 0229 1     MAIL$GC_FLAGS : $BBLOCK,     !Control flags
86 0230 1     MAIL$SD_MAIL : $BBLOCK,      !Descriptor of 'MAIL'
87 0231 1     MAIL$SD_NEWMAIL : $BBLOCK,   !and 'NEWMAIL'
88 0232 1     MAIL$SD_SETF1: $BBLOCK,      !Descriptor for set cmd prompt
89 0233 1     MAIL$SD_SETN1: $BBLOCK,
90 0234 1     MAIL$SD_USER: $BBLOCK,        !Descriptor for USER
91 0235 1     MAIL$Q_RD_DESC : $BBLOCK,      !Current mail file descriptor
92 0236 1     MAIL$T_UAF_BUFF : REF $BBLOCK, !UAF buffer
93 0237 1     MAIL$Q_USERDEF : $BBLOCK,     !dev/dir for mail files
94 0238 1     MAIL$Q_MAILDEF : $BBLOCK,     !above + 'mail.mai'
95 0239 1     MAIL$Q_WASTEDESC : $BBLOCK,   !Descriptor for waste folder
96 0240 1     MAIL$Q_USERNAME: $BBLOCK,   !Descriptor for username
97 0241 1     MAIL$Q_FULLUSER: $BBLOCK,    !and for full name
98 0242 1     MAIL$T_SYSMAIRAB: $BBLOCK,   !RAB for VMSMAIL.DAT
99 0243 1     MAIL$L_MYUIC,                 !UIC of person running MAIL
100 0244 1     MAIL$L_USRSYSBUF: REF $BBLOCK, !User's vmsmail record
101 0245 1     MAIL$L_SYSMABUF: REF $BBLOCK; !Pointer to buffer
102 0246 1 |
103 0247 1 EXTERNAL ROUTINE
104 0248 1     CLISGET_VALUE: ADDRESSING_MODE(GENERAL), !Get value from CLI
105 0249 1     CLISPRESENT: ADDRESSING_MODE(GENERAL), !Test qualifier present
106 0250 1     LIB$CREATE_DIR,                !Create a directory
107 0251 1     LIB$RENAME_FILE,              !Rename files
108 0252 1     MAIL$GETLK_SYSMAI_REC,        !Read and lock VMSMAIL record
109 0253 1     MAIL$MOVE_WASTEBASKET,        !Move contents of wastebasket
110 0254 1     MAIL$OPEN_IF,                 !Open file if not open
111 0255 1     MAIL$UPCASE_Q,                !Cvt to upper case, honor quotes
112 0256 1     MAIL$UPDATE_INFO_REC,        !Update mail file info rec
113 0257 1     MAIL$UPDATE_SYSMAI_REC,     !Update VMSMAIL record
114 0258 1     STR$APPEND,                  !Append a string
115 0259 1     STR$CONCAT: ADDRESSING_MODE(GENERAL), !Concatenate strings
116 0260 1     STR$COPY_DX: ADDRESSING_MODE(GENERAL), !Copy string
117 0261 1     STR$FREE_TDX: ADDRESSING_MODE(GENERAL), !Free dynamic string
118 0262 1     MAIL$GET_USER,                !Get user uaf record
119 0263 1     MAIL$OPEN_FILE,              !Open a mail file
120 0264 1     MAIL$FORW_PRIV_CHECK,        !See if SYSNAM privilege
121 0265 1     UTIL$GETFILENAME,            !Get file spec from FAB
122 0266 1     UTIL$REPORT_IO_ERROR;        !Report I/O error
123 0267 1 |
124 0268 1 | Declare names for string descriptors
125 0269 1 |
126 0270 1 GLOBAL BIND
127 0271 1     MAIL$Q_MAIL_MAI = $DESCRIPTOR('MAIL.MAI') : $BBLOCK;
```

```

: 128
: 129
: 130
: 131
: 132
: 133
: 134
: 135
: 136
: 137
: 138
: 139
: 140
: 141
: 142
: 143
: 144
: 145
: 146
: 147
: 148
: 149
: 150
: 151
: 152
: 153
: 154
: 155
: 156
: 157
: 158
P 0272 1
P 0273 1 SD(
P 0274 1 'FILE',
P 0275 1 'LOG',
P 0276 1 'NOSEND',
P 0277 1 'NOREPLY',
P 0278 1 'SEND',
P 0279 1 'REPLY',
0280 1 'WASTENAME');
0281 1
0282 1 ! Define shared messages
0283 1
P 0284 1 $SHR_MSGDEF(MAIL,126,LOCAL,
P 0285 1 (DIRNOTCRE,ERROR),
P 0286 1 (EXISTS,INFO),
P 0287 1 (RENAMED,INFO),
P 0288 1 (CREATED,INFO),
P 0289 1 (PARSEFAIL,ERROR),
P 0290 1 (SEARCHFAIL,ERROR),
0291 1 (WRITEERR,WARNING));
0292 1 LITERAL
0293 1 PRO_SYS = $BITPOSITION(XABS$V_SYS),
0294 1 PRO_OWN = $BITPOSITION(XABS$V_OWN),
0295 1 PRO_GRP = $BITPOSITION(XABS$V_GRP),
0296 1 PRO_WLD = $BITPOSITION(XABS$V_WLD),
0297 1
0298 1 DIR_PRO = ! Directory file protection
0299 1 ( XABS$M_NODEL) ^ PRO_SYS OR
0300 1 ( XABS$M_NODEL) ^ PRO_OWN OR
0301 1 (XABS$M_NOREAD OR XABS$M_NOWRITE OR XABS$M_NOEXE OR XABS$M_NODEL) ^ PRO_GRP OR
0302 1 (XABS$M_NOREAD OR XABS$M_NOWRITE OR XABS$M_NOEXE OR XABS$M_NODEL) ^ PRO_WLD;

```

```

160 0303 1 GLOBAL ROUTINE MAIL$STRIP_QUOTES(OUT_DESC,IN_DESC) =
161 0304 1 |**
162 0305 1 | FUNCTIONAL DESCRIPTION:
163 0306 1 |
164 0307 1 | This routine strips double quotes (if any) from the input string
165 0308 1 |
166 0309 1 | INPUTS:
167 0310 1 |
168 0311 1 |     IN_DESC = address of string descriptor for input string
169 0312 1 |     OUT_DESC = address of string descriptor for output string
170 0313 1 |
171 0314 1 | --
172 0315 2 BEGIN
173 0316 2 MAP
174 0317 2     IN_DESC : REF $BBLOCK,
175 0318 2     OUT_DESC : REF $BBLOCK;
176 0319 2
177 0320 2 LOCAL
178 0321 2     DESC_1 : $BBLOCK[DSC$C_S_BLN],
179 0322 2     CHAR : BYTE;
180 0323 2
181 0324 2 LITERAL
182 0325 2     DOUBLE_QUOTE = %ASCII '"';
183 0326 2
184 0327 2     DESC_1[DSC$W_LENGTH] = .IN_DESC[DSC$W_LENGTH];
185 0328 2     DESC_1[DSC$B_CLASS] = 0;
186 0329 2     DESC_1[DSC$B_DTYPE] = 0;
187 0330 2     DESC_1[DSC$A_POINTER] = .IN_DESC[DSC$A_POINTER];
188 0331 2
189 0332 2 IF .DESC_1[DSC$W_LENGTH] GTRU 0
190 0333 3 THEN BEGIN
191 0334 3     CHAR = CHRCHAR(.DESC_1[DSC$A_POINTER]);
192 0335 3     IF .DESC_1[DSC$W_LENGTH] GTR T
193 0336 4     AND (.CHAR EQL DOUBLE_QUOTE)
194 0337 4     AND (.CHAR EQL CHRCHAR(.DESC_1[DSC$A_POINTER]+
195 0338 4     .DESC_1[DSC$W_LENGTH] - 1))
196 0339 4     THEN BEGIN
197 0340 4         DESC_1[DSC$W_LENGTH] = .DESC_1[DSC$W_LENGTH] - 2;
198 0341 4         DESC_1[DSC$A_POINTER] = .DESC_1[DSC$A_POINTER] + 1;
199 0342 3     END;
200 0343 2 END;
201 0344 2
202 0345 2 RETURN STR$COPY_DX(.OUT_DESC,DESC_1)
203 0346 1 END;

```

.TITLE MAIL\$SET VAX/VMS MAIL UTILITY SET COMMAND
.IDENT \V04-000\

.PSECT \$CODE\$,NOWRT,2

49	41	4D	2E	4C	49	41	4D	00000	P.AAB:	.ASCII	\MAIL.MAI\	:
								00000008	00008	P.AAA:	.LONG	8
								00000000	0000C		.ADDRESS	P.AAB
				45	4C	49	46	00010	P.AAD:	.ASCII	\FILE\	:
								00000004	00014	P.AAC:	.LONG	4
								0000000C	00018		.ADDRESS	P.AAD

```

47 4F 4C 0001C P.AAF: .ASCII \LOG\
0001F .BLKB 1
00000003 00020 P.AAE: .LONG 3
00000000' 00024 .ADDRESS P.AAF
44 4E 45 53 4F 4E 00028 P.AAH: .ASCII \NOSEND\
0002E .BLKB 2
00000006 00030 P.AAG: .LONG 6
00000000' 00034 .ADDRESS P.AAH
59 4C 50 45 52 4F 4E 00038 P.AAJ: .ASCII \NOREPLY\
0003F .BLKB 1
00000007 00040 P.AAI: .LONG 7
00000000' 00044 .ADDRESS P.AAJ
44 4E 45 53 00048 P.AAL: .ASCII \SEND\
00000004 0004C P.AAK: .LONG 4
00000000' 00050 .ADDRESS P.AAL
59 4C 50 45 52 00054 P.AAN: .ASCII \REPLY\
00059 .BLKB 3
00000005 0005C P.AAM: .LONG 5
00000000' 00060 .ADDRESS P.AAN
45 4D 41 4E 45 54 53 41 57 00064 P.AAP: .ASCII \WASTENAME\
0006D .BLKB 3
00000009 00070 P.AAO: .LONG 9
00000000' 00074 .ADDRESS P.AAP

```

.PSECT \$OWNS,NOEXE,2

```

00# 00000 COMMAND_DESC:
02 0E 00002 .BYTE 0[2]
00004 .BYTE 14, 2
00008 .BLKB 4
00# 00008 DYN_DESC:
02 0E 0000A .BYTE 0[2]
0000C .BYTE 14, 2
00010 .BLKB 4
00# 00010 DYN_DESC_2:
02 0E 00012 .BYTE 0[2]
00014 .BYTE 14, 2
00018 .BLKB 4

```

```

MAILSQ MAIL_MAI=- P.AAA
SD_FILE= P.AAC
SD_LOG= P.AAE
SD_NOSEND= P.AAG
SD_NOREPLY= P.AAI
SD_SEND= P.AAK
SD_REPLY= P.AAM
SD_WASTENAME= P.AAO
.EXTRN MAILST_RD_RAB, MAIL$GL_FLAGS
.EXTRN MAIL$SD_MAIL, MAIL$SD_NEWMAIL
.EXTRN MAIL$SD_SETF1, MAIL$SD_SETN1
.EXTRN MAIL$SD_USER, MAIL$Q_RD_DESC
.EXTRN MAILST_OAF_BUFF
.EXTRN MAIL$Q_USERDEF, MAIL$Q_MAILDEF
.EXTRN MAIL$Q_WASTEDESC
.EXTRN MAIL$Q_USERNAME
.EXTRN MAIL$Q_FULLUSER
.EXTRN MAILST_SYSMAIRAB

```



