```
MMM          MMM     AAAAAAAAA        IIIIIIIII     LLL
MMM          MMM     AAAAAAAAA        IIIIIIIII     LLL
MMM          MMM     AAAAAAAAA        IIIIIIIII     LLL
MMMMMM    MMMMMM  AAA           AAA       III      LLL
MMMMMM    MMMMMM  AAA           AAA       III      LLL
MMMMMM    MMMMMM  AAA           AAA       III      LLL
MMM   MMM   MMM  AAA           AAA       III      LLL
MMM   MMM   MMM  AAA           AAA       III      LLL
MMM   MMM   MMM  AAA           AAA       III      LLL
MMM         MMM  AAA           AAA       III      LLL
MMM         MMM  AAA           AAA       III      LLL
MMM         MMM  AAA           AAA       III      LLL
MMM         MMM  AAAAAAAAAAAAAAA         III      LLL
MMM         MMM  AAAAAAAAAAAAAAA         III      LLL
MMM         MMM  AAAAAAAAAAAAAAA         III      LLL
MMM         MMM  AAA           AAA       III      LLL
MMM         MMM  AAA           AAA       III      LLL
MMM         MMM  AAA           AAA       III      LLL
MMM         MMM  AAA           AAA    IIIIIIIII   LLLLLLLLLLLLLLL
MMM         MMM  AAA           AAA    IIIIIIIII   LLLLLLLLLLLLLLL
MMM         MMM  AAA           AAA    IIIIIIIII   LLLLLLLLLLLLLLL
```

```
 IIIIII      000000      SSSSSSSS  UU      UU  BBBBBBBB      SSSSSSSS
 IIIIII      000000      SSSSSSSS  UU      UU  BBBBBBBB      SSSSSSSS
   II      00      00  SS          UU      UU  BB      BB  SS
   II      00      00  SS          UU      UU  BB      BB  SS
   II      00      00  SS          UU      UU  BB      BB  SS
   II      00      00  SS          UU      UU  BB      BB  SS
   II      00      00    SSSSSS    UU      UU  BBBBBBBB      SSSSSS
   II      00      00    SSSSSS    UU      UU  BBBBBBBB      SSSSSS
   II      00      00          SS  UU      UU  BB      BB          SS
   II      00      00          SS  UU      UU  BB      BB          SS
   II      00      00          SS  UU      UU  BB      BB          SS      ....
   II      00      00          SS  UU      UU  BB      BB          SS      ....
 IIIIII      000000    SSSSSSSS    UUUUUUUUUU  BBBBBB6B    SSSSSSSS      ....
 IIIIII      000000    SSSSSSSS    UUUUUUUUUU  BBBBBBBB    SSSSSSSS      ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
0000     1          .TITLE  MAIL$IOSUBS - INPUT/OUTPUT SUBROUTINES
0000     2          .IDENT  'V04-000'
0000     3 ;
0000     4 ;*****************************************************************************
0000     5 ;*                                                                           *
0000     6 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 B                                   *
0000     7 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     8 ;*  ALL RIGHTS RESERVED.                                                     *
0000     9 ;*                                                                           *
0000    10 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    11 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    12 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    13 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    14 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    15 ;*  TRANSFERRED.                                                             *
0000    16 ;*                                                                           *
0000    17 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    18 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    19 ;*  CORPORATION.                                                             *
0000    20 ;*                                                                           *
0000    21 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    22 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    23 ;*                                                                           *
0000    24 ;*                                                                           *
0000    25 ;*****************************************************************************
0000    26 ;
0000    27
0000    28 ;++
0000    29 ; FACILITY:     VAX/VMS MAIL UTILITY
0000    30 ;
0000    31 ; ABSTRACT:     INPUT AND OUTPUT SUBROUTINES.
0000    32 ;
0000    33 ; ENVIRONMENT:  NATIVE/USER MODE
0000    34 ;
0000    35 ; AUTHOR:       LEN KAWELL, CREATION DATE: 2-FEB-79
0000    36 ;
0000    37 ; MODIFICATION HISTORY:
0000    38 ;
0000    39 ;       V03-008 BLS0255         Benn Schreiber          28-Dec-1983
0000    40 ;               Correct upcase_q, use new global flags
0000    41 ;
0000    42 ;       V03-007 BLS0246         Benn Schreiber           8-Nov-1983
0000    43 ;               Add MAIL$IOSUB_INIT to remove statically allocated buffers
0000    44 ;
0000    45 ;       V03-006 BLS0229         Benn Schreiber          16-Jul-1983
0000    46 ;               Remove routines rewritten into BLISS and fold in what
00C0    47 ;               is left of NETSEND.MAR
0000    48 ;
0000    49 ;       V03-005 BLS0220         Benn Schreiber          30-Apr-1983
0000    50 ;               Keyapd input routine support
0000    51 ;
0000    52 ;       V03-004 BLS0214         Benn Schreiber          27-Mar-1983
0000    53 ;               Add MAIL$FAO_GETMSG to GETMSG a message and then
0000    54 ;               apply fao to it.
