```
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPPP
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPPP
UUU         UUU  EEE                    TTT        PPP         PPP
UUU         UUU  EEE                    TTT        PPP         PPP
UUU         UUU  EEE                    TTT        PPP         PPP
UUU         UUU  EEE                    TTT        PPP         PPP
UUU         UUU  EEE                    TTT        PPP         PPP
UUU         UUU  EEE                    TTT        PPP         PPP
UUU         UUU  EEEEEEEEEEEE           TTT        PPPPPPPPPPPPP
UUU         UUU  EEEEEEEEEEEE           TTT        PPPPPPPPPPPPP
UUU         UUU  EEEEEEEEEEEE           TTT        PPPPPPPPPPPPP
UUU         UUU  EEE                    TTT        PPP
UUU         UUU  EEE                    TTT        PPP
UUU         UUU  EEE                    TTT        PPP
UUU         UUU  EEE                    TTT        PPP
UUU         UUU  EEE                    TTT        PPP
UUU         UUU  EEE                    TTT        PPP
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP
```

```
RRRRRRRR    MM      MM   SSSSSSSS  TTTTTTTTTT  EEEEEEEEEE   SSSSSSSS  TTTTTTTTTT    666666
RRRRRRRR    MM      MM   SSSSSSSS  TTTTTTTTTT  EEEEEEEEEE   SSSSSSSS  TTTTTTTTTT    666666
RR      RR  MMMM  MMMM  SS            TT      EE           SS            TT        66
RR      RR  MMMM  MMMM  SS            TT      EE           SS            TT        66
RR      RR  MM  MM  MM  SS            TT      EE           SS            TT        66
RR      RR  MM  MM  MM  SS            TT      EE           SS            TT        66
RRRRRRRR    MM      MM    SSSSSS      TT      EEEEEEEE       SSSSSS      TT        66666666
RRRRRRRR    MM      MM    SSSSSS      TT      EEEEEEEE       SSSSSS      TT        66666666
RR  RR      MM      MM          SS    TT      EE                   SS    TT        66      66
RR  RR      MM      MM          SS    TT      EE                   SS    TT        66      66
RR    RR    MM      MM          SS    TT      EE                   SS    TT        66      66      ....
RR    RR    MM      MM          SS    TT      EE                   SS    TT        66      66      ....
RR      RR  MM      MM  SSSSSSSS      TT      EEEEEEEEEE  SSSSSSSS      TT          666666        ....
RR      RR  MM      MM  SSSSSSSS      TT      EEEEEEEEEE  SSSSSSSS      TT          666666        ....

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     71          $BEGIN  RMSTEST6,007,__RMSTEST,<TEST MAGTAPE>,<GBL,LONG>
0000     72
0000     73
0000     74          .ENABL  DBG
0000     75
0000     76
0000     77          $DEVDEF
0000     78          .NLIST  MEB
0000     79
0000     80 ;
0000     81 ;  macros:
0000     82 ;
0000     83
0000     84          .MACRO  BUFF NAM,SIZE
0000     85 NAM'BUF::
0000     86          .BLKB   SIZE
0000     87          NAM'BSZ==SIZE
0000     88          .ENDM   BUFF
0000     89
0000     90
0000     91          .MACRO  TYPE STRING, ?L
0000     92          STORE   <STRING>
0000     93          BLBC    VERBOSITY,L
0000     94          MOVL    #$$.TMPX,CMDORAB+RAB$L_RBF
0000     95          MOVW    #$$.TMPX1,CMDORAB+RAB$W_RSZ
0000     96          $PUT    RAB=CMDORAB,ERR=REPORT_ERROR
0000     97          BSBW    ERR
0000     98 L:
0000     99          .ENDM   TYPE
0000    100
0000    101
0000    102          .MACRO  WTTYPE STRING
0000    103          $WAIT   RAB=CMDORAB
0000    104          TYPE    <STRING>
0000    105          .ENDM   WTTYPE
0000    106
0000    107
0000    108          .MACRO  STORE STRING,PRE
0000    109          .SAVE
0000    110          .PSECT  __$RMSNAM
0000    111          $$.TMPX=.
0000    112          PRE                                     ; store any carriage control info
0000    113          .ASCII  %STRING%
0000    114          $$.TMPX1=.-$$.TMPX
0000    115          .RESTORE
0000    116          .ENDM   STORE
```

```
0000    118
0000    119
0000    120          .MACRO  BEGIN TSTNAM
0000    121          STORE   <TSTNAM>
0000    122          MOVL    #$$.TMPX,BEG_DESCR+4    ; addr
0000    123          MOVL    #$$.TMPX1,BEG_DESCR     ; len
0000    124          BSBW    BEGPUT
0000    125          .ENDM   BEGIN
0000    126          .MACRO  FINISH TSTNAM
0000    127          STORE   <TSTNAM>
0000    128          MOVL    #$$.TMPX,FIN_DESCR+4    ; addr
0000    129          MOVL    #$$.TMPX1,FIN_DESCR     ; len
0000    130          BSBW    FINPUT
0000    131          .ENDM   FINISH
0000    132          .MACRO  FIELD FLDNAM
0000    133          STORE   <FLDNAM>
0000    134          MOVL    #$$.TMPX,FLD_DESCR+4    ; addr
0000    135          MOVL    #$$.TMPX1,FLD_DESCR     ; len
0000    136          BSBW    FLDPUT
0000    137          .ENDM   FIELD
0000    138          .MACRO  MBPT, ?L
0000    139          BLBC    VERBOSITY,L
0000    140          BPT
0000    141 L:
0000    142          .ENDM   MBPT
0000    143
0000    144
```

```
                     00000000   146          .PSECT   RMSTEST,GBL,LONG
                         0000   147          .ALIGN   LONG
                         0000   148 T6START::
                         0000   149 MTAFAB: $FAB    DNM=<TST$MTA:.FIL;1>,fns=fnsize
                         0050   150 MTARAB: $RAB    RHB=RHBBUF,UBF=IOBUF,FAB=MTAFAB
                         0094   151          BUFF    IO,512.
                 00000000  0294   152 BUF:    .LONG   0                       ; key for random lookup
                         0298   153
                         0298   154 ;
                         0298   155 ;  file names for various files
                         0298   156 ;
                         0298   157
                         0298   158 MTVAR_FNM:
                52 41 56 54 4D   0298   159          .ASCII  \MTVAR\
                     00000005    029D   160          FNSIZE=.-MTVAR_FNM
                         029D   161 MTFIX_FNM:
                58 49 46 54 4D   029D   162          .ASCII  \MTFIX\
                         02A2   163 MTVFC_FNM:
                43 46 56 54 4D   02A2   164          .ASCII  \MTVFC\
                         02A7   165 MTST1_FNM:
                31 54 53 54 4D   02A7   166          .ASCII  \MTST1\
                         02AC   167 NONFIL_FNM:
                58 49 46 46 4E   02AC   168          .ASCII  \NFFIX\
```

```
                        02B1   170
                        02B1   171 ;
                        02B1   172 ;
                        02B1   173 ; this is a test program to test out the use of magtapes through rms-32
                        02B1   174 ; it exercises the basic functionality
                        02B1   175 ;
                        02B1   176
                        02B1   177 TST$MTA::
                  0FFC  02B1   178         .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                        02B3   179
                        02B3   180 ;
                        02B3   181 ; create 3 files on the tape - with different record formats, and fill them
                        02B3   182 ;
                        02B3   183
                        02B3   184         BEGIN   <MAGTAPE TESTS>