```

.EXTRN MAIL$MYUIC, MAIL$USRSYSBUF
.EXTRN MAIL$SYSMAIBUF
.EXTRN CLISGET VALUE, CLISPRESENT
.EXTRN LIB$CREATE_DIR, LIB$RENAME_FILE
.EXTRN MAIL$GETLK_SYSMAI_REC
.EXTRN MAIL$MOVE_WASTEBASKET
.EXTRN MAIL$OPEN_IF, MAIL$UPCASE_Q
.EXTRN MAIL$UPDATE_INFO_REC
.EXTRN MAIL$UPDATE_SYSMAI_REC
.EXTRN STR$APPEND, STR$CONCAT
.EXTRN STR$COPY_DX, STR$FREE1_DX
.EXTRN MAIL$GET_USER, MAIL$OPEN_FILE
.EXTRN MAIL$FORW_PRIV_CHECK
.EXTRN UTIL$GETFILENAME
.EXTRN UTIL$REPORT_IO_ERROR

```

.PSECT \$CODE\$,NOWRT,2

```

0000 00000
04 C2 00002
08 AC D0 00005
7E 60 3C 00009
04 AE A0 D0 0000C
50 6E 3C 00011
51 1E 13 00014
01 BE 90 00016
50 B1 0001A
22 15 1B 0001D
51 91 0001F
50 10 12 00022
FF AE C0 00024
A0 51 91 00028
6E 06 12 0002C
04 AE D6 00031
5E DD 00034 1$:
04 AC DD 00036
00000000G 00 02 FB 00039
04 04 00040

```

```

.ENTRY MAIL$STRIP_QUOTES, Save nothing : 0303
SUBL2 #4, SP
MOVL IN_DESC, R0 : 0327
MOVZWL (R0), DESC_1
MOVL 4(R0), DESC_1+4 : 0330
MOVZWL DESC_1, R0 : 0332
BEQL 1$
MOVB @DESC_1+4, CHAR : 0334
CMPW R0, #T : 0335
BLEQU 1$
CMPB CHAR, #34 : 0336
BNEQ 1$
ADDL2 DESC_1+4, R0 : 0338
CMPB CHAR, -1(R0)
BNEQ 1$
SUBW2 #2, DESC_1 : 0340
INCL DESC_1+4 : 0341
PUSHL SP : 0345
PUSHL OUT_DESC
CALLS #2, -STR$COPY_DX
RET : 0346

```

; Routine Size: 65 bytes, Routine Base: \$CODE\$ + 0078

```

: 205 0347 1 ROUTINE CHECK_STRING(DESC,MAXLENGTH) =
: 206 0348 1 ++
: 207 0349 1 FUNCTIONAL DESCRIPTION:
: 208 0350 1
: 209 0351 1 Check a string for too long or illegal characters
: 210 0352 1
: 211 0353 1 INPUTS:
: 212 0354 1
: 213 0355 1 DESC = Addr of string descriptor for string
: 214 0356 1 MAXLENGTH = Max length allowable for string
: 215 0357 1
: 216 0358 1 ROUTINE VALUE:
: 217 0359 1
: 218 0360 1 true if all ok, else error
: 219 0361 1 --
: 220 0362 2 BEGIN
: 221 0363 2
: 222 0364 2 MAP
: 223 0365 2 DESC : REF $BLOCK;
: 224 0366 2
: 225 0367 2 LOCAL
: 226 0368 2 CHAR : BYTE,
: 227 0369 2 PTR;
: 228 0370 2
: 229 0371 2
: 230 0372 2 Check length of string
: 231 0373 2
: 232 0374 2 IF .DESC[DSC$W_LENGTH] GTRU .MAXLENGTH
: 233 0375 3 THEN BEGIN
: 234 0376 3 SIGNAL(MAILS_NAMTOOBIG,1,..DESC);
: 235 0377 3 RETURN MAILS_NAMTOOBIG
: 236 0378 2 END;
: 237 0379 2
: 238 0380 2 Check for characters with high bit on or illegal character (less than
: 239 0381 2 a space, and not a tab)
: 240 0382 2
: 241 0383 2 PTR = .DESC[DSC$A_POINTER];
: 242 0384 2 INCRU I FROM 1 TO .DESC[DSC$W_LENGTH]
: 243 0385 2 DO IF ((CHAR = CHRCHAR_A(PTR)) AND %X'80') NEQ 0
: 244 0386 3 OR ((.CHAR LSSU %C'-' ) AND (.CHAR NEQ %C' '))
: 245 0387 3 THEN BEGIN
: 246 0388 3 SIGNAL(MAILS_ILLCHAR,3,.CHAR,.DESC[DSC$W_LENGTH],.DESC[DSC$A_POINTER]);
: 247 0389 3 RETURN MAILS_ILLCHAR
: 248 0390 2 END;
: 249 0391 2
: 250 0392 2 RETURN TRUE
: 251 0393 1 END;

```

```

                                O0FC 00000 CHECK_STRING:
                                .WORD Save R2,R3,R4,R5,R6,R7
08 AC                          57 00000000G 00 9E 00002 MOVAB LIB$SIGNAL, R7 : 0347
                                52 04 AC D0 00009 MOVL DESC, R2 : 0374
                                10 00 ED 0000D CMPZV #0, #16, (R2), MAXLENGTH

```

		15	1B	00013	BLEQU	1\$	
		52	DD	00015	PUSHL	R2	0376
		01	DD	00017	PUSHL	#1	
	007E8092	8F	DD	00019	PUSHL	#8290450	
67		03	FB	0001F	CALLS	#3, LIB\$SIGNAL	
50	007E8092	8F	D0	00022	MOVL	#8290450, R0	0377
			04	00029	RET		
56		A2	D0	0002A	MOVL	4(R2), PTR	0383
54	04	62	3C	0002E	MOVZWL	(R2), R4	0384
55		01	D0	00031	MOVL	#1, I	
		2F	11	00034	BRB	5\$	
50		86	9A	00036	MOVZBL	(PTR)+, R0	0385
53		50	90	00039	MOVB	R0, CHAR	
		0A	19	0003C	BLSS	3\$	
20		53	91	0003E	CMPB	CHAR, #32	0386
		20	1E	00041	BGEQU	4\$	
09		53	91	00043	CMPB	CHAR, #9	
		1B	13	00046	BEQL	4\$	
	04	A2	DD	00048	PUSHL	4(R2)	0388
		54	DD	0004B	PUSHL	R4	
7E		53	9A	0004D	MOVZBL	CHAR, -(SP)	
		03	DD	00050	PUSHL	#3	
	007E809A	8F	DD	00052	PUSHL	#8290458	
67		05	FB	00058	CALLS	#5, LIB\$SIGNAL	
50	007E809A	8F	D0	0005B	MOVL	#8290458, R0	0389
			04	00062	RET		
		55	D6	00063	INCL	I	0385
54		55	D1	00065	CMP	I, R4	
		CC	1B	00068	BLEQU	2\$	
50		01	D0	0006A	MOVL	#1, R0	0392
			04	0006D	RET		0393

; Routine Size: 110 bytes, Routine Base: \$CODE\$ + 00B9

```
253 0394 1 GLOBAL ROUTINE MAIL$CMD_SET_FORWARD (DELETE_FLAG) =
254 0395 1 ++
255 0396 1 FUNCTIONAL DESCRIPTION:
256 0397 1
257 0398 1 This routine is called to modify the forwarding address of a user.
258 0399 1
259 0400 1 INPUTS:
260 0401 1
261 0402 1 DELETE_FLAG Present if SET NOFORWARD. Parameter absent if SET FORWARD
262 0403 1 MAIL$Q_USERNAME Descriptor of username string
263 0404 1
264 0405 1 IMPLICIT INPUTS:
265 0406 1
266 0407 1 MAIL$T_SYSMAIRAB RAB for I/O on SYS$SYSTEM:VMSMAIL.DAT
267 0408 1
268 0409 1 OUTPUTS:
269 0410 1
270 0411 1 SYS$SYSTEM:VMSMAIL.DAT is created if it does not exist. The user
271 0412 1 record is modified, or added if it does not exist.
272 0413 1
273 0414 1 --
274 0415 2 BEGIN
275 0416 2
276 0417 2 BUILTIN
277 0418 2 NULLPARAMETER;
278 0419 2
279 0420 2 LOCAL
280 0421 2 NAMBUF : $BBLOCK[NAM$C_MAXRSS],
281 0422 2 DIRBUF : $BBLOCK[NAM$C_MAXRSS],
282 0423 2 TMPBUF : $BBLOCK[VMD_C_LENGTH + 3*NAM$C_MAXRSS],
283 0424 2 PTR,
284 0425 2 PROXY_FLAG,
285 0426 2 STATUS;
286 0427 2
287 0428 2 PROXY_FLAG = 0; ! Assume for self
288 0429 2 IF NULLPARAMETER(1)
289 0430 3 THEN BEGIN
290 0431 3 CLISGET VALUE(MAIL$SD_SETF1,COMMAND_DESC); !Get forwarding address
291 0432 3 MAIL$STRIP_QUOTES(COMMAND_DESC,COMMAND_DESC);
292 0433 3 MAIL$UPCASE_Q(COMMAND_DESC);
293 0434 3 END
294 0435 2 ELSE STR$FREE1_DX(COMMAND_DESC); ! Ensure 0 forwarding address
295 0436 2
296 P 0437 2 IF_ERR(CHECK_STRING(COMMAND_DESC,NAM$C_MAXRSS);,
297 0438 2 RETURN .STATUS);
298 0439 2
299 0440 2 IF NOT CLISPRESENT(MAIL$SD_USER) ! If not /USER=, then do self
300 0441 2 THEN CH$MOVE(VMD_C_LENGTH+.MAIL$L_USR$SYSBUF[VMD_B_FWDLNG]
301 0442 2 +.MAIL$L_USR$SYSBUF[VMD_B_FNMLNG]
302 0443 2 +.MAIL$L_USR$SYSBUF[VMD_B_DIRLNG],
303 0444 2 .MAIL$L_USR$SYSBUF,TMPBUF)
304 0445 2 ELSE BEGIN
305 0446 2
306 0447 2 Ensure user has SYSNAM in process privilege mask
307 0448 2
308 0449 2 IF NOT MAIL$FORW_PRIV_CHECK()
309 0450 2 THEN RETURN TRUE;
```

```

310 0451 3
311 0452 3
312 0453 3
313 0454 3
314 0455 3
315 0456 3
316 0457 3
317 0458 3
318 0459 4
319 0460 3
320 0461 3
321 0462 2
322 0463 2
323 0464 2
324 0465 2
325 0466 2
326 0467 2
327 0468 2
328 0469 2
329 0470 2
330 0471 2
331 0472 2
332 0473 2
333 0474 2
334 0475 3
335 0476 3
336 0477 3
337 0478 3
338 0479 3
339 0480 2
340 0481 2
341 0482 2
342 0483 2
343 0484 2
344 0485 2
345 0486 2
346 0487 1

```