0000    55 ;
0000    56 ;       V03-003 BLS0211         Benn Schreiber          13-Mar-1983
0000    57 ;               Rework alternate net support for more flexibility.  Handle
```

```
0000     58 ;                  CTRL/C as EOF in MAIL$GET_INPUT
0000     59 ;
0000     60 ;       V03-002 BLS0194         Benn Schreiber          19-Nov-1982
0000     61 ;                Add support for alternate incoming network access
0000     62 ;
0000     63 ;       V03-001 BLS0189         Benn Schreiber          15-Oct-1982
0000     64 ;                Add routine to upcase except in double quote pairs
0000     65 ;                Add routine to do wildcard matching
0000     66 ;--
```

```
                     0000     68          .SBTTL  DECLARATIONS
                     0000     69 ;
                     0000     70 ; INCLUDE FILES:
                     0000     71 ;
                     0000     72
                     0000     73 ;
                     0000     74 ; EQUATED SYMBOLS:
                     0000     75 ;
00000004             0000     76 BUFF_DESC       = 4                         ;BUFFER DESCRIPTOR ARGUMENT
00000008             0000     77 FAOARG1         = 8                         ;FIRST FAO ARGUMENT ARGUMENT
                     0000     78
00000200             0000     79 MAIL$K_INBUFFSZ = 512                       ;INPUT BUFFER SIZE
                     0000     80
                     0000     81          $STSDEF                            ;STATUS MESSAGE DEFINITIONS
                     0000     82          $SHRDEF                            ;SHARED MESSAGE DEFINITIONS
                     0000     83          CNCTDEF                            ;CNCT BLOCK DEFINITIONS
                     0000     84          LNKDEF                             ;LOGICAL LINK DEFINITIONS
                     0000     85          MAIFDEF                            ;MAIL CONTROL FLAG DEFINITIONS
                     0000     86 ;
                     0000     87 ; LOCAL MACROS:
                     0000     88 ;
                     0000     89
                     0000     90 ;
                     0000     91 ; OWN STORAGE:
                     0000     92 ;
                     0000     93
                00000000     94          .PSECT  $OWN$,LONG,RD,WRT,NOEXE
                     0000     95
000000FF             0000     96 MSGDESC:.LONG    255                        ;EVENTUALLY A DESCRIPTOR (ORDER ASSUMED)
00000000             0004     97 MSGBUF: .LONG    0
00000200             0008     98 FAODESC:.LONG    512                        ;AND ANOTHER DESCRIPTOR
00000000             000C     99 FAOBUF: .LONG    0
                     0010    100
                00000000     101          .PSECT  $CODE$,LONG,RD,NOWRT,EXE
```

```
                                                I 2
                     0000   103          .SBTTL  MAIL$IOSUB_INIT - ALLOCATE SOME BUFFERS
                     0000   104 ;++
                     0000   105 ; FUNCTIONAL DESCRIPTION:
                     0000   106 ;
                     0000   107 ;       ALLOCATE BUFFERS FOR ROUTINES IN THIS MODULE
                     0000   108 ;
                     0000   109 ;--
                     0000   110
                001C 0000   111          .ENTRY  MAIL$IOSUB_INIT,^M<R2,R3,R4>
    7E   00FF 8F      3C    0002   112          MOVZWL  #255,-(SP)              ;SIZE OF BUFFER TO ALLOCATE
     00000004'EF      9F    0007   113          PUSHAB  MSGBUF                  ;ADDRESS OF LONGWORD TO STORE ADDR IN
           04 AE      9F    000D   114          PUSHAB  4(SP)                   ;ADDR OF SIZE OF BUFFER
00000000'GF    02     FB    0010   115          CALLS   #2,G^LIB$GET_VM         ;ALLOCATE IT
           18 50      E9    0017   116          BLBC    R0,20$                  ;BRANCH IF ERROR
    6E   0200 8F      3C    001A   117          MOVZWL  #512,(SP)               ;ALLOCATE FAO BUFFER
     0000000C'EF      9F    001F   118          PUSHAB  FAOBUF                  ;ADDR TO STORE BUFFER ADDR INTO
           04 AE      9F    0025   119          PUSHAB  4(SP)                   ;ADDR OF SIZE OF BUFFER
00000000'GF    02     FB    0028   120          CALLS   #2,G^LIB$GET_VM         ;ALLOCATE
           09 50      E8    002F   121          BLBS    R0,40$                  ;BRANCH IF ALL OK
              50      DD    0032   122 20$:     PUSHL   R0                      ;SIGNAL ERROR
00000000'GF    01     FB    0034   123          CALLS   #1,G^LIB$SIGNAL
                04     003B   124 40$:     RET
                       003C   125
```

```
                          003C   127          .SBTTL  MAIL$CONFIRM - PROMPT FOR CONFIRMATION
                          003C   128 ;++
                          003C   129 ; FUNCTIONAL DESCRIPTION:
                          003C   130 ;
                          003C   131 ;       IF THIS IS NOT A NETWORK JOB, THEN PROMPT FOR A YES OR NO
                          003C   132 ;       RESPONSE AND CHECK IF YES OR NO.
                          003C   133 ;
                          003C   134 ; CALLING SEQUENCE:
                          003C   135 ;       CALL MAIL$CONFIRM(BUFF_DESC,PROMPT_DESC)
                          003C   136 ;
                          003C   137 ; INPUT PARAMETERS:
                          003C   138 ;       BUFF_DESC(AP) = ADDR OF BUFFER DESCRIPTOR
                          003C   139 ;       PROMPT_DESC(AP) = ADDR OF PROMPT DESCRIPTOR
                          003C   140 ;
                          003C   141 ; IMPLICIT INPUTS:
                          003C   142 ;       MAIL$GL_FLAGS[MAIF_V_NETJOB] = 1 IF THIS IS A NETWORK JOB.
                          003C   143 ;
                          003C   144 ; OUTPUT PARAMETERS:
                          003C   145 ;       NONE
                          003C   146 ;
                          003C   147 ; IMPLICIT OUTPUTS:
                          003C   148 ;       NONE
                          003C   149 ;
                          003C   150 ; COMPLETION CODES:
                          003C   151 ;       R0 = SUCCESS IF RESPONSE WAS A "Y",
                          003C   152 ;          = ERROR IF NO RESPONSE, RESPONSE WAS NOT A "Y", OR JOB IS
                          003C   153 ;            A NETWORK JOB.
                          003C   154 ;
                          003C   155 ; SIDE EFFECTS:
                          003C   156 ;       PROMPT DISPLAYED AND INPUT OBTAINED FROM INPUT DEVICE.
                          003C   157 ;
                          003C   158 ;--
                    0004  003C   159          .ENTRY  MAIL$CONFIRM,-          ;PROMPT AND GET RESPONSE
                          003E   160                  ^M<R2>                  ;(ENTRY MASK)
2D 00000000'EF    01    E0  003E   161          BBS     #MAIF_V_NETJOB,MAIL$GL_FLAGS,20$ ;BR IF NETWORK JOB
             08   AC    DD  0046   162          PUSHL   8(AP)                   ;STACK PROMPT DESCRIPTOR ADDRESS
             04   AC    DD  0049   163          PUSHL   4(AP)                   ;STACK GET-STRING
       00000000'EF    9F  004C   164          PUSHAB  MAIL$L_SMG_KEYBOARD     ;AND KEYBOARD ID
   00000000'GF      03    FB  0052   165          CALLS   #3,G^SMG$READ_STRING    ;READ FROM SYS$INPUT
         52     04    AC  D0  0059   166          MOVL    BUFF_DESC(AP),R2        ;GET ADDRESS OF BUFFER DESC
                      62    B5  005D   167          TSTW    (R2)                    ;NO RESPONSE?
                      12    13  005F   168          BEQL    20$                     ;BR IF YES - NO RESPONSE = "NO"
