```
UUU          UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP
UUU          UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP
UUU          UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP
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         PPPPPPPPPPPP
UUU          UUU  EEEEEEEEEEEE           TTT         PPPPPPPPPPPP
UUU          UUU  EEEEEEEEEEEE           TTT         PPPPPPPPPPPP
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
UUU          UUU  EEE                    TTT         PPP
UUUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT         PPP
UUUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT         PPP
UUUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT         PPP
```

```
RRRRRRRR    MM      MM    SSSSSSSS  TTTTTTTTTT  EEEEEEEEEE    SSSSSSSS  TTTTTTTTTT    333333
RRRRRRRR    MM      MM    SSSSSSSS  TTTTTTTTTT  EEEEEEEEEE    SSSSSSSS  TTTTTTTTTT    333333
RR      RR  MMMM  MMMM  SS              TT      EE          SS              TT      33      33
RR      RR  MMMM  MMMM  SS              TT      EE          SS              TT      33      33
RR      RR  MM  MM  MM  SS              TT      EE          SS              TT              33
RR      RR  MM  MM  MM  SS              TT      EE          SS              TT              33
RRRRRRRR    MM      MM    SSSSSS        TT      EEEEEEEE      SSSSSS        TT            33
RRRRRRRR    MM      MM    SSSSSS        TT      EEEEEEEE      SSSSSS        TT            33
RR  RR      MM      MM          SS      TT      EE                  SS      TT              33
RR  RR      MM      MM          SS      TT      EE                  SS      TT              33
RR    RR    MM      MM          SS      TT      EE                  SS      TT      33      33      ....
RR    RR    MM      MM          SS      TT      EE                  SS      TT      33      33      ....
RR      RR  MM      MM  SSSSSSSS        TT      EEEEEEEEEE  SSSSSSSS        TT        333333        ....
RR      RR  MM      MM  SSSSSSSS        TT      EEEEEEEEEE  SSSSSSSS        TT        333333        ....

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          .IDENT  'V04-000'
0000    74          $BEGIN  RMSTEST3,009,__RMSTEST,<RELATIVE TEST PROGRAM>,<GBL,LONG>
0000    75
0000    76 ;
0000    77
0000    78          .ENABL  DBG
0000    79
0000    80 ;
0000    81 ; this program tests the relative file org
0000    82 ;
0000    83 ;
0000    84
0000    85 ;
0000    86 ;  macros:
0000    87 ;
0000    88
0000    89          .MACRO  BUFF NAM,SIZE
0000    90 NAM'BUF::
0000    91          .BLKB   SIZE
0000    92          NAM'BSZ==SIZE
0000    93          .ENDM   BUFF
0000    94
0000    95 ;
0000    96 ;
0000    97 ;
0000    98
0000    99          .MACRO  TYPE STRING, ?L
0000   100          STORE   <STRING>
0000   101          BLBC    VERBOSITY,L
0000   102          MOVL    #$$.TMPX,CMDORAB+RAB$L_RBF
0000   103          MOVW    #$$.TMPX1,CMDORAB+RAB$W_RSZ
0000   104          $PUT    RAB=CMDORAB,ERR=REPORT_ERROR
0000   105          BSBW    ERR
0000   106 L:
0000   107          .ENDM   TYPE
0000   108
0000   109 ;
0000   110
0000   111          .MACRO  WTTYPE STRING
0000   112          $WAIT   CMDORAB
0000   113          TYPE    <STRING>
0000   114          .ENDM   WTTYPE
0000   115
0000   116 ;
0000   117
0000   118          .MACRO  STORE STRING,PRE
0000   119          .SAVE
0000   120          .PSECT  __$RMSNAM
0000   121          $$.TMPX=.
0000   122          PRE                                     ; store any carriage control info
0000   123          .ASCII  %STRING%
0000   124          $$.TMPX1=.-$$.TMPX
0000   125          .RESTORE
0000   126          .ENDM   STORE
```

```
0000    128
0000    129 ;
0000    130
0000    131          .MACRO  FNM STRING
0000    132          STORE   <STRING>
0000    133          MOVB    #$$.TMPX1,FAB$B_FNS+RELATIVE_FAB
0000    134          MOVL    #$$.TMPX,FAB$L_FNA+RELATIVE_FAB
0000    135          .ENDM
0000    136
0000    137          .MACRO  BEGIN TSTNAM
0000    138          STORE   <TSTNAM>
0000    139          MOVL    #$$.TMPX,BEG_DESCR+4     ; addr
0000    140          MOVL    #$$.TMPX1,BEG_DESCR      ; len
0000    141          BSBW    BEGPUT
0000    142          .ENDM   BEGIN
0000    143          .MACRO  FINISH TSTNAM
0000    144          STORE   <TSTNAM>
0000    145          MOVL    #$$.TMPX,FIN_DESCR+4     ; addr
0000    146          MOVL    #$$.TMPX1,FIN_DESCR      ; len
0000    147          BSBW    FINPUT
0000    148          .ENDM   FINISH
0000    149          .MACRO  FIELD FLDNAM
0000    150          STORE   <FLDNAM>
0000    151          MOVL    #$$.TMPX,FLD_DESCR+4     ; addr
0000    152          MOVL    #$$.TMPX1,FLD_DESCR      ; len
0000    153          BSBW    FLDPUT
0000    154          .ENDM   FIELD
0000    155          .MACRO  MBPT ?L
0000    156          BLBC    VERBOSITY,L
0000    157          BPT
0000    158 L:
0000    159          .ENDM   MBPT
0000    160
0000    161 ;
0000    162
```

```
                                00000000   164          .PSECT   RMSTEST,GBL,LONG
                                    0000   165          .ALIGN   LONG
                                    0000   166 T3START::
                                    0000   167 RELATIVE_FAB:
                                    0000   168          $FAB     FAC=<GET,PUT,DEL,UPD>,-
                                    0000   169                   DNM=<TST$DISK:.FIL;1>,-
                                    0000   170                   NAM=NAMBLK,-
                                    0000   171                   FOP=<DFW,SUP>,-
                                    0000   172                   ORG=REL,-