                        02C8   185
                        02C8   186 ;
                        02C8   187 ; the 1st file -- rfm=fix
                        02C8   188 ;
                        02C8   189
              59  FD84 CF DE 02C8   190         MOVAL   MTARAB,R9
                        02CD   191         $FAB_STORE      FAB=MTAFAB,RFM=FIX,MRS=#50,FNA=MTFIX_FNM,-
                        02CD   192         FOP=<RWO,SUP>,RAT=<CR,BLK>,FAC=PUT
                        02EF   193         $RAB_STORE      RAB=R9,RSZ=#50,RBF=IOBUF
                        02F9   194
                        02F9   195 ;
                        02F9   196 ; create a file and close it in order to discover if the device is mounted foreign
                        02F9   197 ;
                        02F9   198
                        02F9   199         $CREATE FAB=MTAFAB,ERR=REPORT_ERROR
                FCF3'   30     030A   200         BSBW    ERR
                07B2    30     030D   201         BSBW    CLOSE_MT
     04 FD2B CF   18    E0     0310   202         BBS     #DEV$V_FOR,FAB$L_DEV+MTAFAB,5$
                0005    30     0316   203         BSBW    STRMT
                        04     0319   204         RET
                051E    30     031A   205 5$:     BSBW    NON_FILE
                        04     031D   206         RET
                        031E   207 STRMT:
                        031E   208         TYPE    <CREATE 3 TEST FILES ON TAPE>
              56  32    D0     034D   209         MOVL    #50,R6                  ; alternate record size
                0758    30     0350   210         BSBW    DO_PUT
                        0353   211
                        0353   212 ;
                        0353   213 ;  now the 2nd file w/ rfm = var
                        0353   214 ;
                        0353   215
                        0353   216         $FAB_STORE      FAB=MTAFAB,FNA=MTVAR_FNM,RFM=VAR,MRS=#0,BLS=#512,-
                        0353   217         FOP=<POS,SUP>
     04 40 A0   05    E1     0373   218         BBC     #DEV$V_SQD,FAB$L_DEV(R0),10$; branch if not magtape
        1E A0   08    8A     0378   219         BICB2   #FAB$M_BLK,FAB$B_RAT(R0); verify blk defaulting
                        037C   220 10$:    $RAB_STORE      RAB=R9,RSZ=#79  ; primary length
              56  3B    9A     0382   221         MOVZBL  #59,R6                  ; alternate size
                0723    30     0385   222         BSBW    DO_PUT
                        0388   223
                        0388   224 ;
                        0388   225 ;  now the 3rd file w/ rfm = vfc
                        0388   226 ;
```

```
                     0388   227
                     0388   228          $FAB_STORE      FAB=MTAFAB,FNA=MTVFC_FNM,RFM=VFC,BLS=#1024,-
                     0388   229          FOP=<POS,RWC,SUP>,FSZ=#4
                     03A9   230          $RAB_STORE      RAB=R9,RSZ=#75  ; primary size
56   37   D0         03AF   231          MOVL    #55,R6                  ; alternate size
   06F6   30         03B2   232          BSBW    DO_PUT
```

```
                                03B5    234
                                03B5    235 ;++
                                03B5    236 ;
                                03B5    237 ;  reread 3 created files and verify their contents
                                03B5    238 ;
                                03B5    239 ;--
                                03B5    240
                                03B5    241          TYPE    <RE-READ FILES CREATED IN STEP 1>
                                03E4    242          $RAB_STORE      RAB=R9,ROP=<LOC>
                                03EC    243          $FAB_STORE      FAB=MTAFAB,FNA=MTFIX_FNM,FOP=#0,-
                                03EC    244          RAT=#0,ORG=REL,FAC=<GET,BRO>
                    072F    30  0406    245          BSBW    OPEN_MT
            01  FC12 CF    91  0409    246          CMPB    MTAFAB+FAB$B_RFM,#FAB$C_FIX; is it fixed rfm?
                    15      13  040E    247          BEQL    RMT1
                                0410    248          FIELD   <RFM (NOT = FIXED)>
          1E FC16 CF    05    E1  0425    249 RMT1:    BBC     #DEV$V_SQD,FAB$L_DEV+MTAFAB,RMT2; skip test if disk
      0800 8F   FC0D CF    B1  042B    250          CMPW    MTAFAB+FAB$W_BLS,#2048  ; is block size right?
                    15      13  0432    251          BEQL    RMT2
                                0434    252          FIELD   <BLS>
            32  FBE9 CF    B1  0449    253 RMT2:    CMPW    MTAFAB+FAB$W_MRS,#50    ; is record size right?
                    15      13  044E    254          BEQL    RMT3
                                0450    255          FIELD   <MRS>
                  0738      30  0465    256 RMT3:    BSBW    READ_MT                 ; read and verify records
                       32   32  0468    257          .BYTE   50,50
                  081A      30  046A    258          BSBW    EOF_CLOSE               ; check eof and close
                                046D    259
                                046D    260 ;
                                046D    261 ;  2nd file
                                046D    262 ;
                                046D    263
                                046D    264          $FAB_STORE      FAB=MTAFAB,FNA=MTVAR_FNM,RAT=#0,ORG=REL
                  06B6      30  047F    265          BSBW    OPEN_MT
            02  FB99 CF    91  0482    266          CMPB    MTAFAB+FAB$B_RFM,#FAB$C_VAR
                    15      13  0487    267          BEQL    RMT4
                                0489    268          FIELD   <RFM (NOT = VAR)>
      0200 8F   FB9A CF    B1  049E    269 RMT4:    CMPW    MTAFAB+FAB$W_BLS,#512   ; is block size correct?
                    15      13  04A5    270          BEQL    RMT5
                                04A7    271          FIELD   <BLS>
                FB76 CF    B5  04BC    272 RMT5:    TSTW    MTAFAB+FAB$W_MRS        ; is max record size right?
                    15      13  04C0    273          BEQL    RMT6
                                04C2    274          FIELD   <MRS>
                  06C1      30  04D7    275 RMT6:    BSBW    READ_MT_VAR             ; read & verify records
                  07AA      30  04DA    276          BSBW    EOF_CLOSE
                                04DD    277
                                04DD    278 ;
                                04DD    279 ;  3rd file
                                04DD    280 ;
                                04DD    281
                                04DD    282          $FAB_STORE      FAB=MTAFAB,FNA=MTVFC_FNM,RAT=#0,ORG=REL
                  0646      30  04EF    283          BSBW    OPEN_MT
            03  FB29 CF    91  04F2    284          CMPB    FAB$B_RFM+MTAFAB,#FAB$C_VFC; right record format?
                    15      13  04F7    285          BEQL    RMT7
                                04F9    286          FIELD   <RFM (NOT = VFC)>
          1E FB2D CF    05    E1  050E    287 RMT7:    BBC     #DEV$V_SQD,FAB$L_DEV+MTAFAB,RMT8; skip test if disk