      04 B2     59    8F    91  0061   169          CMPB    #^A"Y",@4(R2)           ;WAS RESPONSE "YES"?
                      07    13  0064   170          BEQL    10$                     ;BR IF YES
      04 B2     79    8F    91  0066   171          CMPB    #^A"y",@4(R2)           ;WAS RESPONSE "yes"?
                      04    12  006D   172          BNEQ    20$                     ;BR IF NO
                            006F   173 10$:
            50    01    D0  006F   174          MOVL    #1,R0                   ;SET "YES" RESPONSE
                      04    0072   175          RET                             ;
                            0073   176 20$:
                  50    D4  0073   177          CLRL    R0                      ;SET "NO" RESPONSE
                      04    0075   178          RET                             ;
                            0076   179
```

```
                                0076   181          .SBTTL  MAIL$UPCASE - CONVERT A STRING TO UPPERCASE
                                0076   182 ;++
                                0076   183 ;
                                0076   184 ; FUNCTIONAL DESCRIPTION:
                                0076   185 ;
                                0076   186 ;       THIS ROUTINE IS CALLED TO CONVERT A STRING TO UPPER CASE CHARACTERS.
                                0076   187 ;
                                0076   188 ; INPUTS:
                                0076   189 ;
                                0076   190 ;       BUFF_DESC = ADDRESS OF STRING DESCRIPTOR.
                                0076   191 ;
                                0076   192 ; OUTPUTS:
                                0076   193 ;
                                0076   194 ;       STRING CONVERTED TO UPPERCASE.
                                0076   195 ;--
                          0004  0076   196          .ENTRY  MAIL$UPCASE,-                   ;CONVERT TO UPPER CASE
                                0078   197                  ^M<R2>                          ;(ENTRY MASK)
          50    04 AC     D0    0078   198          MOVL    BUFF_DESC(AP),R0                ;GET ADDRESS OF DESCRIPTOR
             51    60     3C    007C   199          MOVZWL  (R0),R1                         ;GET SIZE OF STRING
                   1B     13    007F   200          BEQL    30$                             ;EXIT IF ZERO
          50    04 A0     D0    0081   201          MOVL    4(R0),R0                        ;GET ADDRESS OF STRING
             52    80     90    0085   202 10$:     MOVB    (R0)+,R2                        ;GET NEXT CHARACTER
          61 8F    52     91    0088   203          CMPB    R2,#^A"a"                       ;LOWER CASE LETTER?
                   0B     1F    008C   204          BLSSU   20$                             ;IF LSSU NO
          7A 8F    52     91    008E   205          CMPB    R2,#^A"z"                       ;LOWER CASE LETTER?
                   05     1A    0092   206          BGTRU   20$                             ;IF GTRU NO
       FF A0    52    20     83    0094   207          SUBB3   #32,R2,-1(R0)                   ;CONVERT TO UPPERCASE LETTER
                E9 51     F5    0099   208 20$:     SOBGTR  R1,10$                          ;DECREMENT COUNT AND LOOP
                          04    009C   209 30$:     RET                                     ;
```

```
                                009D   211          .SBTTL  MAIL$UPCASE_Q - CONVERT A STRING TO UPPERCASE HANDLING QUOTES
                                009D   212 ;++
                                009D   213 ;
                                009D   214 ; FUNCTIONAL DESCRIPTION:
                                009D   215 ;
                                009D   216 ;       THIS ROUTINE IS CALLED TO CONVERT A STRING TO UPPER CASE CHARACTERS.
                                009D   217 ;       CHARACTERS INSIDE DOUBLE QUOTES ARE NOT CONVERTED
                                009D   218 ;
                                009D   219 ; INPUTS:
                                009D   220 ;
                                009D   221 ;       BUFF_DESC = ADDRESS OF STRING DESCRIPTOR.
                                009D   222 ;
                                009D   223 ; OUTPUTS:
                                009D   224 ;
                                009D   225 ;       STRING CONVERTED TO UPPERCASE.
                                009D   226 ;--
                         000C   009D   227          .ENTRY  MAIL$UPCASE_Q,-                 ;CONVERT TO UPPER CASE
                                009F   228                  ^M<R2,R3>                       ;(ENTRY MASK)
50   04 AC    D0            009F   229          MOVL    BUFF_DESC(AP),R0                    ;GET ADDRESS OF DESCRIPTOR
   51   60    3C            00A3   230          MOVZWL  (R0),R1                             ;GET SIZE OF STRING
        32    13            00A6   231          BEQL    30$                                 ;EXIT IF ZERO
53     FF 8F    98          00A8   232          CVTBL   #-1,R3                              ;ASSUME NOT IN DOUBLE QUOTES
50   04 A0    D0            00AC   233          MOVL    4(R0),R0                            ;GET ADDRESS OF STRING
   52   80    90            00B0   234 10$:     MOVB    (R0)+,R2                            ;GET NEXT CHARACTER
   22   52    91            00B3   235          CMPB    R2,#^A/"/                           ;DOUBLE QUOTE CHARACTER?
        0A    12            00B6   236          BNEQ    15$                                 ;IF NEQ NO
        53    D6            00B8   237          INCL    R3                                  ;YES--INCREMENT LEVEL
        1B    13            00BA   238          BEQL    20$                                 ;IF EQL FIRST QUOTE
53     FF 8F    98          00BC   239          CVTBL   #-1,R3                              ;RESET QUOTE LEVEL
        15    11            00C0   240          BRB     20$                                 ;CONTINUE
        53    D5            00C2   241 15$:     TSTL    R3                                  ;INSIDE QUOTES?
        11    18            00C4   242          BGEQ    20$                                 ;IF GEQ YES
61 8F    52    91            00C6   243          CMPB    R2,#^A"a"                           ;LOWER CASE LETTER?
        0B    1F            00CA   244          BLSSU   20$                                 ;IF LSSU NO
7A 8F    52    91            00CC   245          CMPB    R2,#^A"z"                           ;LOWER CASE LETTER?
