

UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTTTT	PPPPPPPPPPPPPP	
UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTTTT	PPPPPPPPPPPPPP	
UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTTTT	PPPPPPPPPPPPPP	
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	EEE	TTT	PPP	PPP
UUU	UUU	EEEEEEEEEEEEEEEE	TTT	PPPPPPPPPPPPPP	
UUU	UUU	EEEEEEEEEEEEEEEE	TTT	PPPPPPPPPPPPPP	
UUU	UUU	EEEEEEEEEEEEEEEE	TTT	PPPPPPPPPPPPPP	
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	UUUUUUUUUUUUUUUU	EEEEEEEEEEEEEEEE	TTT	PPP	
UUUUUUUUUUUUUUUU	UUUUUUUUUUUUUUUU	EEEEEEEEEEEEEEEE	TTT	PPP	
UUUUUUUUUUUUUUUU	UUUUUUUUUUUUUUUU	EEEEEEEEEEEEEEEE	TTT	PPP	


```
RRRRRRRR  MM      MM  SSSSSSSS  TTTTTTTTTT  EEEEEEEEE  SSSSSSSS  TTTTTTTTTT  8888888  8888888
RRRRRRRR  MM      MM  SSSSSSSS  TTTTTTTTTT  EEEEEEEEE  SSSSSSSS  TTTTTTTTTT  8888888  8888888
RR      RR  MMMM  MMMM  SS      TT      EE      SS      TT      88      88
RR      RR  MMMM  MMMM  SS      TT      EE      SS      TT      88      88
RR      RR  MM  MM  MM  SS      TT      EE      SS      TT      88      88
RR      RR  MM  MM  MM  SS      TT      EE      SS      TT      88      88
RRRRRRRR  MM      MM  SSSSSS  TT      EE      SSSSSS  TT      888888  888888
RRRRRRRR  MM      MM  SSSSSS  TT      EE      SSSSSS  TT      888888  888888
RR      RR  MM      MM      SS      TT      EE      SS      TT      88      88
RR      RR  MM      MM      SS      TT      EE      SS      TT      88      88
RR      RR  MM      MM      SS      TT      EE      SS      TT      88      88
RR      RR  MM      MM      SSSSSSSS  TT      EEEEEEEEE  SSSSSSSS  TT      888888  888888
RR      RR  MM      MM      SSSSSSSS  TT      EEEEEEEEE  SSSSSSSS  TT      888888  888888

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 72      .EXTERNAL  REPORT_ERROR,BEG_DESCR,BEGPUT,FIN_DESCR,FINPUT,FLD_DESCR,FLD
0000 73      $BEGIN  RMSTEST8,002,--RMSTEST,<INDEXED FILE TEST PROGRAM>,<GBL, LONG>
0000 74
0000 75      :
0000 76      : Macros...
0000 77      :
0000 78
0000 79      .MACRO  STORE STRING,PRE
0000 80      .SAVE
0000 81      .PSECT  --$RMSNAM
0000 82      $$TMPX=;
0000 83      PRE
0000 84      .ASCII  %STRING%                                ; store any carriage control info
0000 85      $$TMPX1=.-$$TMPX
0000 86      .RESTORE
0000 87      .ENDM  STORE
0000 88
0000 89      .MACRO  BEGIN TSTNAM
0000 90      STORE  <TSTNAM>
0000 91      MOVL  #$$TMPX,BEG_DESCR+4      ; addr
0000 92      MOVL  #$$TMPX1,BEG_DESCR      ; len
0000 93      BSBW  BEGPOT
0000 94      .ENDM  BEGIN
0000 95
0000 96      .MACRO  FINISH TSTNAM
0000 97      STORE  <TSTNAM>
0000 98      MOVL  #$$TMPX,FIN_DESCR+4      ; addr
0000 99      MOVL  #$$TMPX1,FIN_DESCR      ; len
0000 100     BSBW  FINPUT
0000 101     .ENDM  FINISH
0000 102
0000 103     .MACRO  FIELD FLDNAM
0000 104     STORE  <FLDNAM>
0000 105     MOVL  #$$TMPX,FLD_DESCR+4      ; addr
0000 106     MOVL  #$$TMPX1,FLD_DESCR      ; len
0000 107     BSBW  FLDPOT
0000 108     .ENDM  FIELD
0000 109
0000 110     .MACRO  MBPT ?L                                ; bpt, maybe?