                                    0000   173                   RAT=CR,-
                                    0000   174                   MRS=52,-
                                    0000   175                   MRN=500,-
                                    0000   176                   ALQ=0,-                 ; alq=0 forces extend
                                    0000   177                   FSZ=4
                                    0050   178
                                    0050   179 RELATIVE_RAB:
                                    0050   180          $RAB     FAB=RELATIVE_FAB,-
                                    0050   181                   UBF=RELBUF,-
                                    0050   182                   USZ=RELBSZ,-
                                    0050   183                   RBF=RELBUF,-
                                    0050   184                   RHB=HEAD,-
                                    0050   185                   MBF=3,-
                                    0050   186                   ROP=UIF
                                    0094   187 ALLOC_XAB:
                                    0094   188          $XABALL  AID=0,-
                                    0094   189                   DEQ=4,-
                                    0094   190                   ALQ=4
                                    00B4   191 COUNTER:
                                      00   00B4   192          .BYTE    0
                                      00   00B5   193 COUNT2:  .BYTE    0
                                00000000   00B6   194 HEAD:    .LONG    0
                                00000000   00BA   195 KEY:     .LONG    0
                                000000EE   00BE   196 RFATBL:  .BLKQ    6
                                           00EE   197          .ALIGN   LONG
                                           00F0   198          BUFF     REL,200
                        000001C0'00000029' 01B8   199 RFMSTR:  .LONG    RFML,RFMS
54 41 4C 45 52 20 44 45 4E 45 50 4F        01C0   200 RFMS:    .ASCII   'OPENED RELATIVE FILE WITH FILE ORG !AD !/'
54 49 57 20 45 4C 49 46 20 45 56 49        01CC
21 20 47 52 4F 20 45 4C 49 46 20 48        01D8
                  2F 21 20 44 41           01E4
                        00000029   01E9   201          RFML=.-RFMS
```

```
                         01E9   203 RMT$TEST_34::
                    0FFC 01E9   204         .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                         01EB   205         BEGIN   <RELATIVE TESTS>
                         0200   206
                         0200   207 ;
                         0200   208 ;VARIABLE
                         0200   209 ;
                         0200   210
                         0200   211         TYPE    <TEST WITH VARIABLE RECORD FORMAT>
                         022F   212         FNM     <RELVAR>
            FDDD CF    02   90  023D   213         MOVB    #FAB$C_VAR,FAB$B_RFM+RELATIVE_FAB
                  00E8    30   0242   214         BSBW    REL_TEST
                         0245   215
                         0245   216 ;
                         0245   217 ;VFC
                         0245   218 ;
                         0245   219
                         0245   220         TYPE    <TEST WITH VFC RECORD FORMAT>
                         0274   221         FNM     <RELVFC>
                         0232   222         $FAB_STORE      FAB=RELATIVE_FAB,-
                         0232   223                 RFM=VFC,-
                         0283   224                 FSZ=#4; need to set fsz
     04 A0    02000000 8F   C8  028F   225         BISL2   #FAB$M_CIF,FAB$L_FOP(R0)          ; do create-if
      04 A0    04   CA  0297   226         BICL2   #FAB$M_SUP,FAB$L_FOP(R0)          ; not supercede
                         029B   227         $CREATE FAB=RELATIVE_FAB,-
                         029B   228                 ERR=REPORT_ERROR
                   FD51'  30   02AC   229         BSBW    ERR
          00000000'8F    50   D1  02AF   230         CMPL    R0,#RMS$_CREATED                ; was it created
                    05   13  02B6   231         BEQL    10$
                    10   C8  02B8   232         BISL2   #RAB$M_UIF,-                    ; if opened, do updates
               FD97 CF         02BA   233                 RAB$L_ROP+RELATIVE_RAB
                  0140   30  02BD   234 10$:    BSBW    REL_TEST2
           02000000 8F   CA  02C0   235         BICL2   #FAB$M_CIF,-                    ; restore bits
               FD3B CF         02C6   236                 FAB$L_FOP+RELATIVE_FAB
                    04   C8  02C9   237         BISL2   #FAB$M_SUP,-
               FD36 CF         02CB   238                 FAB$L_FOP+RELATIVE_FAB
                         02CE   239
                         02CE   240 ;
                         02CE   241 ;fix
                         02CE   242 ;
                         02CE   243
                         02CE   244         TYPE    <TEST WITH FIXED LENGTH RECORDS>
                         02FD   245         FNM     <RELFIX>
          FDOF CF    01   90  030B   246         MOVB    #FAB$C_FIX,FAB$B_RFM+RELATIVE_FAB
                001A    30   0310   247         BSBW    REL_TEST
               FDOD CF    D4   0313   248         CLRL    FAB$L_XAB+RELATIVE_FAB          ; clear it for restart
                         0317   249         FINISH  <RELATIVE TESTS>
                    04   032C   250         RET
                         032D   251
```

```
                     032D   253 REL_TEST:
                     032D   254
                     032D   255 ;