      0400 8F   FB24 CF    B1  0514    288          CMPW    MTAFAB+FAB$W_BLS,#1024  ; right block size?
                    15      13  051B    289          BEQL    RMT8
                                051D    290          FIELD   <BLS>
```

```
FB00 CF   B5  0532   291 RMT8:   TSTW    MTAFAB+FAB$W_MRS        ; mrs ok?
     15   13  0536   292         BEQL    RMT9
                0538   293         FIELD   <MRS>
  0650_   30  054D   294 RMT9:    BSBW    READ_MT
       37 4B  0550   295         .BYTE   75,55
  0732    30  0552   296         BSBW    EOF_CLOSE
```

```
                     0555   298
                     0555   299 ;++
                     0555   300 ;
                     0555   301 ;  re-open mtvar.fil and test spacing and rewind
                     0555   302 ;
                     0555   303 ;--
                     0555   304
                     0555   305          TYPE    <PERFORM SPACING AND REWIND TESTS>
                     0584   306          $FAB_STORE      FAB=MTAFAB,FNA=MTVAR_FNM
05A6   30             058F   307          BSBW    OPEN_MT
                     0592   308          $RAB_STORE      RAB=R9,BKT=#16  ; space count
                     0596   309          $SPACE  R9
06F3   30             059F   310          BSBW    CHKEOF1                 ; verify at eof
                     05A2   311          $RAB_STORE      R9,BKT=#16
                     05A6   312          $SPACE  R9                      ; space forward at eof
06E3   30             05AF   313          BSBW    CHKEOF1                 ; verify still at eof
                     05B2   314          $RAB_STORE      R9,BKT=#-16     ; backspace count
                     05BA   315          $SPACE  R9
06DF   30             05C3   316          BSBW    CHKBOF                  ; verify at start of file
                     05C6   317          $RAB_STORE      R9,BKT=#8
                     05CA   318          $SPACE  RAB=R9,ERR=REPORT_ERROR
FA24'  30             05D9   319          BSBW    ERR
                     05DC   320          $RAB_STORE      R9,BKT=#0       ; count = 0
                     05DF   321          $SPACE  RAB=R9,ERR=REPORT_ERROR
FA0F'  30             05EE   322          BSBW    ERR
                     05F1   323          $RAB_STORE      R9,BKT=#-4
                     05F9   324          $SPACE  RAB=R9,ERR=REPORT_ERROR
F9F5'  30             0608   325          BSBW    ERR
                     060B   326
                     060B   327 ;
                     060B   328 ;  read records from block 4 thru end of file, verifying goodness
                     060B   329 ;
                     060B   330
57   39   9A         060B   331          MOVZBL  #^A/!/+24,R7            ; starting rec #
  0592   30           060E   332          BSBW    LOOP1                   ; verify records
     3B 4F           0611   333          .BYTE   79,59
                     0613   334          $RAB_STORE      R9,BKT=#-100
                     061B   335          $SPACE  R9
067E   30             0624   336          BSBW    CHKBOF                  ; verify at start
0571   30             0627   337          BSBW    READ_MT_VAR
                     062A   338          $REWIND RAB=R9,ERR=REPORT_ERROR
F9C4'  30             0639   339          BSBW    ERR
055C   30             063C   340          BSBW    READ_MT_VAR
0645   30             063F   341          BSBW    EOF_CLOSE
```

```
                                0642    343
                                0642    344 ;++
                                0642    345 ;
                                0642    346 ;  test of "create if" and $nxtvol
                                0642    347 ;
                                0642    348 ;--
                                0642    349
                                0642    350          TYPE    <PERFORM "CREATE IF", $TRUNCATE  AND $NXTVOL TESTS>
                                0671    351          $FAB_STORE      FAB=MTAFAB,FAC=PUT,FNA=MTST1_FNM,-
                                0671    352          FOP=CIF,RFM=VAR
                                068C    353          $ERASE  FAB=R0                  ; erase in case disk file
                         0475   30      0695    354          BSBW    CREATE_MT
00000000'8F     F96C CF         D1      0698    355          CMPL    MTAFAB+FAB$L_STS,#RMS$_CREATED
                           18   13      06A1    356          BEQL    RMT11
                                06A3    357          FIELD   <STS (EXPECTED "RMS$_CREATED")>
                        F945'   30      06B8    358          BSBW    ERR
                                06BB    359 RMT11:   $RAB_STORE      RAB=R9,RBF=IOBUF,RSZ=#50
                   04 A9    02  88      06C5    360          BISB2   #RAB$M_TPT,RAB$L_ROP(R9); allow put not at eof
                                06C9    361          $PUT    RAB=R9,ERR=REPORT_ERROR
                        F925'   30      06D8    362          BSBW    ERR
                        03E4    30      06DB    363          BSBW    CLOSE_MT
                                06DE    364
                                06DE    365 ;
                                06DE    366 ;  now re-open file just created above and overwrite record
                                06DE    367 ;
                                06DE    368
                                06DE    369          $FAB_STORE      FAB=MTAFAB,FAC=<GET,PUT,TRN>,FOP=<CIF,NEF>
                         041B   30      06EF    370          BSBW    CREATE_MT
00000000'8F     F912 CF         D1      06F2    371          CMPL    MTAFAB+FAB$L_STS,#RMS$_NORMAL
                           18   13      06FB    372          BEQL    RMT12
                                06FD    373          FIELD   <STS (EXPECTED "NORMAL")>
                        F8EB'   30      0712    374          BSBW    ERR
                                0715    375 RMT12:   $RAB_STORE      RAB=R9,RBF=IOBUF,RSZ=#79
                   56    3B    D0      0721    376          MOVL    #59,R6                  ; alternate rec size
                         03B0   30      0724    377          BSBW    PUTRECS
                                0727    378
                                0727    379 ;
                                0727    380 ;       test truncate
                                0727    381 ;
                                0727    382
                         03AD   30      0727    383          BSBW    PUTRECS                 ; write 2nd half of file
                                072A    384          $REWIND RAB=R9,ERR=REPORT_ERROR
                        F8C4'   30      0739    385          BSBW    ERR
                         045C   30      073C    386          BSBW    READ_MT_VAR             ; read 1st half of file
                                073F    387          $GET    RAB=R9,ERR=REPORT_ERROR ; need to advance the current
                        F8AF'   30      074E    388          BSBW    ERR                     ; record pointer before truncate
                                0751    389          $TRUNCATE       RAB=R9,ERR=REPORT_ERROR
                        F89D'   30      0760    390          BSBW    ERR                     ; truncate 2nd half of file