```

PROXY FLAG = CLISGET VALUE(MAIL$SD_USER,DYN_DESC);
CH$COPY(.DYN_DESC[DSC$W_LENGTH], ! Copy name from /USER=
.DYN_DESC[DSC$A_POINTER],
VMD_C USERNAME,TMPBUF[VMD_T_USERNAME]);
MAILST_SYMAIRAB[RAB$L_KBF] = TMPBUF; ! Set to read the record
MAILST_SYMAIRAB[RAB$L_UBF] = TMPBUF;
MAILST_SYMAIRAB[RAB$V_NLK] = FALSE;
IF NOT $GET(RAB=MAILST_SYMAIRAB) ! If it doesn't exist, zero record
THEN CH$FILL(0,VMD_C LENG^H+NAM$C MAXRSS-$BYTEOFFSET(VMD_W_FLAGS),
TMPBUF[VMD_W_FLAGS]);
END;
! If trying to forward to self, then set no forwarding
CH$MOVE(.TMPBUF[VMD_B_FNMLNG], ! Save full user name
TMPBUF[VMD_T_FWDNAM]+.TMPBUF[VMD_B_FWDLNG],NAMBUF);
CH$MOVE(.TMPBUF[VMD_B_DIRLNG], ! Save mail dir spec
TMPBUF[VMD_T_FWDNAM]+.TMPBUF[VMD_B_FWDLNG]+.TMPBUF[VMD_B_FNMLNG],
DIRBUF);
IF NOT .PROXY FLAG
AND CH$EQ[VMD_C USERNAME,TMPBUF[VMD_T_USERNAME],
.COMMAND_DESC[DSC$W_LENGTH],.COMMAND_DESC[DSC$A_POINTER], ' ')
THEN TMPBUF[VMD_B_FWDLNG] = 0
ELSE BEGIN
CH$MOVE(.COMMAND_DESC[DSC$W_LENGTH],
.COMMAND_DESC[DSC$A_POINTER],
TMPBUF[VMD_T_FWDNAM]);
TMPBUF[VMD_B_FWDLNG] = .COMMAND_DESC[DSC$W_LENGTH];
END;
PTR = CH$MOVE(.TMPBUF[VMD_B_FNMLNG],NAMBUF, ! Restore full user name
TMPBUF[VMD_T_FWDNAM]+.TMPBUF[VMD_B_FWDLNG]);
CH$MOVE(.TMPBUF[VMD_B_DIRLNG],DIRBUF,.PTR);
MAIL$UPDATE_SYMAI_REC(TMPBUF); ! Update the record
RETURN TRUE
END;

```

				.EXTRN	SYSSGET	
			OFFC 00000	.ENTRY	MAIL\$CMD_SET_FORWARD, Save R2,R3,R4,R5,R6,-	0394
					R7,R8,R9,R10,R11	
5B	00000000G	00	9E 00002	MOVAB	MAIL\$SD_USER, R11	
5A	00000000G	00	9E 00009	MOVAB	CLISGET_VALUE, R10	
59	00000000G	00	9E 00010	MOVAB	MAILST_SYMAIRAB+48, R9	
58	00000000'	00	9E 00017	MOVAB	COMMAND_DESC, R8	
5E	FABC	CE	9E 0001E	MOVAB	-1348(SP), SP	
		57	D4 00023	CLRL	PROXY_FLAG	0428
		6C	95 00025	TSTB	(AP)	0429
		05	13 00027	BEQL	1\$	
	04	AC	D5 00029	TSTL	4(AP)	
		1F	12 0002C	BNEQ	2\$	
		58	DD 0002E 1\$:	PUSHL	R8	0431
	00000000G	00	9F 00030	PUSHAB	MAIL\$SD_SETF1	
6A		02	FB 00036	CALLS	#2, CLISGET_VALUE	

				58	DD	00039	PUSHL	R8		0432
				58	DD	0003B	PUSHL	R8		
	FF0F	CF		02	FB	0003D	CALLS	#2, MAIL\$STRIP_QUOTES		
				58	DD	00042	PUSHL	R8		0433
	00000000G	00		01	FB	00044	CALLS	#1, MAIL\$UPCASE_Q		
				09	11	0004B	BRB	3\$		0429
				58	DD	0004D	PUSHL	R8		0435
	00000000G	00		01	FB	0004F	CALLS	#1, STR\$FREE1_DX		
		7E	FF	8F	9A	00056	MOVZBL	#255, -(SP)		0438
				58	DD	0005A	PUSHL	R8		
	FF31	CF		02	FB	0005C	CALLS	#2, CHECK_STRING		
		01		50	E8	00061	BLBS	STATUS, 4\$		
						04	00064	RET		
				5B	DD	00065	PUSHL	R11		0440
	00000000G	00		01	FB	00067	CALLS	#1, CLIS\$PRESENT		
		21		50	E8	0006E	BLBS	R0, 5\$		
		51	00000000G	00	DO	00071	MOVL	MAIL\$L_USR\$SYSBUF, R0		0441
		52		43	A0	9A	00078	MOVZBL	67(R0), R1	0442
		52		42	A0	9A	0007C	MOVZBL	66(R0), R2	
		51			52	CO	00080	ADDL2	R2, R1	
		52		41	A0	9A	00083	MOVZBL	65(R0), R2	0443
		51		44	A241	9E	00087	MOVAB	68(R2)[R1], R1	
6E		60		51	28	0008C	MOVCS	R1, (R0), TMPBUF		0441
				44	11	00090	BRB	7\$		
	00000000G	00		00	FB	00092	CALLS	#0, MAIL\$FORW_PRIV_CHECK		0449
		03		50	E8	00099	BLBS	R0, 6\$		
				00A1	31	0009C	BRW	10\$		
				08	A8	9F	0009F	PUSHAB	DYN_DESC	0452
				5B	DD	000A2	PUSHL	R11		
		6A		02	FB	000A4	CALLS	#2, CLIS\$GET_VALUE		
		57		50	DO	000A7	MOVL	R0, PROXY_FLAG		
1F		50	20	0C	A8	DO	000AA	MOVL	DYN_DESC+4, R0	0454
		60		08	A8	2C	000AE	MOVCS	DYN_DESC, (R0), #32, #31, TMPBUF	0455
				6E		000B4				
		69		6E	9E	000B5	MOVAB	TMPBUF, MAILST_SYSMAIRAB+48		0456
	F4	A9		6E	9E	000B8	MOVAB	TMPBUF, MAILST_SYSMAIRAB+36		0457
	D6	A9		10	8A	000BC	BICB2	#16, MAILST_SYSMAIRAB+6		0458
				DO	A9	9F	000C0	PUSHAB	MAILST_SYSMAIRAB	0459
	00000000G	00		01	FB	000C3	CALLS	#1, SYSS\$GET		
		09		50	E8	000CA	BLBS	R0, 7\$		
0124	8F	00		00	2C	000CD	MOVCS	#0, (SP), #0, #292, TMPBUF+31		0461
				1F	AE	000D4				
		50		42	AE	9A	000D6	MOVZBL	TMPBUF+66, R0	0466
		56		43	AE	9A	000DA	MOVZBL	TMPBUF+67, R6	0467
	FF00	CD		50	28	000DE	MOVCS	R0, TMPBUF+68[R6], NAMBUF		0466
				51	AE	9A	000E6	MOVZBL	TMPBUF+65, R1	0468
				50	AE	9A	000EA	MOVZBL	TMPBUF+66, R0	0469
				56	CO	000EE	ADDL2	R0, R0		
	FE00	CD		51	28	000F1	MOVCS	R1, TMPBUF+68[R0], DIRBUF		0468
				57	E8	000F9	BLBS	PROXY_FLAG, 8\$		0471
				50	A8	DO	000FC	MOVL	COMMAND_DESC+4, R0	0473
68		20		1F	2D	00100	CMPCS	#31, TMPBUF, #32, COMMAND_DESC, (R0)		0472
				60		00105				
				05	12	00106	BNEQ	8\$		
				43	AE	94	00108	CLRB	TMPBUF+67	0474
				10	11	0010B	BRB	9\$		
		56		68	3C	0010D	MOVZWL	COMMAND_DESC, R6		0476

44	AE		50	04	A8	D0	00110		MOVL	COMMAND_DESC+4, R0	:	0477
			60		56	28	00114		MOVCL	R6, (R0), TMPBUF+68	:	0478
		43	AE		56	90	00119		MOVB	R6, TMPBUF+67	:	0479
			51	42	AE	9A	0011D	9\$:	MOVZBL	TMPBUF+66, R1	:	0481
			50	43	AE	9A	00121		MOVZBL	TMPBUF+67, R0	:	0482
44	AE40	FF00	CD		51	28	00125		MOVCL	R1, NAMBUF, TMPBUF+68[R0]	:	
			50	41	AE	9A	0012D		MOVZBL	TMPBUF+65, R0	:	0483
	63	FE00	CD		50	28	00131		MOVCL	R0, DIRBUF, (PTR)	:	
			00		5E	DD	00137		PUSHL	SP	:	0484
	00000000G		00		01	FB	00139		CALLS	#1, MAIL\$UPDATE_SYSMAI_REC	:	
			50		01	D0	00140	10\$:	MOVL	#1, R0	:	0486
					04	00143			RET		:	0487

; Routine Size: 324 bytes, Routine Base: \$CODE\$ + 0127

```

: 348      0488 1 GLOBAL ROUTINE MAIL$CMD_SET_NOFORWARD =
: 349      0489 1 |++
: 350      0490 1 | FUNCTIONAL DESCRIPTION:
: 351      0491 1 |
: 352      0492 1 |           This routine is called to turn off forwarding for a user
: 353      0493 1 |
: 354      0494 1 | --
: 355      0495 1 |
: 356      0496 1 RETURN MAIL$CMD_SET_FORWARD(1);

```

```

                                0000 00000      .ENTRY MAIL$CMD_SET_NOFORWARD, Save nothing      ; 0488
                                01 DD 00002      PUSHL #1                                           ; 0496
                                01 FB 00004      CALLS #1, MAIL$CMD_SET_FORWARD
                                04 00009      RET

```

; Routine Size: 10 bytes, Routine Base: \$CODE\$ + 026B

0
0


```

358 0497 1 GLOBAL ROUTINE MAIL$CMD_SET_SELF =
359 0498 1 |++
360 0499 1 | FUNCTIONAL DESCRIPTION:
361 0500 1 |
362 0501 1 |     Process the SET COPY_SELF command, which sets the commands on
363 0502 1 |     which an automatic copy is sent to the sender.
364 0503 1 |
365 0504 1 | --
366 0505 2 BEGIN
367 0506 2 LOCAL
368 0507 2     DESC : VECTOR[2, LONG],
369 0508 2     STATUS;
370 0509 2
371 0510 2
372 0511 2 IF MAIL$GETLK_SYSMAI_REC(MAIL$Q_USERNAME, DESC)
373 0512 2 THEN CH$MOVE(.DESC[0], .DESC[1], .MAIL$_USRSYSBUF);
374 0513 2
375 0514 2 IF CLIS$PRESENT(SD_SEND)
376 0515 2 THEN MAIL$_USRSYSBUF[VMD_V_SELFSEND] = 1
377 0516 2 ELSE IF CLIS$PRESENT(SD_NOSEND)
378 0517 2 THEN MAIL$_USRSYSBUF[VMD_V_SELFSEND] = 0;
379 0518 2 IF CLIS$PRESENT(SD_REPLY)
380 0519 2 THEN MAIL$_USRSYSBUF[VMD_V_SELFREPLY] = 1
381 0520 2 ELSE IF CLIS$PRESENT(SD_NOREPLY)
382 0521 2 THEN MAIL$_USRSYSBUF[VMD_V_SELFREPLY] = 0;
383 0522 2
384 0523 2 RETURN MAIL$UPDATE_SYSMAI_REC(.MAIL$_USRSYSBUF)
385 0524 1 END;

```

01FC 00000				.ENTRY	MAIL\$CMD_SET_SELF, Save R2,R3,R4,R5,R6,R7,-	
					R8	0497
	58	00000000G	00 9E 00002	MOVAB	CLIS\$PRESENT, R8	
	57	FDCA	CF 9E 00009	MOVAB	SD_SEND, R7	
	56	00000000G	00 9E 0000E	MOVAB	MAIL\$_USRSYSBUF, R6	
	5E		08 C2 00015	SUBL2	#8, SP	
			5E DD 00018	PUSHL	SP	0511
		00000000G	00 9F 0001A	PUSHAB	MAIL\$Q_USERNAME	
			02 FB 00020	CALLS	#2, MAIL\$GETLK_SYSMAI_REC	
	08		50 E9 00027	BLBC	R0, 1\$	
	50		66 D0 0002A	MOVL	MAIL\$_USRSYSBUF, R0	0512
60	04	BE	6E 28 0002D	MOVCB	DESC, @DESC+4, (R0)	
			57 DD 00032 1\$:	PUSHL	R7	0514
	68		01 FB 00034	CALLS	#1, CLIS\$PRESENT	
	09		50 E9 00037	BLBC	R0, 2\$	
	50		66 D0 0003A	MOVL	MAIL\$_USRSYSBUF, R0	0515
	1F	A0	01 88 0003D	BISB2	#1, 31(R0)	
			10 11 00041	BRB	3\$	
		E4	A7 9F 00043 2\$:	PUSHAB	SD_NOSEND	0516
	68		01 FB 00046	CALLS	#1, CLIS\$PRESENT	
	07		50 E9 00049	BLBC	R0, 3\$	
	50		66 D0 0004C	MOVL	MAIL\$_USRSYSBUF, R0	0517
	1F	A0	01 8A 0004F	BICB2	#1, 31(R0)	
		10	A7 9F 00053 3\$:	PUSHAB	SD_REPLY	0518