                     032D   256 ;routine to put relative thru its paces, and call the locking routine
                     032D   257 ;
                     032D   258
                     032D   259         $FAB_STORE      FAB=RELATIVE_FAB,-
                     032D   260                 ALQ=#0                                  ; make sure it's 0
                     0335   261         $CREATE FAB=R0,-
                     0335   262                 ERR=REPORT_ERROR
          FCB9'  30  0344   263         BSBW    ERR
                     0347   264         $XABALL_STORE   XAB=ALLOC_XAB,-
                     0347   265                 ALQ=#4
                     0350   266         $FAB_STORE      FAB=RELATIVE_FAB,-
                     0350   267                 XAB=ALLOC_XAB
                     035B   268         $EXTEND FAB=R0,-
                     035B   269                 ERR=REPORT_ERROR                        ; extend 4 blks, from alq
          FC93'  30  036A   270         BSBW    ERR
04   FC9B CF   D1  036D   271         CMPL    FAB$L_STV+RELATIVE_FAB,#4               ; check returned stv
          15   18  0372   272         BGEQ    STVOK
                     0374   273         FIELD   <STV ( NOT = ALLOC QTY, AFTER EXTEND)>
04   FD17 CF   D1  0389   274 STVOK:   CMPL    XAB$L_ALQ+ALLOC_XAB,#4                  ; check alq
          15   18  038E   275         BGEQ    ALQOK
                     0390   276         FIELD   <ALQ IN XAB ( NOT = ALLOC QTY, AFTER EXTEND)>
                     03A5   277 ALQOK:  $CLOSE  FAB=RELATIVE_FAB,-
                     03A5   278                 ERR=REPORT_ERROR
          FC47'  30  03B6   279         BSBW    ERR
                     03B9   280         $XABALL_STORE   XAB=ALLOC_XAB,-
                     03B9   281                 ALQ=#0
                     03C1   282         $OPEN   FAB=RELATIVE_FAB,-
                     03C1   283                 ERR=REPORT_ERROR
          FC2B'  30  03D2   284         BSBW    ERR
05   FCCB CF   D1  03D5   285         CMPL    ALLOC_XAB+XAB$L_ALQ,#5                  ; alq=4 from extend + 1 since rel
          15   18  03DA   286         BGEQ    ALQOK1
                     03DC   287         FIELD   <ALQ IN XAB ( NOT = ALLOC QTY, AFTER OPEN)>
01   FC2A CF   91  03F1   288 ALQOK1: CMPB    RELATIVE_FAB+FAB$B_RFM,#FAB$C_FIX
          08   13  03F6   289         BEQL    REL_TEST2                               ; on w/ it
                     03F8   290         $FAB_STORE      FAB=RELATIVE_FAB,-              ; if var clear xab
                     03F8   291                 XAB=#0
```

```
                                0400   293 REL_TEST2:
                                C 00   294
                                0400   295 ;
                                0400   296 ;entry point to bypass create
                                0400   297 ;
                                0400   298
                                0400   299         $CONNECT        RAB=RELATIVE_RAB,-
                                0400   300                         ERR=REPORT_ERROR
                         FBEC'  30      0411   301         BSBW    ERR
                                0414   302
                                0414   303 ;
                                0414   304 ;do 26 sequential puts with 'bad' data into file
                                0414   305 ;then delete 5 of those records and do puts to put right data into them
                                0414   306 ;update the other records and then do gets to make sure it's all ok
                                0414   307 ;
                                0414   308
                      59   01   D0      0414   309         MOVL    #1,R9                   ; r9 is record number
                5B    FC35 CF   DE      0417   310         MOVAL   RELATIVE_RAB,R11        ; r11 is address of rab
                                041C   311         TYPE    <FILL FILE>
                                044B   312 PUT_RECORD_SEQ::
                      56   0F   D0      044B   313         MOVL    #15,R6                  ; all of length 15
                FBCC  CF   01   91      044E   314         CMPB    #FAB$C_FIX,FAB$B_RFM+RELATIVE_FAB
                           03   12      0453   315         BNEQ    30$
                      56   34   D0      0455   316         MOVL    #52,R6                  ; len of fixed rec. is 52
   FC90 CF    56   31   6E   00   2C    0458   317 30$:     MOVC5   #0,(SP),#^A/1/,R6,RELBUF; fill buffer for output
                                0460   318                                             ; put ascii '1's into records
                   22 AB   56   B0      0460   319         MOVW    R6,RAB$W_RSZ(R11)       ; give size of record
                FC4D  CF   59   D0      0464   320         MOVL    R9,HEAD                 ; fill header in case it's vfc
                                0469   321
                                0469   322 ;
                                0469   323 ;seq mode is default
                                0469   324 ;
                                0469   325
                                0469   326         $PUT    RAB=R11,-
                                0469   327                 ERR=REPORT_ERROR
                      FB85'  30      0478   328         BSBW    ERR
                59    38 AB   D1      047B   329         CMPL    RAB$L_BKT(R11),R9       ; bkt should be rec. # on output
                           15   13      047F   330         BEQL    BKT_OK
                                0481   331         FIELD   <BKT IN RAB (RECORD NUMBER)>
                B1 59      1A   F3      0496   332 BKT_OK: AOBLEQ  #26,R9,PUT_RECORD_SEQ   ; keep going?