0000 111     BLBC  VERBOSITY,L
0000 112     BPT
0000 113     L:
0000 114     .ENDM  MBPT
0000 115
0000 116     :
0000 117     : IXVAR1 -- program to create and populate the rms-11 file
0000 118     : idxvr1.dat ; records are variable length
0000 119     : keys cannot change; dupes are fixed
0000 120     :
0000 121
0000 122
0000 123     .ENABL  DBG
0000 124     .NLIST  MEB
0000 125     $RMSDEF
0000 126
0000 127
0000 128 ;
```



```
0000 129 ; buffer area
0000 130 ;
0000 131 ;
00000020 0000 132 DELBUF: .BLKB 32. ; store deleted record numbers
0038 0020 133 BASLEN: .WORD 56. ; record 0 base length (changes)
0001 0022 134 PSTEP: .WORD 1 ; pseudo-step buffer
00000624 0024 135 ;
0024 136 RFABUF: .BLKW 768. ; rfa storage, in rec # order
0624 137 ;
0624 138 ;
0624 139 ; Step and first have to be such that recnum is not duplicated in 255 puts
0624 140 ;
0624 141 ;
FF 0624 142 STEP: .BYTE 255. ; set byte by odt to desired step val
00 0625 143 .BYTE 0 ; key to get 1st record on find random
FF 0626 144 ;
0626 145 FIRST: .BYTE 255. ; set byte by odt to desired step val
00 0627 146 ;
00 0627 147 COUNT: .BYTE 0
00 0628 148 LAST: .BYTE 0
00000000 0629 149 OPERATION:
0629 150 .LONG
062D 151 ;
062D 152 ;
062D 153 ; this is the record buffer:
062D 154 ;
062D 155 ;
0000062F 062D 156 MASBUF: .BLKB 2 ; rec # byte, dup control byte
4C 4B 4A 49 48 47 46 45 44 43 42 41 062F 157 .ASCII /ABCDEFGHIJKLMNOPQRSTUVWXYZ/
58 57 56 55 54 53 52 51 50 4F 4E 4D 063B
4A 49 48 47 46 45 44 43 42 41 5A 59 0647
56 55 54 53 52 51 50 4F 4E 4D 4C 4B 0653
5A 59 58 57 065F
0663 158 ;
0663 159 RECNUM: .REPT 131. ; rec # byte, nbytes fill, check byte,
0663 160 .WORD 0 ; 5 bytes for rfa update, 255.-n bytes
0000 0663 161 .ENDM ; left over (n= rec#)
0769 162 ;
0769 163 ;
0769 164 ; structure area
0769 165 ;
0769 166 ;
0769 167 ;
00000000 168 ;
0000 169 FAB1: .PSECT DATA, LONG
0000 170 $FAB FAC=<PUT,GET,UPD,DEL>,-
0000 171 FNM=<TST$DISK:IDXVAR1.DAT>,-
0000 172 ORG=IDX,-
0000 173 XAB=XAB0,-
0000 174 ALQ=300,-
0000 175 DEQ=100,-
0000 176 NAM=NAMBLK,-
0000 177 FOP=<NAM>,-
0050 178 BKS=1
0050 179 NAMBLK: $NAM
00B0 180
00B0 181 RAB1: $RAB RAC=KEY,-
```



```
00B0 182 RBF=MASBUF,-
00B0 183 UBF=MASBUF,-
00B0 184 USZ=311,-
00B0 185 KBF=STEP+1,-
00B0 186 KSZ=1,-
00B0 187 ROP=<KGE>,-
00B0 188 MBF=10,-
00B0 189 FAB=FAB1
00F4 190 :
00F4 191 : key is a,a-z(2),a,a,a,a,a-z(2),a-z(2),rec#
00F4 192 : therefore they are ordered by recnum