                         0526   30      0763    391          BSBW    CHKEOF                  ; check for eof
                                0766    392          $RAB_STORE      RAB=R9,RBF=IOBUF,RSZ=#79
                   56    3B    D0      0772    393          MOVL    #59,R6
                         035F   30      0775    394          BSBW    PUTRECS                 ; re-put 2nd half
                                0778    395          $REWIND RAB=R9,ERR=REPORT_ERROR
                        F876'   30      0787    396          BSBW    ERR
                         040E   30      078A    397          BSBW    READ_MT_VAR             ; read 1st half
                         040B   30      078D    398          BSBW    READ_MT_VAR             ; read 2nd half, verifying
                         04F9   30      0790    399          BSBW    CHKEOF                  ; check for eof again
```

```
                     0793    400          $REWIND RAB=R9,ERR=REPORT_ERROR
          F85B'   30 07A2    401          BSBW    ERR
          03F3    30 07A5    402          BSBW    READ_MT_VAR              ; read 1st half
                     07A8    403          $GET    RAB=R9,ERR=REPORT_ERROR ; need to advance the current
          F846'   30 07B7    404          BSBW    ERR                      ; record pointer before truncate
                     07BA    405          $TRUNCATE       RAB=R9,ERR=REPORT_ERROR; trunc 2nd half
          F834'   30 07C9    406          BSBW    ERR                      ; leave file as before
 19 F86F CF    05   E1 07CC    407          BBC     #DEV$V_SQD,FAB$L_DEV+MTAFAB,20$; branch if not really mt
  12 00000000'EF    E9 07D2    408          BLBC    VERBOSITY,20$            ; branch if not interactive
                     07D9    409          $NXTVOL RAB=R9,ERR=REPORT_ERROR
          F815'   30 07E8    410          BSBW    ERR
                     07EB    411 20$:     $RAB_STORE      RAB=R9,RBF=IOBUF,RSZ=#79
       56    3B   D0 07F7    412          MOVL    #59,R6
          02DA    30 07FA    413          BSBW    PUTRECS
                     07FD    414          $REWIND RAB=R9,ERR=REPORT_ERROR
          F7F1'   30 080C    415          BSBW    ERR
          0389    30 080F    416          BSBW    READ_MT_VAR              ; read recs from rel vol 1
          0386    30 0812    417          BSBW    READ_MT_VAR              ; read recs form rel vol 2
                     0815    418          $FAB_STORE      FAB=MTAFAB,FOP=<RWC>; cause volume set rewind
          0462    30 0822    419          BSBW    EOF_CLOSE
                     0825    420          FINISH  <MAGTAPE TESTS>
                     083A    421
                05   083A    422          RSB
```

```
                        083B    424
                        083B    425 ;++
                        083B    426 ; non-file structured tests
                        083B    427 ;--
                        083B    428
                        083B    429 ;
                        083B    430 ; first write out a fix length 512 byte record file
                        083B    431 ; note:  set tst$mta to disk if non_file structured disk test wanted.
                        083B    432 ; mount/fo dev /rec=512/blo=512
                        083B    433 ;
                        083B    434
                        083B    435 NON_FILE:
                        083B    436         TYPE    <NON_FILE STRUCTURE CREATE TEST>
          59   F7E2 CF   DE    086A    437         MOVAL   MTARAB,R9
                        086F    438         $FAB_STORE      FAB=MTAFAB,RFM=FIX,MRS=#512,FNA=NONFIL_FNM,-
                        086F    439         RAT=#0,FAC=PUT,BLS=#512
                        0891    440         $RAB_STORE      RAB=R9,RSZ=#512,RBF=IOBUF
       56   00000200 8F   D0    089D    441         MOVL    #512,R6
                  0204   30    08A4    442         BSBW    DO_PUT
                        08A7    443
                        08A7    444 ;
                        08A7    445 ; now open file and check attributes
                        08A7    446 ;
                        08A7    447
                        08A7    448         $RAB_STORE      RAB=R9,ROP=<LOC>
                        08AF    449         $FAB_STORE      FAB=MTAFAB,FNA=NONFIL_FNM,FOP=#0,RAT=#0,-
                        08AF    450         FAC=<GET,BRO>
                  0270   30    08C5    451         BSBW    OPEN_MT
                        08C8    452         $REWIND RAB=R9,ERR=REPORT_ERROR
                  F726'  30    08D7    453         BSBW    ERR
                  01E5   30    08DA    454         BSBW    CLOSE_MT
       50   00000000'8F   D0    08DD    455         MOVL    #MTAFAB,R0
                  0251   30    08E4    456         BSBW    OPEN_MT
            01   F734 CF   91    08E7    457         CMPB    MTAFAB+FAB$B_RFM,#FAB$C_FIX
                        15    13    08EC    458         BEQL    NF10
                        08EE    459         FIELD   <RFM(NOT FIXED)>
        1E F738 CF    05    E1    0903    460 NF10:   BBC     #DEV$V_SQD,FAB$L_DEV+MTAFAB,NF20
  0200 8F   F72F CF   B1    0909    461         CMPW    MTAFAB+FAB$W_BLS,#512
                        15    13    0910    462         BEQL    NF20
                        0912    463         FIELD   <BLS>
  0200 8F   F70B CF   B1    0927    464 NF20:   CMPW    MTAFAB+FAB$W_MRS,#512
                        15    13    092E    465         BEQL    NF30
                        0930    466         FIELD   <MRS>
     03 F6F6 CF    05    E0    0945    467 NF30:   BBS     #DEV$V_SQD,FAB$L_DEV+MTAFAB,NF35
                  0089   31    094B    468         BRW     NF40
                        094E    469 NF35:   TYPE    <NON-FILE STRUCTURED MT TESTS>
                        097D    470         $RAB_STORE      RAB=R9,BKT=#10
                        0981    471         $SPACE  RAB=R9,ERR=REPORT_ERROR
                  F66D'  30    0990    472         BSBW    ERR
                        0993    473
                        0993    474 ;
                        0993    475 ; read records from block 11 thru end
                        0993    476 ;
                        0993    477
          57    2B    9A    0993    478         MOVZBL  #^A/!/ + 10,R7
                  020A   30    0996    479         BSBW    LOOP1
                     00 00    0999    480         .BYTE   0,0
```

```
                                 099B   481          $RAB_STORE      RAB=R9,BKT=#-10
                                 09A3   482          $SPACE  RAB=R9,ERR=REPORT_ERROR
                          F64B'  30     09B2   483          BSBW    ERR
                       57 71 8F  9A     09B5   484          MOVZBL  #^X71,R7
                          01E7   30     09B9   485          BSBW    LOOP1
                             00  00     09BC   486          .BYTE   0,0
                          02C6   30     09BE   487          BSBW    EOF_CLOSE
                                 09C1   488          FINISH  <NON-FILE STRUCTURED MT TESTS>
                             05  09D6   489          RSB
                                 09D7   490