                                049A   333
                                049A   334 ;
                                049A   335 ;check one record just to be sure
                                049A   336 ;
                                049A   337
                      59   0F   D0      049A   338         MOVL    #15,R9
          30 AB     FC19 CF   DE      049D   339         MOVAL   KEY,RAB$L_KBF(R11)
             1E AB      01   90      04A3   340         MOVB    #RAB$C_KEY,RAB$B_RAC(R11)
          FC0E CF       59   D0      04A7   341         MOVL    R9,KEY
                                04AC   342         $GET    RAB=R11,ERR=REPORT_ERROR
                      FB42'  30      04BB   343         BSBW    ERR
                      01B7   30      04BE   344         BSBW    CHK_BAD_DATA
             04 AB      10   CA      04C1   345         BICL2   #RAB$M_OIF,RAB$L_ROP(R11)   ; in case its set
                59      05   D0      04C5   346         MOVL    #5,R9
                                04C8   347 FIND_DEL:
                                04C8   348
                                04C8   349 ;
```

```
                        04C8    350 ;do some finds and deletes by keyed access
                        04C8    351 ;
                        04C8    352
FBED CF    59    D0     04C8    353          MOVL    R9,KEY
                        04CD    354          $FIND   RAB=R11,-
                        04CD    355                  ERR=REPORT_ERROR
        FB21'   30      04DC    356          BSBW    ERR
                        04DF    357          $DELETE RAB=R11,-
                        04DF    358                  ERR=REPORT_ERROR
        FB0F'   30      04EE    359          BSBW    ERR
    59    05    C0      04F1    360          ADDL    #5,R9
    1A    59    D1      04F4    361          CMPL    R9,#26
          CF    19      04F7    362          BLSS    FIND_DEL
```

```
                                 04F9   364
                                 04F9   365 ;
                                 04F9   366 ;done with deletes, now 'put' into the deleted records correct data
                                 04F9   367 ;
                                 04F9   368
                  59   05   D0  04F9   369          MOVL    #5,R9
                                 04FC   370 PUT_RECORD_KEY:
                                 04FC   371
                                 04FC   372 ;
                                 04FC   373 ;try to get deleted records, hoping it fails
                                 04FC   374 ;then set nxr and get the deleted records and check them
                                 04FC   375 ;finally put the corrected (previously deleted)records
                                 04FC   376 ;
                                 04FC   377
      00C8 8F   00   6E   00   2C  04FC   378          MOVC5   #0,(SP),#0,#200,RELBUF  ; clr relbuf, to make sure
                FBEA CF          0503
              FBAF CF   59   D0  0506   379          MOVL    R9,KEY                  ; gets by key
                                 050B   380          $GET    RAB=R11                 ; hope it fails
          00000000'8F   50   D1  0514   381          CMPL    R0,#RMS$_RNF            ; record not found?
                        1E   13  051B   382          BEQL    ERR_OK
                                 051D   383          FIELD   <RETURNED ERROR CODE>
                  5A   5B   D0  0532   384          MOVL    R11,R10
                     FAC8' 30  0535   385          BSBW    REPORT_ERR
                     FAC5' 30  0538   386          BSBW    ERR
               00800000 8F   C8  053B   387 ERR_OK: BISL    #RAB$M_NXR,-            ; get non-existant record
                  FB10 CF          0541   388                  RAB$L_ROP+RELATIVE_RAB
                                 0544   389          $GET    RAB=RT1,-               ; this should work
                                 0544   390                  ERR=REPORT_ERROR
                     FAAA' 30  0553   391          BSBW    ERR
                     011F  30  0556   392          BSBW    CHK_BAD_DATA            ; check it out
               00800000 8F   CA  0559   393          BICL    #RAB$M_NXR,-            ; clear bit
                  FAF2 CF          055F   394                  RAB$L_ROP+RELATIVE_RAB
                     0222  30  0562   395          BSBW    SETUP                   ; fills buffer,rsz,head
            FB50 CF   59   D0  0565   396          MOVL    R9,KEY                  ; keyed access
                                 056A   397          $PUT    RAB=R11,-
                                 056A   398                  ERR=REPORT_ERROR
                     FA84' 30  0579   399          BSBW    ERR
                  59   05   C0  057C   400          ADDL    #5,R9
                  1A   59   D1  057F   401          CMPL    R9,#26
                       03   18  0582   402          BGEQ    10$
                     FF75 31   0584   403          BRW     PUT_RECORD_KEY
                                 0587   404 10$:
                                 0587   405
                                 0587   406 ;
                                 0587   407 ;all done with that
                                 0587   408 ;
                                 0587   409
                  59   01   D0  0587   410          MOVL    #1,R9
                                 058A   411
                                 058A   412 ;
                                 058A   413 ;do updates on all other records, by keyed access
                                 058A   414 ;
                                 058A   415
                                 058A   416 UPDATE_RECORD:
                       5A   D4   058A   417          CLRL    R10
      56   52   59   05   7B   058C   418          EDIV    #5,R9,R2,R6
                       56   D5   0591   419          TSTL    R6                      ; is it 5,10,15,20,25
```

```
                07   12  0593   420          BNEQ    20$
          F1 59  1A   F3  0595   421          AOBLEQ  #26,R9,UPDATE_RECORD    ; if so, skip it
               0030   31  0599   422          BRW     NO_MORE                 ; all done
        FB19 CF  59   D0  059C   423 20$:     MOVL    R9,KEY
                      05A1   424           $FIND   RAB=R11,-
                      05A1   425                   ERR=REPORT_ERROR
               FA4D'  30  05B0   426          BSBW    ERR
               01D1   30  05B3   427          BSBW    SETUP                   ; set up for put