00F4 193 :
00F4 194 :
00F4 195 XAB0: $XABKEY NXT=XAB1,-
00F4 196 POS=<2,2,2,2,2,2,2,2>,-
00F4 197 SIZ=<1,52,1,1,1,1,52,53>,-
00F4 198 REF=0
0140 199 :
0140 200 :
0140 201 : key is a,a-z(2),a,a,a,a,a-z(2),a-z(2),rec#
0140 202 : therefore they are ordered by recnum
0140 203 :
0140 204 :
0140 205 XAB1: $XABKEY NXT=XAB2,-
0140 206 POS=<2,2,2,2,2,2,2,2>,-
0140 207 SIZ=<1,52,1,1,1,1,52,53>,-
0140 208 REF=1
018C 209 :
018C 210 :
018C 211 : key is a-z(2),a-z(2),a-z(2),a,a,a,a-b,a
018C 212 : therefore recnums appear in order of input
018C 213 :
018C 214 :
018C 215 XAB2: $XABKEY NXT=XAB3,-
018C 216 POS=<2,2,2,2,2,2,2,2>,-
018C 217 SIZ=<52,52,52,1,1,1,2,1>,-
018C 218 REF=2
01D8 219 :
01D8 220 :
01D8 221 : key is a,a,a,a-z(2),a-z(2),a-z(2),a-b,(127,128)
01D8 222 : therefore mostly dups, but not all
01D8 223 :
01D8 224 :
01D8 225 XAB3: $XABKEY NXT=XAB4,-
01D8 226 POS=<2,2,2,2,2,2,2,1>,-
01D8 227 SIZ=<1,1,1,52,52,52,2,1>,-
01D8 228 REF=3
0224 229 :
0224 230 :
0224 231 : key is a-a,a-z(2),recnum
0224 232 : so is ordered by recnum
0224 233 :
0224 234 :
0224 235 XAB4: $XABKEY NXT=XAB5,-
0224 236 POS=<2,2>,-
0224 237 SIZ=<27,53>,-
0224 238 REF=4
```



```
0270 239
0270 240 :
0270 241 : key is rec# only
0270 242 :
0270 243
0270 244 XAB5: $XABKEY NXT=XAB6,-
0270 245 POS=<0>,-
0270 246 SIZ=<1>,-
0270 247 REF=5
02BC 248
02BC 249 :
02BC 250 : key is a
02BC 251 : therefore records are in input order
02BC 252 :
02BC 253
02BC 254 XAB6: $XABKEY NXT=XAB7,-
02BC 255 POS=<2>,-
02BC 256 SIZ=<1>,-
02BC 257 REF=6
0308 258
0308 259 :
0308 260 : key is (127,128)
0308 261 : mostly all dups
0308 262
0308 263 XAB7: $XABKEY POS=<1>,-
0308 264 SIZ=<1>,-
0308 265 REF=7
0354 266
0354 267
0354 268 :
0354 269 : begin code
0354 270 :
0354 271
00000769 272 .PSECT __RMSTEST, LONG
0769 273
0769 274 RMT$INDEX::
0769 275
0769 276 START: .WORD 0
5A 00000000'EF 9E 076B 277 MOVAB FAB1,R10 ; address of fab
FE9E CF D4 0772 278 BEGIN <INDEXED TEST>
F863' 30 0787 279 CLRL OPERATION ; indicate create operation
078B 280 $CREATE FAB=R10,ERR=REPORT_ERROR; do create
079A 281 BSBW ERR
079D 282
5A 000000B0'EF 9E 079D 283 1$: MOVAB RAB1,R10
FE81 CF D6 07A4 284 INCL OPERATION ; indicate connect operation
07A8 285 $CONNECT RAB=R10,-
07A8 286 ERR=REPORT_ERROR ; do connect
F846' 30 07B7 287 BSBW ERR
07BA 288
FE6E CF 7F 8F 90 07BA 289 2$: MOVAB #127,MASBUF+1 ; set up for inserts
53 FE62 CF 9A 07C0 290 MOVZBL FIRST,R3 ; start record #
07C5 291