                                 09D7   491 ;
                                 09D7   492 ; non-file structured random disk test
                                 09D7   493 ;
                                 09D7   494
                                 09D7   495 NF40:    TYPE    <NON_FILE STRUCTURED RANDOM DISK TEST>
                                 0A06   496
                                 0A06   497 ;
                                 0A06   498 ; now open for keyed access
                                 0A06   499 ;
                                 0A06   500
                                 0A06   501          $RAB_STORE      RAB=R9,ROP=<LOC>,RAC=<KEY>,KBF=BUF
                    F877 CF   01 9A     0A18   502          MOVZBL  #1,BUF
                          0034   30     0A1D   503          BSBW    CHECK_ONE
                 F86E CF  59 8F  9A     0A20   504          MOVZBL  #89,BUF
                          002B   30     0A26   505          BSBW    CHECK_ONE
                    F866 CF   19 9A     0A29   506          MOVZBL  #25,BUF
                                 0A2E   507
                          0023   30     0A2E   508 NF50:    BSBW    CHECK_ONE
       F3 F85A CF  00000059 8F  F3     0A31   509          AOBLEQ  #89,BUF,NF50
                          0084   30     0A3B   510          BSBW    CLOSE_MT
                                 0A3E   511          FINISH  <NON-FILE STRUCTURE DISK TESTS>
                             05  0A53   512          RSB
                                 0A54   513
                                 0A54   514 ;
                                 0A54   515 ; check one disk record
                                 0A54   516 ;
                                 0A54   517
                                 0A54   518 CHECK_ONE:
                                 0A54   519          $GET    RAB=R9,ERR=REPORT_ERROR
                          F59A'  30     0A63   520          BSBW    ERR
                       50    21  9A     0A66   521          MOVZBL  #^A/!/,R0
                    50    F827 CF  C0     0A69   522          ADDL2   BUF,R0
                 22 A9    0200 8F  B1     0A6E   523          CMPW    #512,RAB$W_RSZ(R9)
                             15  13     0A74   524          BEQL    ONE10
                                 0A76   525          FIELD   <RSZ>
6E  00  50  28 B9  22 A9     2D  0A8B   526 ONE10:   CMPC5   RAB$W_RSZ(R9),@RAB$L_RBF(R9),R0,-
                                 0A93   527          #0,(SP)
                             15  13     0A93   528          BEQL    ONE20
                                 0A95   529          FIELD   <RECORD>
                             05  0AAA   530 ONE20:   RSB
```

```
                        0AAB   532
                        0AAB   533 ;++
                        0AAB   534 ;
                        0AAB   535 ;  subroutines to do the nitty-gritties of the magtape tests
                        0AAB   536 ;
                        0AAB   537 ;--
                        0AAB   538 ;
                        0AAB   539 ;       routine to create a magtape test file
                        0AAB   540 ;
                        0AAB   541 ;       r9 points to the rab
                        0AAB   542 ;       rab$w_rsz(r9) contains the size of the records to be output
                        0AAB   543 ;       r6 is the alternate size of the records to be output
                        0AAB   544 ;       r7 will be destroyed
                        0AAB   545 ;       r0 has fab address
                        0AAB   546 ;
                        0AAB   547 ;--
                        0AAB   548
                        0AAB   549 DO_PUT:
005F   30               0AAB   550         BSBW    CREATE_MT
  27   10               0AAE   551         BSBB    PUTRECS                 ; create the records
                        0AB0   552         $DISCONNECT     RAB=R9,ERR=REPORT_ERROR
F53E'  30               0ABF   553         BSBW    ERR
                        0AC2   554 CLOSE_MT:
                        0AC2   555         $CLOSE  FAB=MTAFAB,ERR=REPORT_ERROR
F52A'  30               0AD3   556         BSBW    ERR
       05               0AD6   557         RSB
                        0AD7   558
                        0AD7   559 ;
                        0AD7   560 ;  subroutine to actually put the records in the file
                        0AD7   561 ;
                        0AD7   562
                        0AD7   563 PUTRECS:
                  57    21   D0  0AD7   564         MOVL    #^A/!/,R7               ; starting character
                        0ADA   565 LOOP:                                ; loop to put  records out
F5B1 CF   22 A9   57    6E   00  2C  0ADA   566         MOVC5   #0,(SP),R7,RAB$W_RSZ(R9),IOBUF; fill iobuf w/ characters
                  2C B9 57    D0  0AE3   567         MOVL    R7,@RAB$L_RHB(R9)       ; fill in header
                        0AE7   568         $PUT    RAB=R9,ERR=REPORT_ERROR
                  F507' 30    0AF6   569         BSBW    ERR
                  50  22 A9    B0  0AF9   570         MOVW    RAB$W_RSZ(R9),R0        ; swap rec lengths
                  22 A9  56    B0  0AFD   571         MOVW    R6,RAB$W_RSZ(R9)
                     56  50    D0  0B01   572         MOVL    R0,R6
        CE 57   0000007B 8F    F2  0B04   573         AOBLSS  #^X7B,R7,LOOP
                           05   0B0C   574         RSB
```

```
                        0B0D    576
                        0B0D    577 ;
                        0B0D    578 ;  routine to create a file
                        0B0D    579 ;
                        0B0D    580
                        0B0D    581 CREATE_MT:
                        0B0D    582         $CREATE FAB=MTAFAB,ERR=REPORT_ERROR
          F4DF'  30     0B1E    583         BSBW    ERR
F4DF  CF    D4          0B21    584         CLRL    FAB$L_FOP+MTAFAB
                        0B25    585         $CONNECT        RAB=R9,ERR=REPORT_ERROR
          F4C9'  30     0B34    586         BSBW    ERR
                 05     0B37    587         RSB
                        0B38    588
                        0B38    589
                        0B38    590 ;
                        0B38    591 ;  routine to open a mt file
                        0B38    592 ;
                        0B38    593
                        0B38    594 OPEN_MT:
                        0B38    595         $OPEN   FAB=R0,ERR=REPORT_ERROR
          F4B6'  30     0B47    596         BSBW    ERR
                        0B4A    597         $CONNECT        RAB=R9,ERR=REPORT_ERROR
          F4A4'  30     0B59    598         BSBW    ERR
   00   F4BD  CF   91   0B5C    599         CMPB    MTAFAB+FAB$B_ORG,#FAB$C_SEQ
             15   13   0B61    600         BEQL    OPN1
                        0B63    601         FIELD   <ORG (NOT SEQUENTIAL)>
1C F4C3  CF   18   E0   0B78    602 OPN1:   BBS     #DEV$V_FOR,FAB$L_DEV+MTAFAB,OPN2; branch if foreign test
   F49B  CF   02   93   0B7E    603         BITB    #FAB$M_CR,MTAFAB+FAB$B_RAT
             15   12   0B83    604         BNEQ    OPN2
                        0B85    605         FIELD   <RAT (CR NOT SET)>