                      05B6   428           $UPDATE RAB=R11,-
                      05B6   429                   ERR=REPORT_ERROR
               FA38'  30  05C5   430          BSBW    ERR
          BE 59  1A   F3  05C8   431          AOBLEQ  #26,R9,UPDATE_RECORD
                      05CC   432 NO_MORE:
        FA9D CF  00   90  05CC   433          MOVB    #RAB$C_SEQ,RAB$B_RAC+RELATIVE_RAB
                      05D1   434          TYPE    <VERIFY CONTENTS OF FILE>
                      0600   435          $REWIND RAB=R11,-               ; can now do gets
                      0600   436                   ERR=REPORT_ERROR
               F9EE'  30  060F   437          BSBW    ERR
               00AA   30  0612   438          BSBW    DO_SOME_GETS
                      0615   439          $DISCONNECT     RAB=R11,-
                      0615   440                   ERR=REPORT_ERROR
               F9D9'  30  0624   441          BSBW    ERR
                      0627   442          $CONNECT        RAB=R11,-
                      0627   443                   ERR=REPORT_ERROR
               F9C7'  30  0636   444          BSBW    ERR
               0083   30  0639   445          BSBW    DO_SOME_GETS
                      063C   446          $DISCONNECT     RAB=R11,-
                      063C   447                   ERR=REPORT_ERROR
               F9B2'  30  064B   448          BSBW    ERR
                      064E   449          $CLOSE  FAB=RELATIVE_FAB,-
                      064E   450                   ERR=REPORT_ERROR
               F99E'  30  065F   451          BSBW    ERR
                      0662   452                                           ; tell locking test which file
            F9CE CF   90  0662   453          MOVB    FAB$B_FNS+RELATIVE_FAB,-
        00000034'EF           0666   454                  FAB$B_FNS+LOCK_FAB
            F9BD CF   D0  066B   455          MOVL    FAB$L_FNA+RELATIVE_FAB,-
        0000002C'EF           066F   456                  FAB$L_FNA+LOCK_FAB
                      0674   457                                           ; do locking tests
               F989'  30  0674   458          BSBW    RMT$TEST_5A
                      05  0677   459          RSB
                      0678   460
                      0678   461 ;
                      0678   462 ;subroutine to check 'bad' data ( 1st pass of puts)
                      0678   463 ;r11 is pointer to relative_rab
                      0678   464 ;routine checks rsz and contents of record, now in buffer
                      0678   465 ;
                      0678   466
                      0678   467 CHK_BAD_DATA:
          56    0F    D0  0678   468          MOVL    #15,R6                  ; len of non-fixed records
                01    91  067B   469          CMPB    #FAB$C_FIX,-
            F99F CF         067D   470                  RELATIVE_FAB+FAB$B_RFM
                03    12  0680   471          BNEQ    10$                     ; if fix len is 52
          56    34    D0  0682   472          MOVL    #52,R6
       22 AB    56    B1  0685   473 10$:     CMPW    R6,RAB$W_RSZ(R11)       ; check rsz
                15    13  0689   474          BEQL    RSZ_OK
                      068B   475          FIELD   <RSZ>
                      06A0   476 RSZ_OK:
```

```
                                06A0    477          CMPC5   R6,@RAB$L_RBF(R11),#^A/1/,#0,(SP)
                        15      06A7    478          BEQL    REC_OK
                                06A9    479          FIELD   <RECORD>
                                06BE    480 REC_OK:
                        05      06BE    481          RSB
```

```
                     06BF    483 DO_SOME_GETS::
54   F9FB CF    DE   06BF    484          MOVAL   RFATBL,R4                       ; r4 is index to rfatbl
     F9EC CF    94   06C4    485          CLRB    COUNTER                         ; record number
     F9E9 CF    94   06C8    486          CLRB    COUNT2
     57   1A    D0   06CC    487          MOVL    #26,R7                          ; 1st pass-r7 is # of rec.
5B   F97D CF    DE   06CF    488          MOVAL   RELATIVE_RAB,R11                ; pointer to rab
  1E AB   00    90   06D4    489          MOVB    #RAB$C_SEQ,RAB$B_RAC(R11)       ; do all sequential gets
        00E1    30   06D8    490          BSBW    GET_RECORD_SEQ
  1E AB   02    90   06DB    491          MOVB    #RAB$C_RFA,RAB$B_RAC(R11)       ; do some gets by rfa
54   F9DB CF    DE   06DF    492          MOVAL   RFATBL,R4
        0180    30   06E4    493          BSBW    GET_RECORD_RFA
  1E AB   01    90   06E7    494          MOVB    #RAB$C_KEY,RAB$B_RAC(R11)       ; do some gets by key
        01A2    30   06EB    495          BSBW    GET_RECORD_KEY
  1E AB   00    90   06EE    496          MOVB    #RAB$C_SEQ,RAB$B_RAC(R11)       ; do some seq gets
     57   0D    D0   06F2    497          MOVL    #13,R7                          ; 2nd pass-r7 is # of rec.
54   F9DD CF    DE   06F5    498          MOVAL   RFATBL+24,R4                    ; 24=8.*3, starting in the middle
F9B5 CF   0D    90   06FA    499          MOVB    #13,COUNTER                     ;               ditto
     F9B2 CF    94   06FF    500          CLRB    COUNT2
        00B6    30   0703    501          BSBW    GET_RECORD_SEQ
                05   0706    502          RSB
```

```
                                0707   504
                                0707   505 ;
                                0707   506 ; subroutines to do gets and checks
                                0707   507 ;
                                0707   508
                                0707   509 CHECK_REC:
                                0707   510
                                0707   511 ;
                                0707   512 ;       r9 is the record number
                                0707   513 ;       as before r11 is the addr of the rab
                                0707   514 ;
                                0707   515
          F913 CF    03   91    0707   516          CMPB    #FAB$C_VFC,FAB$B_RFM+RELATIVE_FAB
                       1B   12  070C   517          BNEQ    NO_HEADER
            2C BB      59   D1  070E   518 100$:    CMPL    R9,@RAB$L_RHB(R11)      ; compare header
                       15   13  0712   519          BEQL    NO_HEADER
                                0714   520          FIELD   <HEADER OF A VFC RECORD>
                                0729   521 NO_HEADER:
                1A     59   D1  0729   522          CMPL    R9,#26                  ; is it last record?