```

387 0525 1 GLOBAL ROUTINE MAIL$SET_NAME (FULL_DESC) =
388 0526 1 |++
389 0527 1 | FUNCTIONAL DESCRIPTION:
390 0528 1 |
391 0529 1 |     This routine updates the string MAIL$Q_FULLUSER with the username
392 0530 1 |     from MAIL$Q_USERNAME and the full name passed in FULL_DESC
393 0531 1 |
394 0532 1 | INPUTS:
395 0533 1 |
396 0534 1 |     FUL_DESC = address of descriptor of full name (without quotes)
397 0535 1 |
398 0536 1 | OUTPUTS:
399 0537 1 |
400 0538 1 |     MAIL$Q_FULLUSER updated
401 0539 1 | --
402 0540 2 BEGIN
403 0541 2
404 0542 2 MAP
405 0543 2     FULL_DESC : REF $BBLOCK;
406 0544 2
407 0545 2 BIND
408 0546 2     SPACE_DESC = $DESCRIPTOR(' ');
409 0547 2     QUOTE_DESC = $DESCRIPTOR('');
410 0548 2
411 0549 2 IF .FULL_DESC[DSC$W LENGTH] NEQ 0
412 0550 2     THEN RETURN STR$CONCAT(MAIL$Q_FULLUSER,MAIL$Q_USERNAME,
413 0551 2     SPACE_DESC,QUOTE_DESC,.,FUL_DESC,QUOTE_DESC)
414 0552 2     ELSE RETURN STR$COPY_DX(MAIL$Q_FULLUSER,MAIL$Q_USERNAME)
415 0553 1 END;

```

```

20 002F4 P.AAR: .ASCII \ \
00000001 002F5 .BLKB 3
00000000' 002F8 P.AAQ: .LONG 1
00000000' 002FC .ADDRESS P.AAR
22 00300 P.AAT: .ASCII \ \
00000001 00301 .BLKB 3
00000001 00304 P.AAS: .LONG 1
00000000' 00308 .ADDRESS P.AAT

```

```

SPACE_DESC= P.AAQ
QUOTE_DESC= P.AAS

```

```

54 00000000G 00 001C 00000 .ENTRY MAIL$SET_NAME, Save R2,R3,R4 : 0525
53 00000000G 00 9E 00002 MOVAB MAIL$Q_FOLLUSER, R4 :
52 E5 AF 9E 00010 MOVAB MAIL$Q_USERNAME, R3 :
04 BC B5 00014 MOVAB QUOTE_DESC, R2 :
16 13 00017 TSTW @FULL_DESC : 0549
52 DD 00019 BEQL 1$ :
04 AC DD 0001B PUSHL R2 : 0550
52 DD 0001E PUSHL FULL_DESC : 0551
F4 A2 9F 00020 PUSHAB SPACE_DESC : 0550
53 DD 00023 PUSHL R3 :
54 DD 00025 PUSHL R4 :

```

MAILSSET
V04-000

VAX/VMS MAIL UTILITY SET COMMAND

C 10
16-Sep-1984 01:20:06
14-Sep-1984 12:42:32

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

Page 18
(8)

00000000G 00

06 FB 00027
04 0002E
53 DD 0002F 1\$:
54 DD 00031
02 FB 00033
04 0003A

CALLS #6, STR\$CONCAT
RET
PUSHL R3
PUSHL R4
CALLS #2, STR\$COPY_DX
RET

00000000G 00

: 0552

: 0553

; Routine Size: 59 bytes, Routine Base: \$CODE\$ + 030C

```

: 417 0554 1 ROUTINE ALPHA_CHAR (CHAR) =
: 418 0555 1 |**
: 419 0556 1 | FUNCTIONAL DESCRIPTION:
: 420 0557 1 |
: 421 0558 1 | Tests the character. Returns true if A-Z or a-z, or $ or _, else error
: 422 0559 1 | --
: 423 0560 2 BEGIN
: 424 0561 2 MAP
: 425 0562 2 CHAR : BYTE;
: 426 0563 2
: 427 0564 3 IF ((.CHAR GEQU %C'A') AND (.CHAR LEQU %C'Z'))
: 428 0565 3 OR ((.CHAR GEQU %C'a') AND (.CHAR LEQU %C'z'))
: 429 0566 3 OR (.CHAR EQL %C'$') OR (.CHAR EQL %C'_')
: 430 0567 2 THEN RETURN TRUE
: 431 0568 2 ELSE RETURN FALSE
: 432 0569 1 END;

```

```

                                0000 00000 ALPHA_CHAR:
                                .WORD Save nothing
                                CHAR, R0
: 41 50 04 AC 9A 00002 MOVZBL CHAR, R0 : 0554
: 8F 50 91 00006 CMPB R0, #65 : 0564
: 5A 8F 06 1F 0000A BLSSU 1$
: 8F 50 91 0000C CMPB R0, #90
: 61 8F 17 1B 00010 BLEQU 3$
: 8F 50 91 00012 1$: CMPB R0, #97 : 0565
: 7A 8F 06 1F 00016 BLSSU 2$
: 8F 50 91 00018 CMPB R0, #122
: 24 08 1B 0001C BLEQU 3$
: 8F 50 91 0001E 2$: CMPB R0, #36 : 0566
: 5F 8F 06 13 00021 BEQL 3$
: 8F 50 91 00023 CMPB R0, #95
: 50 04 12 00027 BNEQ 4$
: 01 D0 00029 3$: MOVL #1, R0 : 0568
: 04 0002C RET
: 50 D4 0002D 4$: CLRL R0
: 04 0002F RET : 0569

```

: Routine Size: 48 bytes. Routine Base: \$CODE\$ + 0347

```

: 434 0570 1 ROUTINE SET_NAME (NAME_DESC) =
: 435 0571 1 ++
: 436 0572 1 | FUNCTIONAL DESCRIPTION:
: 437 0573 1 |
: 438 0574 1 |         Set the full username for the SET PERSONAL_NAME command
: 439 0575 1 |
: 440 0576 1 | --
: 441 0577 2 BEGIN
: 442 0578 2
: 443 0579 2 MAP
: 444 0580 2     NAME_DESC : REF $BBLOCK;
: 445 0581 2
: 446 0582 2 LOCAL
: 447 0583 2     DESC : VECTOR[2, LONG],
: 448 0584 2     DIRBUF : $BBLOCK[NAM$C_MAXRSS],
: 449 0585 2     TMPBUF : $BBLOCK[VMD C_LENGTH + 3*NAM$C_MAXRSS],
: 450 0586 2     PTR : REF VECTOR[, BYTE],
: 451 0587 2     STATUS;
: 452 0588 2
: 453 0589 2 MAIL$STRIP_QUOTES(.NAME_DESC, .NAME_DESC);
: 454 0590 2
: 455 P 0591 2 IF_ERR(CHECK_STRING(.NAME_DESC, 127));
: 456 0592 2     RETURN .STATUS);
: 457 0593 2
: 458 0594 2 IF .NAME_DESC[DSC$W_LENGTH] GTRU 0
: 459 0595 3 THEN BEGIN
: 460 0596 3     PTR = .NAME_DESC[DSC$A_POINTER];
: 461 0597 3     IF NOT ALPHA CHAR(.PTR[0])
: 462 0598 3     OR NOT CHR$FAIL(CHR$FIND SUB(.NAME_DESC[DSC$W_LENGTH],
: 463 0599 3     .NAME_DESC[DSC$A_POINTER], 2, UPLIT(' ')))
: 464 0600 3     THEN RETURN SIGNAL(MAIL$_ILLPERNAM, 1, .NAME_DESC);
: 465 0601 2     END;
: 466 0602 2 MAIL$SET_NAME(.NAME_DESC);           ! Set new full name
: 467 0603 2
: 468 0604 2 MAIL$GETLK_SYSM$AI_REC(MAIL$Q_USERNAME, DESC);
: 469 0605 2 CH$MOVE(.DESC[0], .DESC[1], MAIL$_USR$SYSBUF);
: 470 0606 2 CH$MOVE(.MAIL$_USR$SYSBUF[VMD B_DIRLNG],
: 471 0607 2     MAIL$_USR$SYSBUF[VMD T_FWDLNG] + .MAIL$_USR$SYSBUF[VMD B_FWDLNG] +
: 472 0608 2     .MAIL$_USR$SYSBUF[VMD B_FNLNG], DIRBUF);
: 473 0609 2 PTR = MAIL$_USR$SYSBUF[VMD T_FWDLNG] + .MAIL$_USR$SYSBUF[VMD B_FWDLNG];
: 474 0610 2 MAIL$_USR$SYSBUF[VMD B_FNLNG] = .NAME_DESC[DSC$W_LENGTH];
: 475 0611 2 PTR = CH$MOVE(.NAME_DESC[DSC$W_LENGTH], .NAME_DESC[DSC$A_POINTER], PTR[0]);
: 476 0612 2 CH$MOVE(.MAIL$_USR$SYSBUF[VMD B_DIRLNG], DIRBUF, PTR[0]);
: 477 0613 2
: 478 0614 2 RETURN MAIL$UPDATE_SYSM$AI_REC(.MAIL$_USR$SYSBUF)
: 479 0615 1 END;

```

```

00 00 20 20 00377 .BLKB 1
00378 P.AAU: .ASCII \ \<0><0>

```

```

07FC 0000 SET_NAME:
SA FCF6 CF 9E C0002 .WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10 : 0570
MOVAB MAIL$STRIP_QUOTES, R10 :

```

		5E	FBB4	CE	9E	00007	MOVAB	-1100(SP), SP	
		58	04	AC	D0	0000C	MOVL	NAME_DESC, R8	0589
				58	DD	00010	PUSHL	R8	
		6A		02	FB	00014	PUSHL	R8	
		7E	7F	8F	9A	00017	CALLS	#2, MAIL\$STRIP_QUOTES	
				58	DD	0001B	MOVZBL	#127, -(SP)	0592
	41	AA		02	FB	0001D	PUSHL	R8	
		01		50	E8	00021	CALLS	#2, CHECK_STRING	
					04	00024	BLBS	STATUS, 1\$	
				68	B5	00025	RET		
				32	13	00027	TSTW	(R8)	0594
		59	04	A8	D0	00029	BEQL	4\$	
		7E		69	9A	0002D	MOVL	4(R8), PTR	0596
	02CF	CA		01	FB	00030	MOVZBL	(PTR), -(SP)	0597
		11		50	E9	00035	CALLS	#1, ALPHA_CHAR	
04	B8		68	02	39	00038	BLBC	R0, 3\$	
				03	13	0003F	MATCHC	#2, P.AAU, (R8), @4(R8)	0599
		53		02	D0	00041	BEQL	2\$	
		53		02	C2	00044	MOVL	#2, R3	
				12	13	00047	SUBL2	#2, R3	
				58	DD	00049	BEQL	4\$	
				01	DD	0004B	PUSHL	R8	0600
				01	DD	0004B	PUSHL	#1	
				8F	DD	0004D	PUSHL	#8290554	
		00000000G	00	03	FB	00053	CALLS	#3, LIB\$SIGNAL	
					04	0005A	RET		
				58	DD	0005B	PUSHL	R8	0602
	0294	CA		01	FB	0005D	CALLS	#1, MAIL\$SET_NAME	
				F8	AD	9F	PUSHAB	DESC	0604
				00	9F	00065	PUSHAB	MAIL\$Q USERNAME	
		00000000G	00	02	FB	0006B	CALLS	#2, MAIL\$GETLK_SYSMAL_REC	
				00	D0	00072	MOVL	MAIL\$L USRSYSBOF, R6	0605
66	FC	BD	F8	AD	28	00079	MOV3	DESC, @DESC+4, (R6)	
		51	41	A6	9A	0007F	MOVZBL	65(R6), R1	0606
		50	43	A6	9A	00083	MOVZBL	67(R6), R0	0607
57		56		50	C1	00087	ADDL3	R0, R6, R7	
		50	42	A6	9A	0008B	MOVZBL	66(R6), R0	0608
FEF8	CD	44	A047	51	28	0008F	MOV3	R1, 68(R0)[R7], DIRBUF	0606
		59	44	A7	9E	00097	MOVAB	68(R7), PTR	0609
		42	A6	68	90	0009B	MOVB	(R8), 66(R6)	0610
69	04	BB		68	28	0009F	MOV3	(R8), @4(R8), (PTR)	0611
		59		53	D0	000A4	MOVL	R3, PTR	
		50	41	A6	9A	000A7	MOVZBL	65(R6), R0	0612
69	FEF8	CD		50	28	000AB	MOV3	R0, DIRBUF, (PTR)	
				56	DD	000B1	PUSHL	R6	0614
		00000000G	00	01	FB	000B3	CALLS	#1, MAIL\$UPDATE_SYSMAL_REC	
				04	000BA		RET		0615

: Routine Size: 187 bytes. Routine Base: \$CODE\$ + 037C