        05    1A            00D0   246          BGTRU   20$                                 ;IF GTRU NO
FF A0    52    20            00D2   247          SUBB3   #32,R2,-1(R0)                       ;CONVERT TO UPPERCASE LETTER
      D6 51    F5            00D7   248 20$:     SOBGTR  R1,10$                              ;DECREMENT COUNT AND LOOP
               04            00DA   249 30$:     RET                                         ;
```

```
                                00DB   251          .SBTTL  MAIL$GETMSG - Find message text for code
                                00DB   252 ;++
                                00DB   253 ; Functional description:
                                00DB   254 ;
                                00DB   255 ;       return descriptor for message code
                                00DB   256 ;
                                00DB   257 ; inputs:
                                00DB   258 ;
                                00DB   259 ;       4(ap) = value of code to lookup
                                00DB   260 ;
                                00DB   261 ; routine value:
                                00DB   262 ;
                                00DB   263 ;       r0 = address of descriptor of message text
                                00DB   264 ;
                                00DB   265 ;--
                          0004  00DB   266          .ENTRY  MAIL$GETMSG,^M<R2>
                                00DD   267
     52   00000000'EF    9E    00DD   268          MOVAB   L^MSGDESC,R2
          62    0200 8F    3C  00E4   269          MOVZWL  #512,(R2)
                                00E9   270          $GETMSG_S MSGID=4(AP),-
                                00E9   271                    MSGLEN=MSGDESC,-
                                00E9   272                    BUFADR=MSGDESC,-
                                00E9   273                    FLAGS=#1
               50   52    D0  0103   274          MOVL    R2,R0
                          04    0106   275          RET
```

```
                                0107   277          .SBTTL  MAIL$FAO_GETMSG, GETMSG and then FAO message
                                0107   278 ;++
                                0107   279 ; FUNCTIONAL DESCRIPTION:
                                0107   280 ;
                                0107   281 ;       This routine performs a $GETMSG on a message, then does a $FAO,
                                0107   282 ;       and returns the address of the string descriptor of the result.
                                0107   283 ;
                                0107   284 ; Inputs:
                                0107   285 ;
                                0107   286 ;       4(AP) = Value of message code
                                0107   287 ;       8(AP) = Start of $fao arglist
                                0107   288 ;
                                0107   289 ;--
                          0004  0107   290          .ENTRY  MAIL$FAO_GETMSG,^M<R2>
                    04 AC       DD   0109   291          PUSHL   4(AP)
                FFCA CF     01      FB   010C   292          CALLS   #1,W^MAIL$GETMSG
            52   00000008'EF        9E   0111   293          MOVAB   FAODESC,R2              ;GET ADDRESS OF BUFFER DESCRIPTOR
                62    0200 8F       3C   0118   294          MOVZWL  #512,(R2)               ;RESET DESCRIPTOR
                      51   50       D0   011D   295          MOVL    R0,R1                   ;GET ADDRESS OF CONTROL STRING DESCR
                                         0120   296          $FAOL_S CTRSTR=(R1),-           ;DO THE FAO
                                         0120   297                  OUTLEN=(R2),-
                                         0120   298                  OUTBUF=(R2),-
                                         0120   299                  PRMLST=8(AP)
                      03 50       E9   0130   300          BLBC    R0,10$                  ;IF THERE IS AN ERROR, WILL ACCVIO
                      50   52       D0   0133   301          MOVL    R2,R0                   ; EVENTUALLY
                                   04   0136   302 10$:     RET
```

```
                        0137   304          .SBTTL  MAIL$MATCH_NAME, general wild card matching
                        0137   305 ;++
                        0137   306 ; Functional Description:
                        0137   307 ;       This routine performs the general embedded wild card matching
                        0137   308 ;       algorithm.
                        0137   309 ;
                        0137   310 ; Calling Sequence:
                        0137   311 ;       ret_status.wlc.v = STR$MATCH_NAME (CAND.rt.dx,PATRN.rt.dx)
                        0137   312 ;
                        0137   313 ; Formal Parameters:
                        0137   314 ;       CAND.rt.dx      Address of string descriptor for candidate string
                        0137   315 ;                        (The current item being looked at)
                        0137   316 ;       PATRN.rt.dx     Address of string descriptor for pattern string
                        0137   317 ;                        (The item looking for)
                        0137   318 ;
                        0137   319 ; Implicit Inputs:
                        0137   320 ;       none
                        0137   321 ;
                        0137   322 ; Output Parameters:
                        0137   323 ;       none
                        0137   324 ;
                        0137   325 ; Implicit Outputs:
                        0137   326 ;       none
                        0137   327 ;
                        0137   328 ; Routines Called:
                        0137   329 ;       STR$ANALYZE_SDESC_R1
                        0137   330 ;
                        0137   331 ; Routine Value:
                        0137   332 ;       STR$_MATCH   if the strings match.
                        0137   333 ;       STR$_NOMATCH if the strings don't match
                        0137   334 ;
                        0137   335 ; Signals:
                        0137   336 ;       Errors from STR$ANALYZE_SDESC
                        0137   337 ;
                        0137   338 ; Side Effects:
                        0137   339 ;       none
                        0137   340 ;
                        0137   341 ;--
                00000001 0137   342          STR$_MATCH = 1
                00000000 0137   343          STR$_NOMATCH = 0
                        0137   344
                    03FC 0137   345          .ENTRY  mail$match_name,^M<R2,R3,R4,R5,R6,R7,R8,R9>
                        0139   346
     50    04 AC    D0  0139   347          MOVL    4(AP),R0                ; get first descriptor address
00000000'GF    16  013D   348          JSB     G^STR$ANALYZE_SDESC_R1  ; extract string length and address
     7E    50    7D  0143   349          MOVQ    R0,-(SP)                ; save descriptor
     50    08 AC    D0  0146   350          MOVL    8(AP),R0                ; get second descriptor address
00000000'GF    16  014A   351          JSB     G^STR$ANALYZE_SDESC_R1  ; analyze second descriptor
     54    50    7D  0150   352          MOVQ    R0,R4                   ; set up for match algorithm
     52    8E    7D  0153   353          MOVQ    (SP)+,R2                ; retrieve first descriptor
     50    00    D0  0156   354          MOVL    #STR$_NOMATCH,R0        ; Assume failure
           56    D4  0159   355          CLRL    R6                      ; Clear saved candidate count
                        015B   356 ;
                        015B   357 ; Main scanning loop.
                        015B   358 ;
           54    D7  015B   359 10$:     DECL    R4                      ; Pattern exhausted?
           24    19  015D   360          BLSS    30$                     ; Branch if yes
```

```
                51   85   9A  015F   361          MOVZBL  (R5)+,R1                ; Get next character in pattern
                2A   51   91  0162   362          CMPB    R1,#^A'*'               ; Pattern specifies wild string?
                     24   13  0165   363          BEQL    60$                     ; Branch if yes
                     52   D7  0167   364          DECL    R2                      ; Candidate exhausted?
                     1F   19  0169   365          BLSS    50$                     ; Branch if yes
                83   51   91  016B   366          CMPB    R1,(R3)+                ; Compare pattern to candidate
                     EB   13  016E   367          BEQL    10$                     ; Branch if pattern equals candidate
                25   51   91  0170   368          CMPB    R1,#^A'%'               ; Pattern specifies wild character?