                       05   12  072C   523          BNEQ    10$
                56     1A   D0  072E   524          MOVL    #26,R6                  ; length is 26
                       07   11  0731   525          BRB     20$
                       5A   D4  0733   526 10$:     CLRL    R10                     ; quadword divide
      56    52     59   1A   7B  0735   527          EDIV    #26,R9,R2,R6            ; r6 = r9 mod 26, rec. size
55    56     00000040 8F   C1  073A   528 20$:     ADDL3   #^A/A/-1,R6,R5          ; r5 is char.
          F8D8 CF    01   91    0742   529          CMPB    #FAB$C_FIX,FAB$B_RFM+RELATIVE_FAB
                       03   12  0747   530          BNEQ    GOT_RS
                56     34   D0  0749   531          MOVL    #52,R6                  ; mrs for fixed is 26
                                074C   532 GOT_RS:
            22 AB      56   B1  074C   533          CMPW    R6,RAB$W_RSZ(R11)       ; check rsz
                       15   13  0750   534          BEQL    OK_RSZ
                                0752   535          FIELD   <RSZ FIELD IN RAB>
                                0767   536 OK_RSZ:
6E    00    55    28 BB      56   2D  0767   537 10$:     CMPC5   R6,@RAB$L_RBF(R11),R5,#0,(SP)
                       01   12  076E   538          BNEQ    BADREC
                            05  0770   539          RSB                             ; return
                                0771   540 BADREC:
                                0771   541          FIELD   <RECORD>
                            05  0786   542          RSB
```

```
                                0787   544 SETUP:
                                0787   545
                                0787   546 ;
                                0787   547 ;routine to do setup for puts of correct data
                                0787   548 ;r9 is record number -- input
                                0787   549 ;output -- relbuf is filled in with correct char
                                0787   550 ;        -- head is filled in, and rsz is also
                                0787   551 ;
                                0787   552
                     1A   59    D1  0787   553          CMPL    R9,#26
                          05    12  078A   554          BNEQ    10$
                     56   1A    D0  078C   555          MOVL    #26,R6                  ; len of last rec. is 26
                          07    11  078F   556          BRB     20$
                          5A    D4  0791   557 10$:      CLRL    R10
              56     52   59    1A  7B  0793   558          EDIV    #26,R9,R2,R6            ; r6 is rec. # mod 26
        55    56    00000040 8F    C1  0798   559 20$:      ADDL3   #^A/A/-1,R6,R5          ; r5 is char. to fill buffer
                          01    91  07A0   560          CMPB    #FAB$C_FIX,-
                     F87A CF         07A2   561                  FAB$B_RFM+RELATIVE_FAB
                          03    12  07A5   562          BNEQ    30$
                     56   34    D0  07A7   563          MOVL    #52,R6                  ; len. of fixed rec. is 52
  F93E CF    56     55   6E    00  2C  07AA   564 30$:      MOVC5   #0,(SP),R5,R6,RELBUF    ; fill relbuf
                  22 AB   56    B0  07B2   565          MOVW    R6,RAB$W_RSZ(R11)       ; fill rsz
                F8FB CF   59    D0  07B6   566          MOVL    R9,HEAD                 ; fill header, in case vfc
                          05    07  07BB   567          RSB
```

```
                        07BC    569
                        07BC    570 GET_RECORD_SEQ:
                        07BC    571
                        07BC    572 ;
                        07BC    573 ;                       routine to sequentially get and check records
                        07BC    574 ;
                        07BC    575
                        07BC    576         $GET    R11
00000000'8F   50   D1 07C5    577         CMPL    R0,#RMS$_EOF
                OF  12 07CC    578         BNEQ    MORE
    57  F8E3 CF   91 07CE    579         CMPB    COUNT2,R7
                01  12 07D3    580         BNEQ    BADNR
                    05 07D5    581         RSB
                       07D6    582 BADNR:
        5A   5B   D0 07D6    583         MOVL    R11,R10
           F824' 30 07D9    584         BSBW    EOFPUT
                    05 07DC    585         RSB
                       07DD    586 MORE:
          09 50   E8 07DD    587         BLBS    R0,10$
        5A   5B   D0 07E0    588         MOVL    R11,R10
           F81A' 30 07E3    589         BSBW    REPORT_ERR
           F817' 30 07E6    590         BSBW    ERR
    57  F8C8 CF   91 07E9    591 10$:    CMPB    COUNT2,R7
                02  15 07EE    592         BLEQ    20$
                E4  11 07F0    593         BRB     BADNR
        F8BE CF   96 07F2    594 20$:    INCB    COUNTER
        F8BB CF   96 07F6    595         INCB    COUNT2
    59  F8B6 CF   9A 07FA    596         MOVZBL  COUNTER,R9
      38 AB   59   D1 07FF    597         CMPL    R9,RAB$L_BKT(R11)
                15  13 0803    598         BEQL    RNOK
                       0805    599         FIELD   <BKT FIELD IN RAB>
                       081A    600 RNOK:
           FEEA 30   81A    601         BSBW    CHECK_REC
             5A D4   81D    602         CLRL    R10                     ; quad word divide
52  50   59   05 7B 081F    603         EDIV    #5,R9,R0,R2
         01   52 D1 0824    604         CMPL    R2,#1
              03 13 0827    605         BEQL    SAV_RFA
           FF90 31 0829    606         BRW     GET_RECORD_SEQ          ; continue
                       082C    607 SAV_RFA:
                       082C    608
                       082C    609 ;
                       082C    610 ; save record numbers 1,6,11,16,21,26 on 1st pass
                       082C    611 ; check record numbers 16,21,26 on 2nd pass
                       082C    612 ;
                       082C    613
          2A 57   E9 082C    614         BLBC    R7,SAV                  ; which pass?