```

: 481 0616 1 GLOBAL ROUTINE MAIL$CMD_SET_NAME =
: 482 0617 1 |**
: 483 0618 1 | FUNCTIONAL DESCRIPTION:
: 484 0619 1 |
: 485 0620 1 |         Process the SET PERSONAL_NAME command
: 486 0621 1 | --
: 487 0622 2 BEGIN
: 488 0623 2
: 489 0624 2 CLISGET VALUE(MAIL$SD SETN1,COMMAND_DESC);.Get full username
: 490 0625 2 IF .COMMAND_DESC[DSC$D_LENGTH] EQL 0
: 491 0626 2     THEN RETURN TRUE;
: 492 0627 2
: 493 0628 2 RETURN SET_NAME(COMMAND_DESC)
: 494 0629 1 END;

```

		0004 0000	.ENTRY MAIL\$CMD_SET_NAME, Save R2	: 0616
52	00000000'	00 9E 00002	MOVAB COMMAND_DESC, R2	: 0624
	00000000G	52 DD 00009	PUSHL R2	: 0625
00000000G	00	00 9F 0000B	PUSHAB MAIL\$SD SETN1	: 0626
		02 FB 00011	CALLS #2, CLISGET_VALUE	: 0628
		62 B5 00018	TSTW COMMAND_DESC	: 0629
		04 12 0001A	BNEQ 1\$	
	50	01 D0 0001C	MOVL #1, R0	
		04 0001F	RET	
		52 DD 00020 1\$:	PUSHL R2	
	FF1E CF	01 FB 00022	CALLS #1, SET_NAME	
		04 00027	RET	

; Routine Size: 40 bytes, Routine Base: \$CODE\$ + 0437


```

: 496      0630 1 GLOBAL ROUTINE MAIL$CMD_CLEAR_NAME =
: 497      0631 1 |++
: 498      0632 1 | FUNCTIONAL DESCRIPTION:
: 499      0633 1 |
: 500      0634 1 |         Process the SET NOPERSONAL_NAME command (clear the personal name)
: 501      0635 1 |
: 502      0636 1 | |--
: 503      0637 2 BEGIN
: 504      0638 2
: 505      0639 2 LOCAL
: 506      0640 5     DESC : VECTOR[2, LONG];
: 507      0641 2
: 508      0642 2     DESC[0] = 0;
: 509      0643 2     DESC[1] = DESC[0];           !Point to valid address
: 510      0644 2     RETURN SET_NAME(DESC)
: 511      0645 1 END;

```

		0000 0000	.ENTRY	MAIL\$CMD_CLEAR_NAME, Save nothing	: 0630
	5E	04 C2 00002	SUBL2	#4, SP	:
		7E D4 00005	CLRL	DESC	: 0642
04	AE	6E 9E 00007	MOVAB	DESC, DESC+4	: 0643
		5E DD 0000B	PUSHL	SP	: 0644
FF0B	CF	01 FB 0000D	CALLS	#1, SET_NAME	:
		04 00012	RET		: 0645

; Routine Size: 19 bytes, Routine Base: \$CODE\$ + 045F

```

: 513 0646 1 GLOBAL ROUTINE MAIL$CMD_SET_AUTO =
: 514 0647 1 |++
: 515 0648 1 | FUNCTIONAL DESCRIPTION:
: 516 0649 1 |
: 517 0650 1 |         Do the SET AUTO_PURGE command
: 518 0651 1 | --
: 519 0652 2 BEGIN
: 520 0653 2
: 521 0654 2 LOCAL
: 522 0655 2     DESC : VECTOR[2, LONG];
: 523 0656 2
: 524 P 0657 2 IF_ERR(MAIL$GETLK_SYSMAI_REC(MAIL$Q_USERNAME, DESC);,
: 525 0658 2     RETURN TRUE);
: 526 0659 2 CH$MOVE(.DESC[0], .DESC[1], .MAIL$L_USRSYSBUF);
: 527 0660 2 MAIL$L_USRSYSBUF[VMD V NOAUTOPURGE] = 0;
: 528 0661 2 RETURN MAIL$UPDATE_SYSMAI_REC(.MAIL$L_USRSYSBUF)
: 529 0662 1 END;

```

			007C 00000	.ENTRY	MAIL\$CMD_SET_AUTO, Save R2,R3,R4,R5,R6	: 0646
	5E		08 C2 00002	SUBL2	#8, SP	: 0647
			5E DD 00005	PUSHL	SP	: 0648
		00000000G	00 9F 00007	PUSHAB	MAIL\$Q_USERNAME	: 0649
			02 FB 0000D	CALLS	#2, MAIL\$GETLK_SYSMAI_REC	: 0650
	04		50 E8 00014	BLBS	STATUS, 1\$: 0651
	50		01 D0 00017	MOVL	#1, R0	: 0652
			04 0001A	RET		: 0653
		56 00000000G	00 D0 0001B 1\$:	MOVL	MAIL\$L_USRSYSBUF, R6	: 0654
66	04	BE	6E 28 00022	MOVC3	DESC, @DESC+4, (R6)	: 0655
	1F	A6	04 8A 00027	BICB2	#4, 31(R6)	: 0656
			56 DD 0002B	PUSHL	R6	: 0657
		00000000G	01 FB 0002D	CALLS	#1, MAIL\$UPDATE_SYSMAI_REC	: 0658
			04 00034	RET		: 0659

; Routine Size: 53 bytes, Routine Base: \$CODE\$ + 0472

```

: 531 0663 1 GLOBAL ROUTINE MAIL$CMD_SET_NOAUTO =
: 532 0664 1 |++
: 533 0665 1 | FUNCTIONAL DESCRIPTION:
: 534 0666 1 |
: 535 0667 1 |         Do the SET AUTO_PURGE command
: 536 0668 1 | --
: 537 0669 2 BEGIN
: 538 0670 2 LOCAL
: 539 0671 2     DESC : VECTOR[2, LONG];
: 540 0672 2
: 541 P 0673 2 IF_ERR(MAIL$GETLK_SYSMAL_REC(MAIL$Q_USERNAME, DESC);,
: 542 0674 2     RETURN TRUE);
: 543 0675 2 CH$MOVE(.DESC[0], .DESC[1], .MAIL$L_USRSYSBUF);
: 544 0676 2
: 545 0677 2 MAIL$L_USRSYSBUF[VMD_V_NOAUTOPURGE] = 1;
: 546 0678 2
: 547 0679 2 RETURN MAIL$UPDATE_SYSMAL_REC(.MAIL$L_USRSYSBUF)
: 548 0680 1 END;

```

			007C 00000	.ENTRY	MAIL\$CMD_SET_NOAUTO, Save R2,R3,R4,R5,R6	: 0663
	5E		08 C2 00002	SUBL2	#8, SP	: 0674
		00000000G	5E DD 00005	PUSHL	SP	: 0674
			00 9F 00007	PUSHAB	MAIL\$Q_USERNAME	: 0674
	00000000G	00	02 FB 0000D	CALLS	#2, MAIL\$GETLK_SYSMAL_REC	: 0674
		04	50 E8 00014	BLBS	STATUS, 1\$: 0674
		50	01 D0 00017	MOVL	#1, R0	: 0674
				RET		: 0674
		56 00000000G	00 D0 0001A	RET		: 0674
			00 D0 0001B 1\$:	MOVL	MAIL\$L_USRSYSBUF, R6	: 0675
66	04	BE	6E 28 00022	MOVCL	DESC, @DESC+4, (R6)	: 0677
	1F	A6	04 88 00027	BISB2	#4, \$1(R6)	: 0677
			56 DD 0002B	PUSHL	R6	: 0679
	00000000G	00	01 FB 0002D	CALLS	#1, MAIL\$UPDATE_SYSMAL_REC	: 0680
			04 00034	RET		: 0680

: Routine Size: 53 bytes, Routine Base: \$CODE\$ + 04A7