                     E6   13  0173   369          BEQL    10$                     ; Branch if yes
                              0175   370 ;
                              0175   371 ; We have detected a mismatch, or we are out of pattern while there is
                              0175   372 ; candidate left.  Back up to the last '*', advance a candidate character,
                              0175   373 ; and try again.
                              0175   374 ;
                     56   D7  0175   375 20$:     DECL    R6                      ; Count a saved candidate character
                     11   19  0177   376          BLSS    50$                     ; Branch if no saved candidate
                     57   D6  0179   377          INCL    R7                      ; Set to try next character
                52   56   7D  017B   378          MOVQ    R6,R2                   ; Restore descriptors to backup point
                54   58   7D  017E   379          MOVQ    R8,R4                   ;
                     D8   11  0181   380          BRB     10$                     ; Continue testing
                              0183   381 ;
                              0183   382 ; Here when pattern is exhausted.
                              0183   383 ;
                     52   D5  0183   384 30$:     TSTL    R2                      ; Candidate exhausted?
                     EE   12  0185   385          BNEQ    20$                     ; Branch if no
                              0187   386 ;
                              0187   387 ; Here to return.
                              0187   388 ;
                50   01   D0  0187   389 40$:     MOVL    #STR$_MATCH,R0          ; Set success return
                          04  018A   390 50$:     RET                             ; Return
                              018B   391 ;
                              018B   392 ; We have detected a '*' in the pattern.  Save the pointers for backtracking.
                              018B   393 ;
                     54   D5  018B   394 60$:     TSTL    R4                      ; Pattern null after '*'?
                     F8   13  018D   395          BEQL    40$                     ; Branch if yes
                56   52   7D  018F   396          MOVQ    R2,R6                   ; Save descriptors of both strings
                58   54   7D  0192   397          MOVQ    R4,R8                   ;
                     C4   11  0195   398          BRB     10$                     ; Continue testing
```

```
                        0197   400          .SBTTL  MAIL$READ_ERROR_TEXT
                        0197   401 ;
                        0197   402 ; FUNCTIONAL DESCRIPTION:
                        0197   403 ;
                        0197   404 ;       READ THE ERROR MESSAGE TEXT FROM THE REMOTE NODE AND SIGNAL
                        0197   405 ;       THEM ALL AT ONCE.
                        0197   406 ;
                        0197   407 ; INPUTS:
                        0197   408 ;
                        0197   409 ;       4(AP) CONTEXT BLOCK ADDRESS
                        0197   410 ;       8(AP) ADDR OF ROUTINE TO CALL TO READ NEXT RECORD
                        0197   411 ;
                        0197   412 ;       QUITS READING ON BYTE OF 0
                        0197   413 ;
                        0197   414 ;
                        0197   415 ; GET STATUS MESSAGE(S) FROM SLAVE AND SIGNAL
                        0197   416 ;
                001C    0197   417          .ENTRY MAIL$READ_ERROR_TEXT,^M<R2,R3,R4>
            53    D4    0199   418          CLRL    R3                          ;SET END-OF-MESSAGES MARKER
                        019B   419 20$:
      5E    FE00 CE     019B   420          MOVAL   -MAIL$K_INBUFFSZ(SP),SP ;CREATE MESSAGE BUFFER
            53    DD    01A0   421          PUSHL   R3                          ;STACK ADDR OF PREVIOUS MSG DESC
              04  AE    01A2   422          PUSHAB  4(SP)                       ;CREATE DESCRIPTOR OF THIS MSG DESC
     00000200     8F    01A5   423          PUSHL   #MAIL$K_INBUFFSZ            ;...
            54    5E    01AB   424          MOVL    SP,R4                       ;SAVE ADDR OF THIS ONE
            54    DD    01AE   425          PUSHL   R4                          ;STACK DESCRIPTOR ADDRESS
              04  AC    01B0   426          PUSHL   4(AP)                       ;STACK CONTEXT ADDRESS
        08    BC    02    01B3   427          CALLS   #2,@8(AP)                   ;CALL GET-ROUTINE
              0A  50    E9    01B7   428          BLBC    R0,30$                  ;TREAT ERROR AS END OF MESSAGE
              04  B4    95    01BA   429 27$:     TSTB    @4(R4)                  ;END OF MESSAGES?
                  05    13    01BD   430          BEQL    30$                     ;BR IF YES - SIGNAL THEM
            53    54    D0    01BF   431          MOVL    R4,R3                   ;SET NEW PREVIOUS DESC ADDR
                  D7    11    01C2   432          BRB     20$                     ;GET NEXT MESSAGE
                        01C4   433 30$:
                  54    D4    01C4   434          CLRL    R4                      ;INIT SIGNAL ARGUMENT COUNT
                        01C6   435 40$:
                  53    DD    01C6   436          PUSHL   R3                      ;SET ADDR OF MESSAGE DESC
                  1C    13    01C8   437          BEQL    60$                     ;BRANCH IF NONE
                7E    01    B0    01CA   438          MOVW    #1,-(SP)            ;SET MESSAGE FLAGS (NO %MAIL-F-TEXT,)
                7E    01    B0    01CD   439          MOVW    #1,-(SP)            ;SET FAO COUNT
     00001130'  8F    DD    01D0   440          PUSHL   #SHR$_TEXT!-                ;SET MESSAGE NAME
                        01D6   441                  <MAIL$_FACILITY @ STS$V_FAC_NO>
            54    03    C0    01D6   442          ADDL    #3,R4                   ;INCREMENT SIGNAL ARGUMENT COUNT
          53    08    A3    D0    01D9   443          MOVL    8(R3),R3            ;GET ADDR OF NEXT DESC
                  E7    12    01DD   444          BNEQ    40$                     ;BR IF THERE IS ONE
00000000'GF       54    FB    01DF   445          CALLS   R4,G^LIB$SIGNAL         ;SIGNAL THE MESSAGES
                  04    01E6   446 60$:     RET                                 ;DONE