                 05     0B9A    606 OPN2:   RSB
```

```
                                0B9B    608
                                0B9B    609 ;
                                0B9B    610 ;  read and check magtape file
                                0B9B    611 ;
                                0B9B    612
                                0B9B    613 READ_MT_VAR:
                         03 10  0B9B    614         BSBB    READ_MT
                            3B  0B9D    615         .BYTE   79,59
                                0B9F    616         RSB
                                0BA0    617
                                0BA0    618 READ_MT:
                      57  21 D0 0BA0    619         MOVL    #^A/!/,R7                       ; starting character
                                0BA3    620 LOOP1:
                   51  3C A9 D0 0BA3    621         MOVL    RAB$L_FAB(R9),R1                ; Get the FAB address
                   1F A1  01 91 0BA7    622         CMPB    #FAB$C_FIX,FAB$B_RFM(R1) ; See if file is fixed
                          0B 12 0BAB    623         BNEQ    10$                             ; If not then skip next test
                57  0000005E 8F D1 0BAD    624         CMPL    #^A/^/,R7                       ; otherwise is the record of carret's
                          02 12 0BB4    625         BNEQ    10$                             ; Not carrets then continue
                          57 D6 0BB6    626         INCL    R7                              ; Skip that record
                                0BB8    627 10$:
                                0BB8    628         $GET    RAB=R9,ERR=REPORT_ERROR
                       F436' 30 0BC7    629         BSBW    ERR
            00000094'8F  28 A9  D1 0BCA    630         CMPL    RAB$L_RBF(R9),#IOBUF
                          1D 12 0BD2    631         BNEQ    LP1                             ; branch if locate mode worked
                                0BD4    632         FIELD   <RBF (INDICATES MOVE MODE)>
                                0BE9    633         MBPT
                   50  00 BE 9A 0BF1    634 LP1:    MOVZBL  @(SP),R0                        ; get expected size
                          08 12 0BF5    635         BNEQ    5$
                50  0200 8F 3C 0BF7    636         MOVZWL  #512,R0
                       000A 31 0BFC    637         BRW     10$
                      07 57 E8 0BFF    638 5$:     BLBS    R7,10$                          ; branch unless alternate
                   50  6E D0 0C02    639         MOVL    (SP),R0
                   50  01 A0 9A 0C05    640         MOVZBL  1(R0),R0                        ; get alternate record size
                   22 A9  50 B1 0C09    641 10$:    CMPW    R0,RAB$W_RSZ(R9)                ; size right?
                          1D 13 0C0D    642         BEQL    LP2                             ; branch if right
                                0C0F    643         FIELD   <RSZ>
                                0C24    644         MBPT
28 B9   50  57  6E  00 2D 0C2C    645 LP2:    CMPC5   #0,(SP),R7,R0,@RAB$L_RBF(R9); record contents ok?
                          1D 13 0C33    646         BEQL    LP3                             ; branch if yes
                                0C35    647         FIELD   <RBF (BAD RECORD CONTENTS)>
                                0C4A    648         MBPT
                03  F3C9 CF 91 0C52    649 LP3:    CMPB    MTAFAB+FAB$B_RFM,#FAB$C_VFC
                          23 12 0C57    650         BNEQ    LP4                             ; branch if not vfc rfm
                   57  2C B9 D1 0C59    651         CMPL    @RAB$L_RHB(R9),R7               ; rhb contents ok?
                          1D 13 0C5D    652         BEQL    LP4
                                0C5F    653         FIELD   <RHB>
                                0C74    654         MBPT
         FF20 57  01  7A 8F 9D 0C7C    655 LP4:    ACBB    #^X7A,#1,R7,LOOP1
                      6E 02 C0 0C83    656         ADDL2   #2,(SP)                         ; bump past args
                                05 0C86    657         RSB
```

```
                        0C87    659
                        0C87    660 ;
                        0C87    661 ;  subroutine to check for at eof and close mt file
                        0C87    662 ;
                        0C87    663
                        0C87    664 EOF_CLOSE:
                 03  10 0C87    665          BSBB    CHKEOF
               FE36  31 0C89    666          BRW     CLOSE_MT
                        0C8C    667
                        0C8C    668
                        0C8C    669 ;
                        0C8C    670 ;  subroutine to check for at eof
                        0C8C    671 ;
                        0C8C    672
                        0C8C    673 CHKEOF: $GET    R9                          ; expect eof
                        0C95    674 CHKEOF1:
00000000'8F  50  D1     0C95    675          CMPL    R0,#RMS$_EOF
                 06  13 0C9C    676          BEQL    CK5
            5A   59  D0 0C9E    677          MOVL    R9,R10
               F35C' 30 0CA1    678          BSBW    EOFPUT                      ; give error mesg's
                     05 0CA4    679 CK5:     RSB
                        0CA5    680
                        0CA5    681
                        0CA5    682 ;
                        0CA5    683 ;  subroutine to check for error code = rms$_bof
                        0CA5    684 ;
                        0CA5    685
00000000'8F  50  D1     0CA5    686 CHKBOF: CMPL    R0,#RMS$_BOF
                 1E  13 0CAC    687          BEQL    CKBEND
            5A   59  D0 0CAE    688          MOVL    R9,R10
               F34C' 30 0CB1    689          BSBW    REPORT_ERR
                        0CB4    690          FIELD   <STS (EXPECTED RMS$_BOF)>
               F334' 30 0CC9    691          BSBW    ERR
                     05 0CCC    692 CKBEND: RSB
                        0CCD    693
                        0CCD    694
                        0CCD    695          .END
```

```
$$.PSECT_EP                     = 00000000
$$.TAB                          = 00000050 R   D   01
$$.TABEND                       = 00000094 R   D   01
$$.TMP                          = 00010000     D
$$.TMP1                         = 00000001     D
$$.TMP2                         = 00000059
$$.TMPX                         = 0000022A R   D   04
$$.TMPX1                        = 00000017     D
$$RMSTEST                       = 0000001E
$$RMS_PBUGCHK                   = 00000010
$$RMS_TBUGCHK                   = 00000008
$$RMS_UMODE                     = 00000004
..AFLG                          = 00000000     D
..FLG                           = 00000002     D
..MOD                           = 00000001     D
..N                             = 00000001
..TYP                           = 00000003     D
.LEN                            = 00000004     D
BEGPUT                            ********   X   01
BEG_DESCR                         ********   X   01
BUF                               00000294 R   D   01
CHECK_ONE                         00000A54 R   D   01