  64   10 AB   06 29 082F    615         CMPC3   #6,RAB$W_RFA(R11),(R4)  ; 2nd, check them
                1D 13 0834    616         BEQL    RFA_OK
                       0836    617         FIELD   <RFA>
                       084B    618         MBPT
                       0853    619 RFA_OK:
        54   08   C0 0853    620         ADDL2   #8,R4
           FF63 31 0856    621         BRW     GET_RECORD_SEQ          ; on to next record
                       0859    622 SAV:
     84   10 AB   7D 0859    623         MOVQ    RAB$W_RFA(R11),(R4)+    ; 1st pass, save them
  52 54   01   C3 085D    624         SUBL3   #1,R4,R2
        62   59 90 0861    625         MOVB    R9,(R2)                 ; also store record number
```

```
                FF55   31  0864    626          BRW     GET_RECORD_SEQ
                       0867    627
                       0867    628 GET_RECORD_RFA:
                       0867    629
                       0867    630 ;
                       0867    631 ;       get records by rfa
                       0867    632 ;
                       0867    633
000000EE'8F      54    D1  0867    634          CMPL    R4,#RFATBL+48           ; more rfa entries
                  1F    18  086E    635          BGEQ    END_OF_RFA
      10 AB       84    7D  0870    636          MOVQ    (R4)+,RAB$W_RFA(R11)    ; load rab w/ rfa
                        0874    637          $GET    RAB=R11,ERR=REPORT_ERROR
            F77A'     30  0883    638          BSBW    ERR
      59    17 AB     90  0886    639          MOVB    RAB$W_RFA+7(R11),R9     ; get record number, as stored
            FE7A      30  088A    640          BSBW    CHECK_REC               ; r9 is now rec. #
              D8      11  088D    641          BRB     GET_RECORD_RFA
                      088F    642 END_OF_RFA:
                  05  088F    643          RSB
                      0890    644
                      0890    645 GET_RECORD_KEY:
                      0890    646
                      0890    647 ;
                      0890    648 ;get records by key
                      0890    649 ;
                      0890    650
   30 AB    F826 CF    DE  0890    651          MOVAL   KEY,RAB$L_KBF(R11)
      F81F CF   01    D0  0896    652          MOVL    #1,KEY                  ; get 1st record
                0010  30  089B    653          BSBW    GETCHK                  ; get and check
      F817 CF   1A    D0  089E    654          MOVL    #26,KEY
                0008  30  08A3    655          BSBW    GETCHK
      F80F CF   0D    D0  08A6    656          MOVL    #13,KEY
                0000  31  08AB    657          BRW     GETCHK
                      08AE    658
                      08AE    659 GETCHK:
                      08AE    660          $GET    RAB=R11,-
                      08AE    661                  ERR=REPORT_ERROR
            F740'     30  08BD    662          BSBW    ERR
59      F7F6 CF       D0  08C0    663          MOVL    KEY,R9
            FE3F      31  08C5    664          BRW     CHECK_REC
                      08C8    665          .END
```

```
$$.PSECT_EP                     = 00000000
$$.TAB                          = 00000094 R   D  01
$$.TABEND                       = 000000B4 R   D  01
$$.TMP                          = 00000002     D
$$.TMP1                         = 00000002     D
$$.TMP2                         = 0000005B
$$.TMPX                         = 00000191 R   D  04
$$.TMPX1                        = 00000003     D
$$RMSTEST                       = 0000001E
$$RMS_PBUGCHK                   = 00000010
$$RMS_TBUGCHK                   = 00000008
$$RMS_UMODE                     = 00000004
..AFLG                          = 00000000     D
..FLG                           = 00000001     D
..MOD                           = 00000001     D
..TYP                           = 000000CF
.LEN                            = 00000004     D
ALLOC_XAB                       00000094 R   D  01
ALQOK                           000003A5 R   D  01
ALQOK1                          000003F1 R   D  01
BADNR                           000007D6 R   D  01
BADREC                          00000771 R   D  01
BEGPUT                          ********    X   01
BEG_DESCR                       ********    X   01
BKT_OK                          00000496 R   D  01
CHECK_REC                       00000707 R   D  01
CHK_BAD_DATA                    00000678 R   D  01
CMDORAB                         ********    X   01
COUNT2                          000000B5 R   D  01
COUNTER                         000000B4 R   D  01
DO_SOME_GETS                    000006BF RG  D  01
END_OF_RFA                      0000088F R   D  01
EOFPUT                          ********    X   01
ERR                             ********    X   01
ERR_OK                          0000053B R   D  01
FAB$B_DNS                       = 00000035     D
FAB$B_FNS                       = 00000034     D
FAB$B_FSZ                       = 0000003F     D
FAB$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$C_VAR                       = 00000002     D
FAB$C_VFC                       = 00000003     D
FAB$L_ALQ                       = 00000010     D
FAB$L_DNA                       = 00000030     D
FAB$L_FNA                       = 0000002C     D
FAB$L_FOP                       = 00000004     D
FAB$L_STV                       = 0000000C     D
FAB$L_XAB                       = 00000024     D
FAB$M_CIF                       = 02000000     D
FAB$M_SUP                       = 00000004     D
FAB$V_CHAN_MODE                 = 00000002     D
FAB$V_CR                        = 00000001     D
FAB$V_DEL                       = 00000002     D
FAB$V_DFW                       = 00000005     D
FAB$V_FILE_MODE                 = 00000004     D
FAB$V_GET                       = 00000001     D
FAB$V_LNM_MODE                  = 00000000     D