                        01E7   447
                        01E7   448
                        01E7   449
```

```
                                01E7   451          .SBTTL  MAIL$INCLINKADR - INCREMENT LINK ADDRESSEE COUNT
                                01E7   452 ;++
                                01E7   453 ; FUNCTIONAL DESCRIPTION:
                                01E7   454 ;
                                01E7   455 ;       INCREMENT THE ADDRESSEE COUNT ASSOCIATED WITH A LINK
                                01E7   456 ;
                                01E7   457 ; INPUTS:
                                01E7   458 ;
                                01E7   459 ;       4(AP)   ADDRESS OF LINK BLOCK
                                01E7   460 ;
                                01E7   461 ; OUTPUTS:
                                01E7   462 ;
                                01E7   463 ;       LNK_L_ADRCNT IN LINK BLOCK INCREMENTED IF LINK BLOCK ADDRESS
                                01E7   464 ;       NON=0
                                01E7   465 ;
                                01E7   466 ;--
                          0004  01E7   467          .ENTRY  MAIL$INCLINKADR,^M<R2>
                                01E9   468
     52    04 AC      D0      01EB   469          MOVL    4(AP),R2                ;GET LINK BLOCK ADDRESS
                08      13      01ED   470          BEQL    10$                     ;IF EQL ALL DONE
03 2E A2      01      E0      01EF   471          BBS     #LNK_V_DEAD,LNK_B_FLAGS(R2),10$ ;DON'T INCREMENT IT IF DEAD
          08 A2       D6      01F4   472          INCL    LNK_L_ADRCNT(R2)        ;INCREMENT THE COUNT
                      04      01F7   473 10$:     RET
```

```
                     01F8   475          .SBTTL  MAIL$DECLINKADR - DECREMENT LINK ADDRESSEE COUNT
                     01F8   476
                     01F8   477 ;++
                     01F8   478 ; FUNCTIONAL DESCRIPTION:
                     01F8   479 ;
                     01F8   480 ;       DECREMENTS THE ADDRESSEE COUNT ASSOCIATED WITH A LINK
                     01F8   481 ;
                     01F8   482 ; INPUTS:
                     01F8   483 ;
                     01F8   484 ;       4(AP)   ADDRESS OF THE LINK BLOCK
                     01F8   485 ;
                     01F8   486 ;--
                     01F8   487
                0004 01F8   488          .ENTRY  MAIL$DECLINKADR,^M<R2>
52   04 AC     D0   01FA   489          MOVL    4(AP),R2                ;GET ADDRESS OF LINK BLOCK
        08      13   01FE   490          BEQL    10$                     ;IF 0 THEN NONE
     08 A2      D5   0200   491          TSTL    LNK_L_ADRCNT(R2)        ;ALREADY 0?
        03      13   0203   492          BEQL    10$
     08 A2      D7   0205   493          DECL    LNK_L_ADRCNT(R2)        ;DECREMENT IT'S ADDRESEE COUNT
                04   0208   494 10$:     RET
```

```
                     0209   496          .SBTTL  MAIL$DELEMPLINK - DELETE ALL LINKS WITH NO ADDRESSEES
                     0209   497 ;++
                     0209   498 ; FUNCTIONAL DESCRIPTION
                     0209   499 ;
                     0209   500 ;       DELETE ALL LOGICAL LINKS WHICH HAVE NO ADDRESSEES ASSOCIATED
                     0209   501 ;       WITH THEM
                     0209   502 ;
                     0209   503 ; INPUTS:
                     0209   504 ;
                     0209   505 ;       THE LINK LIST
                     0209   506 ;
                     0209   507 ;--
                     0209   508
                001C 0209   509          .ENTRY  MAIL$DELEMPLINK,^M<R2,R3,R4>
                     020B   510
52   04 AC    D0     020B   511          MOVL    4(AP),R2                ;START AT THE HEAD OF THE LIST
     54   52  D0     020F   512          MOVL    R2,R4                   ;SAVE LISTHEAD ADDRESS
     52   62  D0     0212   513 10$:     MOVL    LNK_L_FLINK(R2),R2      ;GET NEXT LINK BLOCK
     54   52  D1     0215   514 20$:     CMPL    R2,R4                   ;DONE WITH LIST?
          12  13     0218   515          BEQL    30$                     ;IF EQL YES
        08 A2  D5     021A   516          TSTL    LNK_L_ADRCNT(R2)        ;NO--ANY ADDRESSEES THIS NODE?
          F3  12     021D   517          BNEQ    10$                     ;IF NEQ YES
     53   62  D0     021F   518          MOVL    LNK_L_FLINK(R2),R3      ;NO--GET NEXT IN LIST
     52   62  0F     0222   519          REMQUE  (R2),R2                 ;REMOVE THIS LINK FROM THE QUEUE
          17  10     0225   520          BSBB    DEALLOC_LNK             ;DELETE THE LINK AND MEMORY
     52   53  D0     0227   521          MOVL    R3,R2                   ;CONTINUE SCANNING
          E9  11     022A   522          BRB     20$                     ;CONTINUE
              04     022C   523 30$:     RET
```

```
                                022D   525          .SBTTL  MAIL$DELETELINK - DELETE ALL THE LOGICAL LINKS
                                022D   526 ;++
                                022D   527 ; FUNCTIONAL DESCRIPTION:
                                022D   528 ;
                                022D   529 ;       DELETE ALL THE LOGICAL LINKS IN THE LOGICAL LINK LIST.
                                022D   530 ;
                                022D   531 ; CALLING SEQUENCE:
                                022D   532 ;       CALL MAIL$DELETELINK(LISTHEAD)
                                022D   533 ;
                                022D   534 ; INPUT PARAMETERS:
                                022D   535 ;       NONE
                                022D   536 ;
                                022D   537 ; IMPLICIT INPUTS:
                                022D   538 ;
                                022D   539 ; OUTPUT PARAMETERS:
                                022D   540 ;       NONE
                                022D   541 ;
                                022D   542 ; IMPLICIT OUTPUTS:
                                022D   543 ;       NONE
                                022D   544 ;
                                022D   545 ; COMPLETION CODES:
                                022D   546 ;       NONE
                                022D   547 ;
                                022D   548 ; SIDE EFFECTS:
                                022D   549 ;       ALL LOGICAL LINKS ARE DELETED AND SLAVE MAIL JOBS ARE STOPPED.