53 FFFFFFF00 8F CA 07C5 292 PNEXT: BICL2 #^X<FFFFFFF00>,R3 ; insert the records
FE5C CF 53 90 07CC 293 MOVAB R3,MASBUF
FE8D CF 53 90 07D1 294 MOVAB R3,RECNUM
00000664'E3 53 90 07D6 295 MOVAB R3,L^RECNUM+1(R3)
```



```
22 AA F83B CF AO 07DD 296 $RAB_STORE RAB=R10,-
FE3D CF 02 9A 07DD 297 RSZ=R3
F802' 30 07E1 298 ADDW2 BASLEN,RAB$W,RSZ(R10)
50 06 9A 07E7 299 MOVZBL #2,OPERATION ; indicate put
50 53 C4 07EC 300 $PUT RAB=R10,ERR=REPORT_ERROR
10 AA D0 07FB 301 BSBW ERR
00000024'E0 14 AA B0 07FE 302 MOVZBL #6,R0
00000028'E0 00000028'E0 0801 303 MULL2 R3,R0
FE13 CF FE16 CF 92 0804 304 MOVL RAB$L,RFA0(R10),-
00000664'E3 94 0807 305 L^RFA0(R10) ; move rfa of record
50 FDFF CF 9A 080C 306 MOVW RAB$W,RFA4(R10),-
FDF9 CF 04 80 080F 307 L^RFA0(R10)+4(R0) ; 6 char long
FDF2 CF 53 50 80 0814 308 MCOMB MASBUF+1,MASBUF+1
00000A59'EF 16 80 081B 309 CLRB L^RECNUM+1(R3)
FDE5 CF D6 0821 310 MOVZBL STEP,R0
F7AA' 30 0826 311 ADDL2 R0,R3
53 FDCA CF 9A 0829 312 ADDB2 #4,COUNT
54 53 06 C5 082E 313 BNEQ PNEXT
00000024'8F C0 0830 314 SUBB3 R0,R3,LAST ; move last key put into last
F7B1 CF 3D B0 0836 315 JSB CHKALL ; check all the indexes
FDB5 CF 04 9A 083C 316 $RAB_STORE RAB=R10,-
F77A' 30 083C 317 RAC=KEY
FDA3 CF 9A 0840 318 INCL OPERATION
53 56 D1 0844 319 $FIND RAB=R10,ERR=REPORT_ERROR
15 13 0853 321 BSBW ERR
50 FDBC CF DE 0856 322 $RAB_STORE RAB=R10,-
50 53 C0 0856 323 RAC=SEQ
80 84 D0 085A 324 MOVZBL LAST,R3
80 84 90 085F 325 MULL3 #6,R3,R4
22 AA 05 AO 0863 326 ADDL2 #RFA0(R10),R4
FD6B CF 05 9A 086A 327 MOVW #61.,BASLEN
F730' 30 086F 328 UNEXT: MOVZBL #4,OPERATION
FD91 CF DE 0874 329 $GET RAB=R10,ERR=REPORT_ERROR
50 53 C0 0883 331 BSBW ERR
FF AO 94 0886 332 MOVZBL MASBUF,R6
60 10 D1 088B 333 CMPL R6,R3
15 13 088E 334 BEQL DAT_OK
0890 335 FIELD <DATA>
08A5 336 DAT_OK: MOVAL RECNUM+2,R0
50 53 C0 08A5 337 ADDL2 R3,R0 ; position to end of buffer
80 84 D0 08AA 338 MOVL (R4)+,(R0)+ ; move rfa in for update
80 84 90 08AD 339 MOVW (R4)+,(R0)+
22 AA 05 AO 08B0 340 INCL R4
FD6B CF 05 9A 08B3 341 ADDW2 #5,RAB$W,RSZ(R10)
F730' 30 08B5 342 MOVZBL #5,OPERATION
FD91 CF DE 08B9 343 $UPDATE RAB=R10,ERR=REPORT_ERROR
50 53 C0 08BE 344 BSBW ERR
FF AO 94 08CD 345 MOVAL RECNUM+2,R0 ; position to end of buffer
60 10 D1 08D0 346 ADDL2 R3,R0
15 13 08D5 347 CLRB -1(R0)
08DB 348 CMPL RAB$L,RFA0(R10),(R0) ; check rfa after update
08DF 349 BEQL RFA_OK
08E1 350 FIELD <DATA (RFA) AFTER UPDATE>
08F6 352
```



```