```

550 0681 1 GLOBAL ROUTINE MAIL$CMD_SET_WASTE =
551 0682 1 |++
552 0683 1 | FUNCTIONAL DESCRIPTION:
553 0684 1 |
554 0685 1 |     Process the SET WASTEBASKET_FOLDER command
555 0686 1 | --
556 0687 2 BEGIN
557 0688 2
558 0689 2 LOCAL
559 0690 2     RECDISC : VECTOR[2, LONG],
560 0691 2     RECBUF : $BBLOCK[$BYTEOFFSET(MFR_T_DATA)+MFR_S_FILEKEY+1];
561 0692 2
562 0693 2 BIND
563 0694 2     RECORD TYPE = RECBUF[MFR_Q_DATIM] : VECTOR[ , LONG],
564 0695 2     WASTENAME = RECBUF[MFR_T_DATA] : VECTOR[ , BYTE];
565 0696 2
566 0697 2 CLISGET VALUE(SD WASTENAME, COMMAND_DESC);
567 0698 2 IF .COMMAND_DESC[DSCSW_LENGTH] EQL 0
568 0699 2     THEN RETURN TRUE;
569 0700 2
570 P 0701 2 IF_ERR(MAIL$OPEN IF());
571 0702 2     RETURN TRUE;
572 0703 2
573 0704 2 IF NOT .MAIL$GL_FLAGS[MAIF V ISAM]
574 0705 2     THEN RETURN SIGNAL(MAIL$_NOTISAM);
575 0706 2
576 0707 2 IF CH$EQL(.COMMAND_DESC[DSCSW_LENGTH], .COMMAND_DESC[DSCSA_POINTER],
577 0708 2     .MAIL$SD_MAIL[DSCSW_LENGTH], .MAIL$SD_MAIL[DSCSA_POINTER])
578 0709 2     OR CH$EQL(.COMMAND_DESC[DSCSW_LENGTH], .COMMAND_DESC[DSCSA_POINTER],
579 0710 2     .MAIL$SD_NEWMAIL[DSCSW_LENGTH], .MAIL$SD_NEWMAIL[DSCSA_POINTER])
580 0711 2     THEN RETURN SIGNAL(MAIL$_ILLFOLNAM, 1, COMMAND_DESC);
581 0712 2
582 0713 2 CH$FILL(0, $BYTEOFFSET(MFR_T_DATA), RECBUF);
583 0714 2 RECORD TYPE[0] = MFR_C_WASTENAME;
584 0715 2 WASTENAME[0] = .COMMAND_DESC[DSCSW_LENGTH];
585 0716 2 CH$MOVE(.COMMAND_DESC[DSCSW_LENGTH], .COMMAND_DESC[DSCSA_POINTER],
586 0717 2     WASTENAME[1]);
587 0718 2
588 0719 2 RECDISC[0] = $BYTEOFFSET(MFR_T_DATA) + .COMMAND_DESC[DSCSW_LENGTH] + 1;
589 0720 2 RECDISC[1] = RECBUF;
590 0721 2 IF MAIL$UPDATE_INFO_REC(MAIL$T_RD_RAB, RECDISC)
591 0722 3 THEN BEGIN
592 0723 3 |
593 0724 3 |     Copy anything in old wastebasket folder to new wastebasket folder
594 0725 3 |
595 0726 3     MAIL$MOVE WASTEBASKET(COMMAND_DESC);
596 0727 3     STR$COPY_DX(MAIL$Q_WASTEDESC, COMMAND_DESC);
597 0728 2     END;
598 0729 2 RETURN TRUE
599 0730 1 END;

```

01FC 0000

.ENTRY MAIL\$CMD_SET_WASTE, Save R2,R3,R4,R5,R6,R7,-; 0681
R8

			58	00000000G	00	9E	00002	MOVAB	LIB\$SIGNAL, R8		
			57	00000000'	00	9E	00009	MOVAB	COMMAND_DESC, R7		
			5E		AO	AE	00010	MOVAB	-96(SP), SP		
						57	DD	00014	PUSHL	R7	
					FB7A	CF	9F	00016	PUSHAB	SD_WASTENAME	
			00000000G	00		02	FB	0001A	CALLS	#2, CL\$GET VALUE	
						67	B5	00021	TSTW	COMMAND_DESC	
						03	12	00023	BNEQ	2\$	
			00000000G	00		00A3	31	00025	BRW	6\$	
						00	FB	00028	CALLS	#0, MAIL\$OPEN_IF	
						F3	E9	0002F	BLBC	STATUS, 1\$	
						0A	E8	00032	BLBS	MAIL\$GL_FLAGS, 3\$	
					007E80AA	00	E8	00032	PUSHL	#829047\$	
						8F	DD	00039	CALLS	#1, LIB\$SIGNAL	
						68	01	FB	0003F	RET	
							04	00042			
						55	67	3C	00043	MOVZWL	COMMAND_DESC, R5
						54	A7	D0	00046	MOVL	COMMAND_DESC+4, R4
						50	00	D0	0004A	MOVL	MAIL\$SD_MAIL+4, R0
00000000G	00					64	55	2D	00051	CMPCS	R5, (R4), #0, MAIL\$SD_MAIL, (R0)
							60		0005A		
							13		0005B	BEQL	4\$
						50	00	D0	0005D	MOVL	MAIL\$SD_NEWMAIL+4, R0
00000000G	00					64	55	2D	00064	CMPCS	R5, (R4), #0, MAIL\$SD_NEWMAIL, (R0)
							60		0006D		
							0E	12	0006E	BNEQ	5\$
							57	DD	00070	PUSHL	R7
							01	DD	00072	PUSHL	#1
							8F	DD	00074	PUSHL	#8290530
					007E80E2		03	FB	0007A	CALLS	#3, LIB\$SIGNAL
							04	0007D	RET		
							00	2C	0007E	MOVCS	#0, (SP), #0, #48, RECBUF
							6E		00083		
							02	D0	0C084	MOVL	#2, RECORD TYPE
							67	3C	00087	MOVZWL	COMMAND_DESC, R6
							56	90	0008A	MOVB	R6, WASTENAME
					30		A7	D0	0008E	MOVL	COMMAND_DESC+4, R0
							56	28	00092	MOVCS	R6, (R0), WASTENAME+1
							A6	9E	00097	MOVAB	49(R6), RECDESC
							6E	9E	0009C	MOVAB	RECBUF, RECDESC+4
							AE	9F	000A0	PUSHAB	RECDESC
							00	9F	000A3	PUSHAB	MAIL\$T_RD_RAB
							02	FB	000A9	CALLS	#2, MAIL\$UPDATE_INFO_REC
							50	E9	000B0	BLBC	R0, 6\$
							57	DD	000B3	PUSHL	R7
							01	FB	000B5	CALLS	#1, MAIL\$MOVE_WASTEBASKET
							57	DD	000BC	PUSHL	R7
							00	9F	000BE	PUSHAB	MAIL\$Q_WASTEDESC
							02	FB	000C4	CALLS	#2, STR\$COPY_DX
							01	D0	000CB	MOVL	#1, R0
							04	000CE	RET		

; Routine Size: 207 bytes, Routine Base: \$CODE\$ + 04DC

```

: 601      0731 1 GLOBAL ROUTINE MAIL$CMD_SET_FILE =
: 602      0732 1 |++
: 603      0733 1 | FUNCTIONAL DESCRIPTION:
: 604      0734 1 |
: 605      0735 1 |         process the SET FILE command
: 606      0736 1 |
: 607      0737 1 | --
: 608      0738 2 BEGIN
: 609      0739 2 CLISGET_VALUE(SD_FILE,COMMAND_DESC);
: 610      0740 2
: 611      0741 2 MAIL$OPEN_FILE(COMMAND_DESC);
: 612      0742 2 RETURN TRUE
: 613      0743 1 END;

```

```

          52 00000000' 0004 00000
          FASA      00 9E 00002
          00000000G 00 52 DD 00009
          00000000G 00  CF 9F 0000B
                   02 FB 0000F
                   52 DD 00016
                   01 FB 00018
                   01 D0 0001F
                   04 00022

```

```

.ENTRY MAIL$CMD_SET_FILE, Save R2
MOVAB  COMMAND_DESC, R2
PUSHL  R2
PUSHAB SD_FILE
CALLS  #2, CLISGET_VALUE
PUSHL  R2
CALLS  #1, MAIL$OPEN_FILE
MOVL   #1, R0
RET

```

```

: 0731
: 0739
: 0741
: 0742
: 0743

```

: Routine Size: 35 bytes, Routine Base: \$CODE\$ + 05AB

```

: 615 0744 1 ROUTINE RENAME_SUCCESS (OLD_SPEC,NEW_SPEC,LOG_FLAG) =
: 616 0745 1 |++
: 617 0746 1 | FUNCTIONAL DESCRIPTION:
: 618 0747 1 |
: 619 0748 1 |         This routine is called by LIB$RENAME_FILE on a successful RENAME
: 620 0749 1 | |--
: 621 0750 2 BEGIN
: 622 0751 2
: 623 0752 2 MAP
: 624 0753 2     OLD_SPEC : REF $BBLOCK;
: 625 0754 2
: 626 0755 2 IF CH$EQL(.OLD_SPEC[DSC$W_LENGTH],.OLD_SPEC[DSC$A_POINTER],
: 627 0756 2     .MAIL$Q_RD_DESC[DSC$W_LENGTH],.MAIL$Q_RD_DESC[DSC$A_POINTER])
: 628 0757 2     THEN STR$COPY_DX(MAIL$Q_RD_DESC,.NEW_SPEC);
: 629 0758 2
: 630 0759 2 IF .LOG_FLAG
: 631 0760 2     THEN SIGNAL(MAIL$_RENAMED,2,.OLD_SPEC,.NEW_SPEC);
: 632 0761 2
: 633 0762 2 RETURN TRUE
: 634 0763 1 END;

```

```

                                003C 00000 RENAME_SUCCESS:
                                .WORD      Save R2,R3,R4,R5
                                55 0000000G 00 9E 00002  MOVAB  MAIL$Q_RD_DESC, R5
                                54          04 AC  D0 00009  MOVL   OLD_SPEC, R4
                                50          04 A5  D0 0000D  MOVL   MAIL$Q_RD_DESC+4, R0
65          00          04 B4          64 2D 00011  CMPC5  (R4), #4(R4), #0, MAIL$Q_RD_DESC, (R0)
                                60          00017
                                0C 12 00018  BNEQ   1$
                                08 AC  DD 0001A  PUSHL  NEW_SPEC
                                55 DD 0001D  PUSHL  R5
                                00000000G 00 02 FB 0001F  CALLS  #2, STR$COPY_DX
                                14          0C AC  E9 00026 1$:  BLBC  LOG_FLAG, 2$
                                08 AC  DD 0002A  PUSHL  NEW_SPEC
                                54 DD 0002D  PUSHL  R4
                                02 DD 0002F  PUSHL  #2
                                00000000G 00 07E12A3 8F DD 00031  PUSHL  #8262307
                                50          04 FB 00037  CALLS  #4, LIB$SIGNAL
                                01 D0 0003E 2$:  MOVL  #1, R0
                                04 00041  RET

```

: Routine Size: 66 bytes, Routine Base: \$CODES + 05CE


```

: 663 0790 1 ROUTINE SET_NEW_DIR (SUB_DESC, FULL_DESC, LOG_FLAG) =
: 664 0791 1 +-
: 665 0792 1 FUNCTIONAL DESCRIPTION:
: 666 0793 1
: 667 0794 1 Rename MAIL.MAI into the new directory. Then update
: 668 0795 1 VMSMAIL.DAT to have the new subdirectory spec. Then rename
: 669 0796 1 the rest of the .MAI files
: 670 0797 1
: 671 0798 1 Inputs:
: 672 0799 1
: 673 0800 1 SUB_DESC = address of descriptor of subdir ([.MAIL])
: 674 0801 1 FULL_DESC = address of descriptor of full subdir name (d:[user.mail])
: 675 0802 1 LOG_FLAG = true to /log, false otherwise
: 676 0803 1
: 677 0804 1 --
: 678 0805 2 BEGIN
: 679 0806 2
: 680 0807 2 MAP
: 681 0808 2 SUB_DESC : REF $BLOCK,
: 682 0809 2 FULL_DESC : REF $BLOCK;
: 683 0810 2
: 684 0811 2 LOCAL
: 685 0812 2 DESC : VECTOR[2, LONG];
: 686 0813 2
: 687 0814 2 Rename MAIL.MAI;* to the new directory
: 688 0815 2
: 689 0816 2 STR$CONCAT(DYN_DESC, MAIL$Q_MAIL_MAI, $DESCRIPTOR(';*'));
: 690 P 0817 2 IF_ERR(LIB$RENAME_FILE(DYN_DESC, FULL_DESC, MAIL$Q_USERDEF, 0, 0,
: 691 P 0818 2 RENAME_SUCCESS, RENAME_ERROR, 0, LOG_FLAG);,
: 692 0819 2 RETURN TRUE);
: 693 0820 2
: 694 0821 2
: 695 0822 2 Update record in VMSMAIL. If an error occurs, rename MAIL.MAI;* to
: 696 0823 2 the old directory.
: 697 0824 2
: 698 P 0825 2 IF_ERR(MAIL$GETLK SYSMAI REC(MAIL$Q_USERNAME, DESC);,
: 699 P 0826 2 LIB$RENAME_FILE(DYN_DESC, MAIL$Q_USERDEF, FULL_DESC, 0, 0,
: 700 P 0827 2 RENAME_SUCCESS, RENAME_ERROR, 0, 1); !always log
: 701 0828 2 RETURN TRUE);
: 702 0829 2 CH$MOVE(.DESC[0], DESC[1], MAIL$L_USR$SYSBUF);
: 703 0830 2 CH$MOVE(.SUB_DESC[DSC$W_LENGTH], SUB_DESC[DSC$A_POINTER],
: 704 0831 2 MAIL$L_USR$SYSBUF[VMD_T_FWDNAM]+.MAIL$L_USR$SYSBUF[VMD_B_FWDLNG]+
: 705 0832 2 .MAIL$L_USR$SYSBUF[VMD_B_FNLNG]);
: 706 0833 2 MAIL$L_USR$SYSBUF[VMD_B_DIRLNG] = .SUB_DESC[DSC$W_LENGTH];
: 707 0834 2
: 708 P 0835 2 IF_ERR(MAIL$UPDATE_SYSMAI REC(.MAIL$L_USR$SYSBUF);,
: 709 P 0836 2 LIB$RENAME_FILE(DYN_DESC, MAIL$Q_USERDEF, FULL_DESC, 0, 0,
: 710 P 0837 2 RENAME_SUCCESS, RENAME_ERROR, 0, 1); !always log
: 711 0838 2 RETURN TRUE);
: 712 0839 2
: 713 0840 2
: 714 0841 2 Finally rename the rest of the mail files to the new directory
: 715 0842 2
: 716 0843 2 LIB$RENAME_FILE($DESCRIPTOR('*.MAI;*'), FULL_DESC, MAIL$Q_USERDEF, 0, 0,
: 717 0844 2 RENAME_SUCCESS, RENAME_ERROR, 0, LOG_FLAG);
: 718 0845 2
: 719 0846 2 Set up the strings containing the mail file directory spec and the
```