                                022D   550 ;
                                022D   551 ;--
                        000C    022D   552          .ENTRY  MAIL$DELETELINK,-       ;DELETE ALL THE LOGICAL LINKS
                                022F   553                  ^M<R2,R3>               ;(ENTRY MASK)
                53   04 AC   D0 022F   554          MOVL    4(AP),R3
                                0233   555 10$:
                52   00 B3   0F 0233   556          REMQUE  @(R3),R2                ;REMOVE NEXT ENTRY
                        04      1D 0237   557          BVS     20$                     ;BR IF NONE - ALL DONE
                        03      10 0239   558          BSBB    DEALLOC_LNK             ;DEALLOCATE THE LINK BLOCK
                        F6      11 023B   559          BRB     10$                     ;CHECK FOR ANOTHER
                                023D   560 20$:
                                04 023D   561          RET                             ;RETURN
                                023E   562
                                023E   563 ;
                                023E   564 ; DEALLOCATE THE LINK BLOCK AFTER DELETING THE LOGICAL LINK
                                023E   565 ;
                                023E   566 ; LINK BLOCK ADDRESS IN R2
                                023E   567 ;
                                023E   568 DEALLOC_LNK:
             21 2E A2   02      E0 023E   569          BBS     #LNK_V_ALTP,LNK_B_FLAGS(R2),20$ ;BR IF ALTERNATE PROTOCOL
                                0243   570          $DASSGN_S CHAN=LNK_W_CHAN(R2)   ;DELETE THE LOGICAL LINK
                50   4F A2   9A 024E   571 10$:    MOVZBL  LNK_B_PNLEN(R2),R0      ;GET LENGTH OF PROTOCOL NAME
                        50 A0   9F 0252   572          PUSHAB  LNK_C_LENGTH(R0)        ;SET SIZE OF ENTRY TO DEALLOCATE
                           52   DD 0255   573          PUSHL   R2                      ;SET ADDR OF ENTRY
                           6E   DF 0257   574          PUSHAL  (SP)                    ;SET ADDR OF ADDR OF ENTRY
                        08 AE   DF 0259   575          PUSHAL  8(SP)                   ;SET ADDR OF SIZE OF ENTRY
00000000'GF             04      FB 025C   576          CALLS   #4,G^LIB$FREE_VM        ;DEALLOCATE THE LIST ENTRY
                                05 0263   577          RSB
                50   10 A2   D0 0264   578 20$:    MOVL    LNK_L_TFRADR(R2),R0     ;WAS ROUTINE OBTAINED?
                        E4      13 0268   579          BEQL    10$                     ;IF EQL NO--JUST CONTINUE
                                07 DD 026A   580          PUSHL   #LNK_C_OUT_DEACCESS     ;CODE TO DEACESS
                        0C A2   9F 026C   581          PUSHAB  LNK_C_CONTEXT(R2)       ;CONTEXT ADDRESS
```

```
                                                I  3

60  02  FB  026F   582          CALLS   #2,(R0)                 ;CALL ALTERNATE NET PROTOCOL HANDLER
    DA  11  0272   583          BRB     10$                     ;DEALLOCATE LINK BLOCK
            0274   584
            0274   585          .END
```

```
BUFF_DESC                       =  00000004
CNCTDEF                         =  00000000
CNCT_B_FILRAT                   =  00000080
CNCT_B_FILRFM                   =  00000081
CNCT_B_FLAGS                    =  00000084
CNCT_B_TRAN                     =  00000085
CNCT_C_BUFSIZ                   =  00000200
CNCT_C_GETADDR                  =  00000001
CNCT_C_GETMSG                   =  00000004
CNCT_C_GETSENDER                =  00000000
CNCT_C_GETSUBJ                  =  00000003
CNCT_C_GETTO                    =  00000002
CNCT_C_LENGTH                   =  00000286
CNCT_L_BLINK                    =  00000004
CNCT_L_FILERR                   =  00000048
CNCT_L_FILESTS                  =  0000004C
CNCT_L_FILESTV                  =  00000050
CNCT_L_FLINK                    =  00000000
CNCT_L_MSGBLKS                  =  0000003C
CNCT_L_NADDRS                   =  00000038
CNCT_L_NRECS                    =  00000064
CNCT_L_SPARES                   =  00000068
CNCT_M_BLKMODE                  =  00000004
CNCT_M_DISCONNECT               =  00000001
CNCT_M_SQ                       =  00000002
CNCT_Q_ADRLST                   =  00000028
CNCT_Q_CCDESC                   =  00000020
CNCT_Q_ESDESC                   =  00000054
CNCT_Q_FROMDESC                 =  00000008
CNCT_Q_IOSB                     =  00000040
CNCT_Q_LNKLST                   =  00000030
CNCT_Q_SENDQ                    =  0000005C
CNCT_Q_SUBJDESC                 =  00000018
CNCT_Q_TODESC                   =  00000010
CNCT_S_ADRLST                   =  00000008
CNCT_S_BUFFER                   =  00000200
CNCT_S_CCDESC                   =  00000008
CNCT_S_CNCTDEF                  =  00000286
CNCT_S_ESDESC                   =  00000008
CNCT_S_FROMDESC                 =  00000008
CNCT_S_IOSB                     =  00000008
CNCT_S_LNKLST                   =  00000008
CNCT_S_SENDQ                    =  00000008
CNCT_S_SPARES                   =  00000018
CNCT_S_SUBJDESC                 =  00000008
CNCT_S_TODESC                   =  00000008
CNCT_T_BUFFER                   =  00000086
CNCT_T_RAB                      =  00000286
CNCT_V_BLKMODE                  =  00000002
CNCT_V_DISCONNECT               =  00000000
CNCT_V_SQ                       =  00000001
CNCT_W_CHAN                     =  00000082
CNCT_W_IOSB1                    =  00000040
CNCT_W_IOSB2                    =  00000042
CNCT_W_IOSB3                    =  00000044
CNCT_W_IOSB4                    =  00000046
DEALLOC_LNK                       0000023E R      03
FAOARG1                         =  00000008
FAOBUF                            0000000C R      02
FAODESC                           00000008 R      02
LIB$FREE_VM                       ********     X  03
LIB$GET_VM                        ********     X  03
LIB$SIGNAL                        ********     X  03
LNKDEF                          =  00000000
LNK_B_FLAGS                     =  0000002E
LNK_B_NODLEN                    =  0000002F
LNK_B_PNLEN                     =  0000004F
LNK_C_IN_CKUSER                 =  0000000A
LNK_C_IN_CONNECT                =  00000008