60 14 80 D4 08F6 353 RFA_OK: CLRL (R0)+
AA 91 08F8 354 CMPB RAB$W RFA4(R10),(R0)
15 13 08FC 355 BEQL RFA_OK1
08FE 356 FIELD <DATA (RFA) AFTER UPDATE>
0913 357 RFA_OK1:
60 94 0913 358 CLRB (R0)
53 96 0915 359 INCB R3 ; r3 started as last key put
03 13 0917 360 ; and they increase by 1
FF53 31 0917 361 BEQL 10$
0919 362 BRW UNEXT
091C 363 10$: $RAB_STORE RAB=R10,-
091C 364 USZ=#316.
F6FB CF 01 B0 0922 365 MOVW #1,PSTEP
00000A59'EF 16 0927 366 JSB CHKALL
092D 367
092D 368 : test delete
092D 369 :
092D 370 :
092D 371 :
53 FCF6 CF 94 092D 372 CLRB COUNT
FCF3 CF 9A 0931 373 MOVZBL LAST,R3
0936 374 $RAB_STORE RAB=R10,RAC=RFA
093A 375
FCEA CF 04 9A 093A 376 DNEXT: MOVZBL #4,OPERATION
50 06 9A 093F 377 MOVZBL #6,R0
50 53 C4 0942 378 MULL2 R3,R0
00000024'E0 D0 0945 379 MOVL L^RFABUF(R0),-
10 AA 00000028'E0 B0 094B 380 RAB$W RFA0(R10) ; move rfa into rab for get
094D 381 MOVW L^RFABUF+4(R0),RAB$W RFA4(R10)
0955 382 $GET RAB=R10,ERR=REPORT_ERROR; get the record, locking it
F699' 30 0964 383 BSBW ERR
50 FCBC CF 9A 0967 384 MOVZBL COUNT,R0
00000000'E0 FCBD CF 90 096C 385 MOVB MASBUF,L^DELBUF(R0) ; store away its rec number
FCAF CF 06 9A 0975 386 MOVZBL #6,OPERATION
097A 387 $DELETE RAB=R10,ERR=REPORT_ERROR
F674' 30 0989 388 BSBW ERR
FC97 CF 96 098C 389 INCB COUNT
53 08 80 0990 390 ADDB2 #8,R3
A5 12 0993 391 BNEQ DNEXT ; deleted 32 records (every 8th)
0995 392
0995 393 :
0995 394 : check deleted records no longer appear anywhere
0995 395 :
0995 396 :
35 AA 07 90 0995 397 MOVB #7,RAB$B_KRF(R10) ; do all 8 keys
0999 398 $RAB_STORE RAB=R10,RAC=SEQ
099D 399
FC87 CF 08 9A 099D 400 KNEXT: MOVZBL #8,OPERATION
F64C' 30 09A2 401 $REWIND RAB=R10,ERR=REPORT_ERROR; rewind on new key of reference
53 38 D0 09B1 402 BSBW ERR
FC6D CF 04 9A 09B4 403 MOVL #56,R3 ; get all the records sequentially
09B7 404 MOVZBL #4,OPERATION
09BC 405
09BC 406 GNEXTD: $GET RAB=R10,ERR=REPORT_ERROR
56 F632' 30 09CB 407 BSBW ERR
50 F62E CF DE 09CE 408 MOVAL DELBUF,R6
50 56 08 C1 09D3 409 ADDL3 #8,R6,R0
```



```
FC51 CF 86 91 09D7 410 CHK_DEL:
      15 12 09D7 411      CMPB      (R6)+,MASBUF      ; make sure it wasn't deleted
      50 56 D1 09DC 412      BNEQ      DEL_OK
      C1 53 1F 09DE 413      FIELD      <DATA (RECORD WAS DELETED)>
      50 56 D1 09F3 414 DEL_OK:  CMPL      R6,R0
      C1 53 1F 09F6 415      BLSSU      CHK_DEL
      0001827A 8F 50 D1 09F8 416      SOBGTR      R3,GNEXTD
      OB 13 09FB 417      $GET      RAB=R10      ; verify end of file