```

: 720 0847 2 : default mail file spec.
: 721 0848 2 :
: 722 0849 2 STR$COPY DX(MAIL$Q_USERDEF, FULL_DESC);
: 723 0850 2 STR$CONCAT(MAIL$Q_MAILDEF, FULL_DESC, MAIL$Q_MAIL_MAI);
: 724 0851 2 RETURN TRUE
: 725 0852 1 END;

```

```

                2A 3B 00651 P.AAW: .ASCII \;*\  

                00653 .BLKB 1  

                00000002 00654 P.AAV: .LONG 2  

                00000000' 00658 .ADDRESS P.AAW  

2A 3B 49 41 4D 2E 2A 0065C P.AAY: .ASCII \*.MAI;*\  

                00663 .BLKB 1  

                00000007 00664 P.AAX: .LONG 7  

                00000000' 00668 .ADDRESS P.AAY

```

OFFC 00000 SET_NEW_DIR:

```

5B 00000000' 00 9E 00002 .WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 : 0790
5A 00000000G 00 9E 00009 MOVAB DYN_DESC, R11
59 91 AF 9E 00010 MOVAB MAIL$Q_USERDEF, R10
5E 08 C2 00014 MOVAB RENAME_ERROR, R9
CE AF 9F 00017 SUBL2 #8, SP
F97E CF 9F 0001A PUSHAB P.AAV : 0816
5B DD 0001E PUSHAB MAIL$Q_MAIL_MAI
00000000G 00 03 FB 00020 PUSHL R11
OC AC DD 00027 CALLS #3, STR$CONCAT : 0819
7E D4 0002A PUSHL LOG_FLAG
59 DD 0002C CLRL -(SP)
BE A9 9F 0002E PUSHL R9
7E 7C 00031 PUSHAB RENAME_SUCCESS
5A DD 00033 CLRL -(SP)
58 08 AC D0 00035 PUSHL R10
58 DD 00039 MOVL FULL_DESC, R8
5B DD 0003B PUSHL R8
00000000G 00 09 FB 0003D CALLS #9, LIB$RENAME_FILE : 0828
5C 50 E9 00044 BLBC STATUS, 2$
5E DD 00047 PUSHL SP
00000000G 00 00 9F 00049 PUSHAB MAIL$Q_USERNAME : 0829
00000000G 00 02 FB 0004F CALLS #2, MAIL$GETLK_SYSMAI_REC
32 50 E9 00056 BLBC STATUS, 1$
66 04 56 00000000G 00 D0 00059 MOVL MAIL$L_USRSYSBUF, R6 : 0830
BE 6E 28 00060 MOV3 DESC, @DESC+4, (R6) : 0831
57 04 AC D0 00065 MOVL SUB_DESC, R7
50 43 A6 9A 00069 MOVZBL 67(R6), R0
50 56 C0 0006D ADDL2 R6, R0
51 42 A6 9A 00070 MOVZBL 66(R6), R1 : 0832
44 A140 04 B7 67 28 00074 MOV3 (R7), @4(R7), 68(R1)[R0] : 0831
41 A6 67 90 0007B MOV (R7), 65(R6) : 0833
00000000G 00 56 DD 0007F PUSHL R6 : 0838
1A 01 FB 00081 CALLS #1, MAIL$UPDATE_SYSMAI_REC
50 E8 00088 BLBS STATUS, 3$
01 DD 0008B 1$: PUSHL #1

```

		7E	D4	0008D	CLRL	-(SP)	:
		59	DD	0008F	PUSHL	R9	:
	BE	A9	9F	00091	PUSHAB	RENAME_SUCCESS	:
		7E	7C	00094	CLRQ	-(SP)	:
		58	DD	00096	PUSHL	R8	:
		5A	DD	00098	PUSHL	R10	:
		5B	DD	0009A	PUSHL	R11	:
00000000G	00	09	FB	0009C	CALLS	#9, LIB\$RENAME_FILE	:
		39	11	000A3	BRB	4\$:
		OC	AC	DD	000A5	PUSHL	LOG FLAG
		7E	D4	000A8	CLRL	-(SP)	: 0844
		59	DD	000AA	PUSHL	R9	: 0843
	BE	A9	9F	000AC	PUSHAB	RENAME_SUCCESS	:
		7E	7C	000AF	CLRQ	-(SP)	:
	0500	8F	BB	000B1	PUSHR	#*M<R8,R10>	:
	FF3F	CF	9F	000B5	PUSHAB	P.AAX	:
00000000G	00	09	FB	000B9	CALLS	#9, LIB\$RENAME_FILE	:
		58	DD	000C0	PUSHL	R8	: 0849
		5A	DD	000C2	PUSHL	R10	:
00000000G	00	02	FB	000C4	CALLS	#2, STR\$COPY DX	:
	F8CD	CF	9F	000CB	PUSHAB	MAIL\$Q_MAIL_MAI	: 0850
		58	DD	000CF	PUSHL	R8	:
	00000000G	00	9F	000D1	PUSHAB	MAIL\$Q_MAILDEF	:
00000000G	00	03	FB	000D7	CALLS	#3, STR\$CONCAT	:
	50	01	D0	000DE	MOVL	#1, R0	: 0851
		04	000E1	RET			: 0852

: Routine Size: 226 bytes, Routine Base: \$CODE\$ + 066C

```
0853 1 GLOBAL ROUTINE MAIL$CMD_SET_DIR =
0854 1 |++
0855 1 | FUNCTIONAL DESCRIPTION:
0856 1 |
0857 1 |     The SET MAIL_DIRECTORY command specifies where MAIL should put
0858 1 |     all its .MAI and .MSG files.
0859 1 |
0860 1 | --
0861 2 BEGIN
0862 2
0863 2 BIND
0864 2     SUB_STRING = DYN_DESC[DSC$A_POINTER] : REF VECTOR[,BYTE];
0865 2
0866 2 LOCAL
0867 2     TMPBUF : VECTOR[NAM$C_MAXRSS,BYTE],
0868 2     PTR,
0869 2     STATUS,
0870 2     LOG_FLAG,
0871 2     DESC : VECTOR[2, LONG],
0872 2     END_CHAR : REF VECTOR[,BYTE];
0873 2
P 0874 2 IF_ERR(CLIS$GET_VALUE(SD_FILE,COMMAND_DESC);,
0875 2     RETURN TRUE);
0876 2
0877 2 IF .COMMAND_DESC[DSC$W_LENGTH] EQL 0
0878 2     THEN RETURN TRUE;
0879 2
P 0880 2 IF_ERR(CHECK_STRING(COMMAND_DESC,NAM$C_MAXRSS);,
0881 2     RETURN TRUE);
0882 2
0883 2 LOG_FLAG = CLIS$PRESENT(SD_LOG);
0884 2
0885 2 | Syntax check that the parameter is indeed a sub-directory name
0886 2 |
0887 2 STR$COPY_DX(DYN_DESC,COMMAND_DESC);
0888 2 END_CHAR = SUB_STRING[.COMMAND_DESC[DSC$W_LENGTH]-1];
0889 2 IF NOT CH$FAIL(CH$FIND_CH(.COMMAND_DESC[DSC$W_LENGTH],
0890 2     .COMMAND_DESC[DSC$A_POINTER],%C'-'))
0891 2     OR NOT CH$FAIL(CH$FIND_SUB(.COMMAND_DESC[DSC$W_LENGTH],
0892 2     .COMMAND_DESC[DSC$A_POINTER],2,UPLIT('..'))
0893 2     THEN RETURN SIGNAL(MAIL$ILLSUBDIR,1,COMMAND_DESC);
0894 4 IF ((.SUB_STRING[0] EQL %C'[')
0895 3     AND (.END_CHAR[0] NEQ %C']'))
0896 4     OR ((.SUB_STRING[0] EQL %C'<')
0897 3     AND (.END_CHAR[0] NEQ %C'>'))
0898 2     THEN RETURN SIGNAL(MAIL$NOTSUBDIR,1,COMMAND_DESC);
0899 2
0900 3 IF (.SUB_STRING[0] EQL %C'[')
0901 2     THEN SUB_STRING[0] = %C'<';
0902 3 IF (.END_CHAR[0] EQL %C']')
0903 2     THEN END_CHAR[0] = %C'>';
0904 3 IF (.SUB_STRING[0] NEQ %C'<') OR (.SUB_STRING[1] NEQ %C'.')
0905 3     OR (.END_CHAR[0] NEQ %C'>')
0906 2     THEN RETURN SIGNAL(MAIL$NOTSUBDIR,1,COMMAND_DESC);
0907 2
P 0908 2 IF_ERR(MAIL$GET_USER(MAIL$Q_USERNAME);,
0909 2     RETURN TRUE);
```