LNK_C_IN_FILE                   =  0000000D
LNK_C_IN_SENDER                 =  00000009
LNK_C_IN_SUBJ                   =  0000000C
LNK_C_IN_TO                     =  0000000B
LNK_C_IO_READ                   =  0000000E
LNK_C_IO_WRITE                  =  0000000F
LNK_C_LENGTH                    =  00000050
LNK_C_OUT_CKSEND                =  00000006
LNK_C_OUT_CKUSER                =  00000002
LNK_C_OUT_CONNECT               =  00000000
LNK_C_OUT_DEACCESS              =  00000007
LNK_C_OUT_FILE                  =  00000005
LNK_C_OUT_SENDER                =  00000001
LNK_C_OUT_SUBJ                  =  00000004
LNK_C_OUT_TO                    =  00000003
LNK_L_ADRCNT                    =  00000008
LNK_L_BLINK                     =  00000004
LNK_L_CONTEXT                   =  0000000C
LNK_L_FLINK                     =  00000000
LNK_L_SPARES                    =  00000020
LNK_L_STS                       =  0000001C
LNK_L_TFRADR                    =  00000010
LNK_M_ALTP                      =  00000004
LNK_M_BLKMODE                   =  00000010
LNK_M_DEAD                      =  00000002
LNK_M_FSENT                     =  00000008
LNK_M_MSGSNT                    =  00000001
LNK_Q_IOSB                      =  00000014
LNK_S_IOSB                      =  00000008
LNK_S_LNKDEF                    =  00000050
LNK_S_NODE                      =  0000001F
LNK_S_SPARES                    =  00000008
LNK_T_NODE                      =  00000030
LNK_T_PNAM                      =  00000050
LNK_V_ALTP                      =  00000002
LNK_V_BLKMODE                   =  00000004
LNK_V_DEAD                      =  00000001
LNK_V_FSENT                     =  00000003
LNK_V_MSGSNT                    =  00000000
LNK_W_CHAN                      =  0000002C
LNK_W_MBXCHAN                   =  0000002A
LNK_W_SPAREW                    =  00000028
MAILDEF                         =  00000000
MAIL_L_FLAGS                    =  00000000
MAIL_M_ALTP                     =  00000400
```

```
MAIF_M_ATTACHMENT               = 00080000
MAIF_M_BATCH                    = 00000200
MAIF_M_CALEDT                   = 00008000
MAIF_M_CAPTIVE                  = 00004000
MAIF_M_CTRLCFL                  = 00000100
MAIF_M_DFMSG                    = 00001000
MAIF_M_EDTEDT                   = 00010000
MAIF_M_EDTXYZ                   = 00020000
MAIF_M_FILOPN                   = 00000800
MAIF_M_ISAM                     = 00000001
MAIF_M_ITERM                    = 00000004
MAIF_M_NETJOB                   = 00000002
MAIF_M_OTERM                    = 00000008
MAIF_M_RDHOLD                   = 00000020
MAIF_M_READ_FL                  = 00000040
MAIF_M_RECL                     = 00000080
MAIF_M_SCOPE                    = 00000010
MAIF_M_SERVERLOOP               = 00040000
MAIF_M_USEREDIT                 = 00002000
MAIF_S_MAIFDEF                  = 00000004
MAIF_V_ALTP                     = 0000000A
MAIF_V_ATTACHMENT               = 00000013
MAIF_V_BATCH                    = 00000009
MAIF_V_CALEDT                   = 0000000F
MAIF_V_CAPTIVE                  = 0000000E
MAIF_V_CTRLCFL                  = 00000008
MAIF_V_DFMSG                    = 0000000C
MAIF_V_EDTEDT                   = 00000010
MAIF_V_EDTXYZ                   = 00000011
MAIF_V_FILOPN                   = 0000000B
MAIF_V_ISAM                     = 00000000
MAIF_V_ITERM                    = 00000002
MAIF_V_NETJOB                   = 00000001
MAIF_V_OTERM                    = 00000003
MAIF_V_RDHOLD                   = 00000005
MAIF_V_READ_FL                  = 00000006
MAIF_V_RECL                     = 00000007
MAIF_V_SCOPE                    = 00000004
MAIF_V_SERVERLOOP               = 00000012
MAIF_V_USEREDIT                 = 0000000D
MAIL$CONFIRM                      0000003C RG     03
MAIL$DECLINKADR                   000001F8 RG     03
MAIL$DELEMPLINK                   00000209 RG     03
MAIL$DELETELINK                   0000022D RG     03
MAIL$FAO_GETMSG                   00000107 RG     03
MAIL$GETMSG                       000000DB RG     03
MAIL$GL_FLAGS                     ******** X      03
MAIL$INCLINKADR                   000001E7 RG     03
MAIL$IOSUB_INIT                   00000000 RG     03
MAIL$K_INBUFFSZ                 = 00000200
MAIL$L_SMG_KEYBOARD               ******** X      03
MAIL$MATCH_NAME                   00000137 RG     03
MAIL$READ_ERROR_TEXT              00000197 RG     03
MAIL$UPCASE                       00000076 RG     03
MAIL$UPCASE_Q                     0000009D RG     03
MAIL$_FACILITY                    ******** X      03
MSGBUF                            00000004 R      02
MSGDESC                           00000000 R      02
SHR$_TEXT                       = 00001130
SMG$READ_STRING                   ******** X    03
STR$ANALYZE_SDESC_R1              ******** X    03
STR$_MATCH                      = 00000001
STR$_NOMATCH                    = 00000000
STS$V_FAC_NO                    = 00000010
SYS$DASSGN                        ******** GX   03
SYS$FAOL                          ******** GX   03
SYS$GETMSG                        ******** GX   03
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
OWN	00000010 (16.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
$CODE$	00000274 (628.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.03	00:00:02.76
Command processing	111	00:00:00.44	00:00:01.67
Pass 1	195	00:00:02.46	00:00:14.85
Symbol table sort	0	00:00:00.29	00:00:01.81
Pass 2	113	00:00:00.87	00:00:03.44
Symbol table output	22	00:00:00.10	00:00:00.10
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	474	00:00:04.21	00:00:24.65

The working set limit was 1350 pages.
21256 bytes (42 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 320 non-local and 32 local symbols.
585 source lines were read in Pass 1, producing 51 object records in Pass 2.
20 pages of virtual memory were used to define 15 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[MAIL.OBJ]MAILMAC.MLB;1	3
_255DUA28:[SYS.OBJ]LIB.MLB;1	0
_255DUA28:[SYSLIB]STARLET.MLB;2	9
TOTALS (all libraries)	12

385 GETS were required to define 12 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:IOSUBS/OBJ=OBJ$:IOSUBS MSRC$:IOSUBS/UPDATE=(ENH$:IOSUBS)+EXECML$/LIB+LIB$:MAILMAC/LIB
```

IOSUBS
LIS

HANDLER
LIS

MAIL
LIS

ISUBS
LIS