      F5F0' 30 0A04 418      CMPL      R0,#RMSS_EOF
      0A0B 419      BEQL      EOF
      0A0D 420      BSBW      EOFPUT
      0A10 421      MBPT
      0A18 422
      0A18 423 EOF:
      35 AA 97 0A18 424      DECB      RAB$B_KRF(R10)
      80 18 0A1B 425      BGEQ      KNEXT
      FC07 CF 07 9A 0A1D 426      MOVZBL      #7,OPERATION      ; indicate close
      5A 00000000'EF DE 0A22 427      MOVAL      FAB1,R10
      04 AA 01008000 8F D0 0A29 428      MOVL      #<FAB$M_NAM!FAB$M_DLT>,FAB$L_FOP(R10)
      F5BD' 30 0A31 429      $CLOSE      FAB=R10,ERR=REPORT_ERROR
      0A40 430      BSBW      ERR
      0A43 431      FINISH      <INDEXED TEST>
      04 0A58 432      RET
      0A59 433
      0A59 434
      0A59 435 ;
      0A59 436 ; subroutines
      0A59 437 ;
      0A59 438 ;
      0A59 439 ;
      0A59 440 ; this routine checks the indices in various ways
      0A59 441 ;
      0A59 442 ;
      53 FBCB CF 9A 0A59 443 CHKALL: MOVZBL      LAST,R3      ; set up to check index on key 0
      0093 30 0A5E 444      BSBW      CHKIDX      ; and do so
      53 FBC3 CF 9A 0A61 445      MOVZBL      LAST,R3      ; set up to check index on key 1
      35 AA 96 0A66 446      INCB      RAB$B_KRF(R10)
      0088 30 0A69 447      BSBW      CHKIDX
      53 FBB8 CF 9A 0A6C 448      MOVZBL      LAST,R3      ; set up to check index on key 4
      007C 30 0A71 449      $RAB_STORE      RAB=R10,-
      FBAC CF 9A 0A75 450      0A71 450      KRF=#4
      35 AA 96 0A78 451      BSBW      CHKIDX      ; and do so
      0071 30 0A7D 452      MOVZBL      LAST,R3      ; set up to check index on key 5
      F598 CF FB9D CF 9B 0A80 453      INCB      RAB$B_KRF(R10)
      53 FB98 CF 9A 0A83 454      BSBW      CHKIDX      ; and do so
      35 AA 96 0A8A 455      MOVZBL      STEP,PSTEP      ; set up to check index on key 6
      005F 30 0A8F 456      MOVZBL      FIRST,R3
      53 FB8D CF 9A 0A92 457      INCB      RAB$B_KRF(R10)
      0A95 458      BSBW      CHKIDX      ; and do so
      0A9A 459      MOVZBL      FIRST,R3      ; set up to check index on key 2
      0A9A 460      $RAB_STORE      RAB=R10,-
      0A9E 461      0A9A 461      KRF=#2
      50 54 10 0A9E 462      BSBW      CHKIDX      ; and do so
      50 02 3C 0AA0 463      MOVZBL      #2,R0
      F57A CF 50 A4 0AA3 464      MULW2      R0,PSTEP
      53 FB7A CF 9A 0AA8 465      MOVZBL      FIRST,R3      ; start at beginning