FAB$V_PUT                       = 00000000     D
FAB$V_SUP                       = 00000002     D
FAB$V_UPD                       = 00000003     D
FAB$W_GBC                       = 00000048     D
FIND_DEL                        000004C8 R   D  01
FINPUT                          ********    X   01
FIN_DESCR                       ********    X   01
FLDPUT                          ********    X   01
FLD_DESCR                       ********    X   01
GETCHK                          000008AE R   D  01
GET_RECORD_KEY                  00000890 R   D  01
GET_RECORD_RFA                  00000867 R   D  01
GET_RECORD_SEQ                  000007BC R   D  01
GOT_RS                          0000074C R   D  01
HEAD                            000000B6 R   D  01
KEY                             000000BA R   D  01
LOCK_FAB                        ********    X   01
MORE                            000007DD R   D  01
NAMBLK                          ********    X   01
NO_HEADER                       00000729 R   D  01
NO_MORE                         000005CC R   D  01
OK_RSZ                          00000767 R   D  01
PUT_RECORD_KEY                  000004FC R   D  01
PUT_RECORD_SEQ                  0000044B RG  D  01
RAB$B_RAC                       = 0000001E     D
RAB$C_BID                       = 00000001     D
RAB$C_BLN                       = 00000044     D
RAB$C_KEY                       = 00000001     D
RAB$C_RFA                       = 00000002     D
RAB$C_SEQ                       = 00000000     D
RAB$L_BKT                       = 00000038     D
RAB$L_CTX                       = 00000018     D
RAB$L_KBF                       = 00000030     D
RAB$L_RBF                       = 00000028     D
RAB$L_RHB                       = 0000002C     D
RAB$L_ROP                       = 00000004     D
RAB$M_NXR                       = 00800000     D
RAB$M_UIF                       = 00000010     D
RAB$V_UIF                       = 00000004     D
RAB$W_RFA                       = 00000010     D
RAB$W_RSZ                       = 00000022     D
REC_OK                          000006BE R   D  01
RELATIVE_FAB                    00000000 R   D  01
RELATIVE_RAB                    00000050 R   D  01
RELBSZ                          = 000000C8   G D
RELBUF                          000000F0 RG  D  01
REL_TEST                        0000032D R   D  01
REL_TEST2                       00000400 R   D  01
REPORT_ERR                      ********    X   01
REPORT_ERROR                    ********    X   01
RFATBL                          000000BE R   D  01
RFA_OK                          00000853 R   D  01
RFMC                            = 00000029     D
RFMS                            000001C0 R   D  01
```

```
RFMSTR                          000001B8 R   D  01
RMS$_CREATED                    ******** X    01
RMS$_EOF                        ******** X    01
RMS$_RNF                        ******** X    01
RMT$TEST_3A                     000001E9 RG  D  01
RMT$TEST_5A                     ******** X    01
RNOK                            0000081A R   D  01
RSZ_OK                          000006A0 R   D  01
SAV                             00000859 R   D  01
SAV_RFA                         0000082C R   D  01
SETUP                           00000787 R   D  01
STVOK                           00000389 R   D  01
SYS$CLOSE                       ******** GX   01
SYS$CONNECT                     ******** GX   01
SYS$CREATE                      ******** GX   01
SYS$DELETE                      ******** GX   01
SYS$DISCONNECT                  ******** GX   01
SYS$EXTEND                      ******** GX   01
SYS$FIND                        ******** GX   01
SYS$GET                         ******** GX   01
SYS$OPEN                        ******** GX   01
SYS$PUT                         ******** GX   01
SYS$REWIND                      ******** GX   01
SYS$UPDATE                      ******** GX   01
T3START                         00000000 RG  D  01
UPDATE_RECORD                   0000058A R   D  01
VERBOSITY                       ******** X    01
XAB$B_AID                     = 00000017     D
XAB$B_AOP                     = 00000008     D
XAB$B_BKZ                     = 00000016     D
XAB$C_ALL                     = 00000014     D
XAB$C_ALLLEN                  = 00000020     D
XAB$L_ALQ                     = 00000010     D
XAB$L_LOC                     = 0000000C     D
XAB$L_NXT                     = 00000004     D
XAB$W_DEQ                     = 00000014     D
XAB$W_RFI0                    = 00000018     D
XAB$W_RFI2                    = 0000001A     D
XAB$W_RFI4                    = 0000001C     D
XAB$W_VOL                     = 0000000A     D
```

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	000008C8 (2248.)	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
$RMSNAM	0000000F (15.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
__$RMSNAM	00000194 (404.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	38	00:00:00.09	00:00:00.38
Command processing	133	00:00:00.60	00:00:03.73
Pass 1	299	00:00:11.36	00:00:25.25
Symbol table sort	0	00:00:00.55	00:00:01.11
Pass 2	118	00:00:02.85	00:00:06.66
Symbol table output	19	00:00:00.12	00:00:00.16
Psect synopsis output	2	00:00:00.04	00:00:00.04
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	611	00:00:15.61	00:00:37.33

The working set limit was 1500 pages.
54514 bytes (107 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 441 non-local and 20 local symbols.
665 source lines were read in Pass 1, producing 46 object records in Pass 2.
64 pages of virtual memory were used to define 49 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	0
_255DUA28:[SYSLIB]STARLET.MLB;2	35
TOTALS (all libraries)	35

870 GETS were required to define 35 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:RMSTEST3/OBJ=OBJ$:RMSTEST3 MSRC$:RMSTEST3/UPDATE=(ENH$:RMSTEST3)+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