CHKBOF                            00000CA5 R   D   01
CHKEOF                            00000C8C R   D   01
CHKEOF1                           00000C95 R   D   01
CK5                               00000CA4 R   D   01
CKBEND                            00000CCC R   D   01
CLOSE_MT                          00000AC2 R   D   01
CMDORAB                           ********   X   01
CREATE_MT                         00000B0D R   D   01
DEV$V_FOR                       = 00000018     D
DEV$V_SQD                       = 00000005     D
DO_PUT                            00000AAB R   D   01
EOFPUT                            ********   X   01
EOF_CLOSE                         00000C87 R   D   01
ERR                               ********   X   01
FAB$B_DNS                       = 00000035     D
FAB$B_FAC                       = 00000016     D
FAB$B_FSZ                       = 0000003F     D
FAB$B_ORG                       = 0000001D     D
FAB$B_RAT                       = 0000001E     D
FAB$B_RFM                       = 0000001F     D
FAB$C_BID                       = 00000003     D
FAB$C_BLN                       = 00000050     D
FAB$C_FIX                       = 00000001     D
FAB$C_REL                       = 00000010     D
FAB$C_SEQ                       = 00000000     D
FAB$C_VAR                       = 00000002     D
FAB$C_VFC                       = 00000003     D
FAB$L_ALQ                       = 00000010     D
FAB$L_DEV                       = 00000040     D
FAB$L_DNA                       = 00000030     D
FAB$L_FNA                       = 0000002C     D
FAB$L_FOP                       = 00000004     D
FAB$L_STS                       = 00000008     D
FAB$M_BLK                       = 00000008     D
FAB$M_CR                        = 00000002     D
FAB$V_BLK                       = 00000003     D
FAB$V_BRO                       = 00000006     D
FAB$V_CHAN_MODE                 = 00000002     D
FAB$V_CIF                       = 00000019     D
FAB$V_CR                        = 00000001     D
FAB$V_FILE_MODE                 = 00000004     D
FAB$V_GET                       = 00000001     D
FAB$V_LNM_MODE                  = 00000000     D
FAB$V_NEF                       = 0000000A     D
FAB$V_POS                       = 00000008     D
FAB$V_PUT                       = 00000000     D
FAB$V_RWC                       = 0000000B     D
FAB$V_RWO                       = 00000007     D
FAB$V_SUP                       = 00000002     D
FAB$V_TRN                       = 00000004     D
FAB$W_BLS                       = 0000003C     D
FAB$W_GBC                       = 00000048     D
FAB$W_MRS                       = 00000036     D
FINPUT                            ********   X   01
FIN_DESCR                         ********   X   01
FLDPUT                            ********   X   01
FLD_DESCR                         ********   X   01
FNSIZE                          = 00000005     D
IOBSZ                           = 00000200   G D
IOBUF                             00000094  RG D   01
LOOP                              00000ADA  R   D   01
LOOP1                             00000BA3  R   D   01
LP1                               00000BF1  R   D   01
LP2                               00000C2C  R   D   01
LP3                               00000C52  R   D   01
LP4                               00000C7C  R   D   01
MTAFAB                            00000000  R   D   01
MTARAB                            00000050  R   D   01
MTFIX_FNM                         0000029D  R   D   01
MTST1_FNM                         000002A7  R   D   01
MTVAR_FNM                         00000298  R   D   01
MTVFC_FNM                         000002A2  R   D   01
NF10                              00000903  R   D   01
NF20                              00000927  R   D   01
NF30                              00000945  R   D   01
NF35                              0000094E  R   D   01
NF40                              000009D7  R   D   01
NF50                              00000A2E  R   D   01
NONFIL_FNM                        000002AC  R   D   01
NON_FICE                          0000083B  R   D   01
ONET0                             00000A8B  R   D   01
ONE20                             00000AAA  R   D   01
OPEN_MT                           00000B38  R   D   01
OPN1                              00000B78  R   D   01
OPN2                              00000B9A  R   D   01
PUTRECS                           00000AD7  R   D   01
RAB$B_RAC                       = 0000001E      D
RAB$C_BID                       = 00000001      D
RAB$C_BLN                       = 00000044      D
RAB$C_KEY                       = 00000001      D
RAB$C_SEQ                       = 00000000      D
RAB$L_BKT                       = 00000038      D
```

```
RAB$L_CTX                       = 00000018     D
RAB$L_FAB                       = 0000003C     D
RAB$L_KBF                       = 00000030     D
RAB$L_RBF                       = 00000028     D
RAB$L_RHB                       = 0000002C     D
RAB$L_ROP                       = 00000004     D
RAB$M_TPT                       = 00000002     D
RAB$V_LOC                       = 00000010     D
RAB$W_RSZ                       = 00000022     D
READ_MT                           00000BA0 R   D   01
READ_MT_VAR                       00000B9B R   D   01
REPORT_ERR                        ********     X   01
REPORT_ERROR                      ********     X   01
RHBBUF                            ********     X   01
RMS$_BOF                          ********     X   01
RMS$_CREATED                      ********     X   01
RMS$_EOF                          ********     X   01
RMS$_NORMAL                       ********     X   01
RMT1                              00000425 R   D   01
RMT11                             000006BB R   D   01
RMT12                             00000715 R   D   01
RMT2                              00000449 R   D   01
RMT3                              00000465 R   D   01
RMT4                              0000049E R   D   01
RMT5                              000004BC R   D   01
RMT6                              000004D7 R   D   01
RMT7                              0000050E R   D   01
RMT8                              00000532 R   D   01
RMT9                              0000054D R   D   01
STRMT                             0000031E R   D   01
SYS$CLOSE                         ********   GX     01
SYS$CONNECT                       ********   GX     01
SYS$CREATE                        ********   GX     01
SYS$DISCONNECT                    ********   GX     01
SYS$ERASE                         ********   GX     01
SYS$GET                           ********   GX     01
SYS$NXTVOL                        ********   GX     01
SYS$OPEN                          ********   GX     01
SYS$PUT                           ********   GX     01
SYS$REWIND                        ********   GX     01
SYS$SPACE                         ********   GX     01
SYS$TRUNCATE                      ********   GX     01
T6START                           00000000 RG  D   01
TST$MTA                           000002B1 RG  D   01
VERBOSITY                         ********     X   01
```