```

784 0910 2
785 0911 2 DESC[0] = (MAIL$T_UAF_BUFF[UAF$T_DEFDEV])<CS_COUNT>;
786 0912 2 DESC[1] = (MAIL$T_OAF_BUFF[UAF$T_DEFDEV])<CS_STRING>;
787 0913 2 STR$COPY_DX(DYN_DESC_2,DESC);
788 0914 2 DESC[0] = (MAIL$T_UAF_BUFF[UAF$T_DEFDIR])<CS_COUNT>;
789 0915 2 DESC[1] = (MAIL$T_OAF_BUFF[UAF$T_DEFDIR])<CS_STRING>;
790 0916 2 STR$APPEND(DYN_DESC_2,DESC);
791 0917 2
792 0918 2 PTR = CH$MOVE(.DYN_DESC_2[DSC$W_LENGTH]-1,.DYN_DESC_2[DSC$A_POINTER],TMPBUF);
793 0919 2 PTR = CH$MOVE(.DYN_DESC[DSC$W_LENGTH]-1,.DYN_DESC[DSC$A_POINTER]+1,.PTR);
794 0920 2
795 0921 2 DESC[0] = .PTR - TMPBUF;
796 0922 2 DESC[1] = TMPBUF;
797 0923 2 END_CHAR = .DYN_DESC_2[DSC$W_LENGTH] + .DYN_DESC_2[DSC$A_POINTER]-1;
798 0924 2 IF .END_CHAR[0] EQL '%C''
799 0925 2 THEN BEGIN
800 0926 2     END_CHAR = .DESC[0] + .DESC[1] - 1;
801 0927 2     END_CHAR[0] = '%C'';
802 0928 2 END;
803 0929 2
804 0930 2 : First create the directory. Then, rename MAIL.MAI into that directory.
805 0931 2 : If that fails, give up. If that succeeds, then update VMSMAIL.DAT to
806 0932 2 : contain the subdirectory. After that, rename the rest of the *.mai
807 0933 2 : files to the new directory. By doing it this way, the time that the
808 0934 2 : mail.mai file and vmsmail.dat are out of synch is minimized. Note
809 0935 2 : that errors on the rename of the rest of the files are not dealt with
810 0936 2 : in a meaningful way.
811 0937 2
812 0938 2 IF NOT (STATUS = LIB$CREATE_DIR(DESC,MAIL$L_MYUIC,%REF(-1),%REF(DIR_PRO)))
813 0939 2 THEN RETURN SIGNAL(MAIL$ DIRNOTCRE,1,DESC,.STATUS)
814 0940 2 ELSE SIGNAL((IF .STATUS EQL SSS_CREATED
815 0941 2 THEN MAIL$_CREATED
816 0942 2 ELSE MAIL$_EXISTS),1,DESC);
817 0943 2
818 P 0944 2 IF_ERR(SET_NEW_DIR(COMMAND_DESC,DESC,.LOG_FLAG);,
819 0945 2 RETURN .STATUS);
820 0946 2
821 0947 2 RETURN TRUE
822 0948 1 END;

```

00 00 2E 2E 0074E .BLKB 2
00750 P.AAZ: .ASCII \..\<0><0>

SUB_STRING= DYN_DESC+4

		OFFC 00000	.ENTRY	MAIL\$CMD SET DIR, Save R2,R3,R4,R5,R6,R7,-	0853
				R8,R9,R10,R11	
	5B FBBA	CF 9E 00002	MOVAB	SD FILE, R11	
	5A 00000000	00 9E 00007	MOVAB	COMMAND_DESC, R10	
	5E FEFO	CE 9E 0000E	MOVAB	-272(SPT), SP	
		5A DD 00013	PUSHL	R10	0875
		5B DD 00015	PUSHL	R11	
00000000G	00	02 FB 00017	CALLS	#2, CLISGET_VALUE	
	04	50 E9 0001E	BLBC	STATUS, 1\$	

			04	000D7				RET			
		00000000G	00	9F	000D8	13\$:		PUSHAB	MAIL\$Q_USERNAME		0909
		00	01	FB	000DE			CALLS	#1, MAIL\$GET_USER		
		03	50	E8	000E5			BLBS	STATUS, 14\$		
			00EF	31	000E8			BRW	19\$		
		00000000G	00	D0	000EB	14\$:		MOVL	MAIL\$T_UAF_BUFF, R0		0911
	08	AE	74	A0	9A	000F2		MOVZBL	116(R0), DESC		
	0C	AE	75	A0	9E	000F7		MOVAB	117(R0), DESC+4		0912
			08	AE	9F	000FC		PUSHAB	DESC		0913
			10	AA	9F	000FF		PUSHAB	DYN_DESC 2		
		00000000G	00	02	FB	00102		CALLS	#2, -STR\$COPY DX		
		00000000G	00	D0	00109			MOVL	MAIL\$T_UAF_BOFF, R0		0914
	08	AE	0094	C0	9A	00110		MOVZBL	148(R0), DESC		
	0C	AE	0095	C0	9E	00116		MOVAB	149(R0), DESC+4		0915
			08	AE	9F	0011C		PUSHAB	DESC		0916
			10	AA	9F	0011F		PUSHAB	DYN_DESC 2		
		00000000G	00	02	FB	00122		CALLS	#2, -STR\$APPEND		
			57	10	AA	3C	00129	MOVZWL	DYN_DESC 2, R7		0918
			50	FF	A7	9E	0012D	MOVAB	-1(R7), R0		
			58	14	AA	D0	00131	MOVL	DYN_DESC_2+4, R8		
10	AE		68	50	28	00135		MOV3	R0, -(R8), TMPBUF		
			51	08	AA	3C	0013A	MOVZWL	DYN_DESC, R1		0919
				51	D7	0013E		DECL	R1		
			50	0C	AA	D0	00140	MOVL	DYN_DESC+4, R0		
63	01		A0	51	28	00144		MOV3	R1, -1(R0), (PTR)		
			50	10	AE	9E	00149	MOVAB	TMPBUF, R0		0921
08	AE		53	50	C3	0014D		SUBL3	R0, PTR, DESC		
			0C	AE	9E	00152		MOVAB	TMPBUF, DESC+4		0922
			56	FF	A847	9E	00157	MOVAB	-1(R8)[R7], END_CHAR		0923
			5D	8F	66	91	0015C	CMPB	(END_CHAR), #93		0924
				0E	12	00160		BNEQ	15\$		
	50	08	AE	0C	AE	C1	00162	ADDL3	DESC+4, DESC, R0		0926
			56	FF	A0	9E	00168	MOVAB	-1(R0), END_CHAR		
			66	5D	8F	90	0016C	MOVB	#93, (END_CHAR)		0927
			04	AE	FF88	8F	3C	00170	15\$:		0938
				04	AE	9F	00176	MOVZWL	#65416, 4(TSP)		
			04	AE	01	CE	00179	PUSHAB	4(SP)		
				04	AE	9F	0017D	MNEGL	#1, 4(SP)		
				00	9F	00180		PUSHAB	4(SP)		
		00000000G	00	04	FB	00189		PUSHAB	MAIL\$SL_MYUIC		
			15	50	E8	00190		PUSHAB	DESC		
				50	DD	00193		CALLS	#4, LIB\$CREATE_DIR		
				0C	AE	9F	00195	BLBS	STATUS, 16\$		0939
				01	DD	00198		PUSHL	STATUS		
				8F	DD	0019A		PUSHAB	DESC		
		00000000G	00	04	FB	001A0		PUSHL	#1		
				04	04	001A7		PUSHL	#8262338		
				08	AE	9F	001A8	CALLS	#4, LIB\$SIGNAL		
				01	DD	001AB		RET			0940
				50	D1	001AD		PUSHAB	DESC		
				08	12	001B4		PUSHL	#1		
				8F	DD	001B6		CPL	STATUS, #1561		
				06	11	001BC		BNEQ	17\$		
				8F	DD	001BE		PUSHL	#8261747		
				03	FB	001C4		BRB	18\$		
				59	DD	001CB		PUSHL	#8262291		
								CALLS	#3, LIB\$SIGNAL		
								PUSHL	LOG_FLAG		0945

MAILSSET
V04-000

VAX/VMS MAIL UTILITY SET COMMAND

11
16-Sep-1984 01:20:06
14-Sep-1984 12:42:32

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

Page 38
(20)

		OC	AE	9F	001CD	PUSHAB	DESC	
			5A	DD	001D0	PUSHL	R10	:
FD41	CF		03	FB	001D2	CALLS	#3, SET_NEW_DIR	:
	03		50	E9	001D7	BLBC	STATUS, -20\$:
	50		01	D0	001DA	MOVL	#1, R0	:
				04	001DD	RET		: 0947
								: 0948

: Routine Size: 478 bytes, Routine Base: \$CODE\$ + 0754


```

: 824 0949 1 GLOBAL ROUTINE MAIL$CMD_SET_NODIR =
: 825 0950 1 |**
: 826 0951 1 | FUNCTIONAL DESCRIPTION:
: 827 0952 1 |
: 828 0953 1 |         Cancel any SET MAIL_DIRECTORY setting. Cause the mail directory
: 829 0954 1 |         to be SYS$LOGIN:
: 830 0955 1 |
: 831 0956 1 | --
: 832 0957 2 BEGIN
: 833 0958 2
: 834 0959 2 LOCAL
: 835 0960 2     DESC : VECTOR[2,ONG],
: 836 0961 2     LOGIN_DESC : VECTOR[2,LONG],
: 837 0962 2     LOG_FLAG;
: 838 0963 2
: 839 0964 2 |
: 840 0965 2 |         If no mail directory set, then there is nothing to do
: 841 0966 2 |
: 842 0967 2 IF .MAIL$L_USRSYSBUF[VMD_B_DIRLNG] EQL 0
: 843 0968 2     THEN RETURN TRUE;
: 844 0969 2
: 845 0970 2 LOG_FLAG = CLIP$PRESENT(SD_LOG);
: 846 0971 2
: 847 0972 2 |
: 848 0973 2 |         Get uaf record and get descriptor for logindev:[logindir]
: 849 0974 2 |
: 850 P 0975 2 IF_ERR(MAIL$GET_USER(MAIL$Q_USERNAME);,
: 851 0976 2     RETURN TRUE);
: 852 0977 2
: 853 0978 2 DESC[0] = (.MAIL$T_UAF_BUFF[UAF$T_DEFDEV])<CS_COUNT>;
: 854 0979 2 DESC[1] = (.MAIL$T_OAF_BUFF[UAF$T_DEFDEV])<CS_STRING>;
: 855 0980 2 STR$COPY_DX(COMMAND_DESC,DESC);
: 856 0981 2 DESC[0] = (.MAIL$T_OAF_BUFF[UAF$T_DEFDIR])<CS_COUNT>;
: 857 0982 2 DESC[1] = (.MAIL$T_OAF_BUFF[UAF$T_DEFDIR])<CS_STRING>;
: 858 0983 2 STR$APPEND(COMMAND_DESC,DESC);
: 859 0984 2 DESC[0] = 0;
: 860 0985 2 DESC[1] = DESC[0];
: 861 0986 2
: 862 0987 2 |
: 863 0988 2 |         Rename all the mail files to the login directory.
: 864 0989 2 |         Note that if the login directory doesn't exist, this command won't create it.
: 865 0990 2 |
: 866 0991 2 SET_NEW_DIR(DESC,COMMAND_DESC,.LOG_FLAG);
: 867 0992 2
: 868 0993 2 RETURN TRUE
: 869 0994 1 END;

```

		003C 00000	.ENTRY MAIL\$CMD SET NODIR, Save R2,R3,R4,R5	: 0949
55	00000000G	00 9E 00002	MOVAB MAIL\$T_UAF_BUFF, R5	:
54	F6E1	CF 9E 00009	MOVAB SD_LOG, R4	:
53	00000000'	00 9E 0000E	MOVAB COMMAND_DESC, R3	:
5E		10 C2 00015	SUBL2 #16, SP	:
50	00000000G	00 D0 00018	MOVL MAIL\$L_USRSYSBUF, R0	: 0967

		41	A0	95	0001F	TSTB	65(R0)	
			64	13	00022	BEQL	1\$	
			54	DD	00024	PUSHL	R4	0970
00000000G	00		01	FB	00026	CALLS	#1, CLIS\$PRESENT	
	52		50	DD	0002D	MOVL	R0, LOG_FLAG	
		00000000G	00	9F	00030	PUSHAB	MAIL\$Q D\$SERNAME	0976
00000000G	00		01	FB	00036	CALLS	#1, MAIL\$GET_USER	
	48		50	E9	0003D	BLBC	STATUS, 1\$	
	50		65	DD	00040	MOVL	MAIL\$T UAF_BUFF, R0	0978
08	AE	74	A0	9A	00043	MOVZBL	116(R0), DESC	
0C	AE	75	A0	9E	00048	MOVAB	117(R0), DESC+4	0979
		08	AE	9F	0004D	PUSHAB	DESC	0980
			53	DD	00050	PUSHL	R3	
00000000G	00		02	FB	00052	CALLS	#2, STR\$COPY_DX	
	50		65	DD	00059	MOVL	MAIL\$T UAF_BUFF, R0	0981
08	AE	0094	C0	9A	0005C	MOVZBL	148(R0), DESC	
0C	AE	0095	C0	9E	00062	MOVAB	149(R0), DESC+4	0982
		08	AE	9F	00068	PUSHAB	DESC	0983
			53	DD	0006B	PUSHL	R3	
00000000G	00		02	FB	0006D	CALLS	#2, STR\$APPEND	
		08	AE	D4	00074	CLRL	DESC	0984
0C	AE	08	AE	9E	00077	MOVAB	DESC, DESC+4	0985
			52	DD	0007C	PUSHL	LOG_FLAG	0991
			53	DD	0007E	PUSHL	R3	
		10	AE	9F	00080	PUSHAB	DESC	
FCB2	CF		03	FB	00083	CALLS	#3, SET_NEW_DIR	
	50		01	DD	00088	MOVL	#1, R0	0993
			04	0008B	1\$:	RET		0994

: Routine Size: 140 bytes, Routine Base: \$CODE\$ + 0932

: 871 0995 1 END
: 872 0996 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	24	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODES	2494	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Symbols -----		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	46 0	1000	00:01.4
_\$255\$DUA28:[MAIL.OBJ]MAILDEF.L32;1	457	23 5	26	00:00.4

COMMAND QUALIFIERS

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

: Size: 2312 code + 206 data bytes
: Run Time: 00:26.1
: Elapsed Time: 01:36.7
: Lines/CPU Min: 2288
: Lexemes/CPU-Min: 27842
: Memory Used: 200 pages
: Compilation Complete

01	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100