+----------------+
! Psect synopsis !
+----------------+

```
PSECT name                      Allocation                PSECT No.  Attributes
----------                      ----------                ---------  ----------

.  ABS  .                       00000000  (       0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
_RMSTEST                        00000CCD  (    3277.)  01 (  1.)  NOPIC   USR   CON   REL   GBL NOSHR   EXE   RD    WRT NOVEC LONG
$ABS$                           00000000  (       0.)  02 (  2.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
$RMSNAM                         0000000E  (      14.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
__$RMSNAM                       00000241  (     577.)  04 (  4.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
```

+-------------------------+
! Performance indicators !
+-------------------------+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.08	00:00:00.52
Command processing	105	00:00:00.56	00:00:03.36
Pass 1	431	00:00:25.02	00:00:54.87
Symbol table sort	0	00:00:00.56	00:00:01.23
Pass 2	141	00:00:04.74	00:00:12.35
Symbol table output	20	00:00:00.15	00:00:00.35
Psect synopsis output	2	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	730	00:00:31.15	00:01:12.72

The working set limit was 1500 pages.
115188 bytes (225 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 479 non-local and 17 local symbols.
695 source lines were read in Pass 1, producing 50 object records in Pass 2.
68 pages of virtual memory were used to define 50 macros.

+---------------------------+
! Macro library statistics !
+---------------------------+

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	0
_255DUA28:[SYSLIB]STARLET.MLB;2	37
TOTALS (all libraries)	37

945 GETS were required to define 37 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:RMSTEST6/OBJ=OBJ$:RMSTEST6 MSRC$:RMSTEST6/UPDATE=(ENH$:RMSTEST6)+EXECML$/LIB

0409 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RMSTEST4
LIS

RMSTEST7
LIS

RMSTEST8
LIS

RMSTESTR
LIS

RMSTEST0
LIS

RMSTEST2
LIS

RMSTEST3
LIS

RMSTEST5
LIS

RMSTEST6
LIS

RMSTESTT
LIS

RMSTEST1
LIS

SAT55F17
LIS