      35 AA 96 0AAD 466      INCB      RAB$B_KRF(R10)      ; index =3
```



```
FB71 CF 80 8F 90 OAB0 467 MOVB #128.,COUNT ; check half
          3C 10 OAB6 468 BSBB CHKIDX
          53 FB6A CF 9A OAB8 469 MOVZBL FIRST,R3 ; setup to check second half
          53 FB63 CF 80 OABD 470 ADDB2 STEP,R3
FB5F CF 80 8F 90 OAC2 471 MOVB #128.,COUNT
          46 10 OAC8 472 BSBB GNEXT ; cont where left off
          53 FB58 CF 9A OACA 473 MOVZBL FIRST,R3
          OACF 474 $RAB_STORE RAB=R10,-
          OACF 475 KRF=#7
FB4E CF 80 8F 90 OAD3 476 MOVB #128.,COUNT
          19 10 OAD9 477 BSBB CHKIDX
          53 FB47 CF 9A OADB 478 MOVZBL FIRST,R3
          56 FB40 CF 9A OAE0 479 MOVZBL STEP,R6
          53 56 C0 OAE5 480 ADDL2 R6,R3
FB39 CF 80 8F 90 OAE8 481 MOVB #128.,COUNT
          20 10 OAE9 482 BSBB GNEXT
          35 AA 94 OAF0 483 CLRB RAB$B_KRF(R10)
          05 OAF3 484 RSB
          OAF4 485
          OAF4 486 :
          OAF4 487 : this routine checks one index
          OAF4 488 : input: r3 - start record number
          OAF4 489 :
          OAF4 490
          OAF4 491 CHKIDX: $RAB_STORE RAB=R10,-
          OAF4 492 RAC=KEY
          FB2C CF 03 9A OAF8 493 MOVZBL #3,OPERATION ; indicate find function
          OAFD 494 $FIND RAB=R10,ERR=REPORT_ERROR
          OB0C 495 $RAB_STORE RAB=R10,-
          OB0C 496 RAC=SEQ
          OB10 497
          53 FFFFFFF00 8F CA OB10 498 GNEXT: BICL2 #^X<FFFFFF00>,R3
          FB0D CF 04 9A OB17 499 MOVZBL #4,OPERATION ; indicate get function
          56 FAFE CF 9A OB2B 500 $GET RAB=R10,ERR=REPORT_ERROR
          53 56 91 OB30 501 MOVZBL MASBUF,R6
          15 13 OB33 502 CMPB R6,R3
          OB35 503 BEQL D_OK
          OB4A 504 FIELD <DATA>
          505
          53 00000664'E3 91 OB4A 506 D_OK: CMPB L^RECNUM+1(R3),R3
          15 13 OB51 507 BEQL D_OK1
          508
          50 22 AA F4B4 CF A3 OB68 509 D_OK1: SUBW3 BASLEN,RAB$W_RSZ(R10),R0; calc rec # from record length
          53 50 B1 OB6F 510 CMPW R0,R3
          15 13 OB72 511 BEQL RSZ
          OB74 512 FIELD <RSZ>
          OB89 513
          50 06 9A OB89 514 RSZ: MOVZBL #6,R0
          50 53 C4 OB8C 515 MULL2 R3,R0
          00000024'E0 10 AA D1 OB8F 516 CMPL RAB$L_RFA0(R10),L^RFABUF(R0)
          15 13 OB97 517 BEQL RFA_OK100
          OB99 518 FIELD <RFA>
          519
          00000028'E0 14 AA B1 OBAE 520 RFA_OK100:
          15 13 OBAE 521 CMPW RAB$W_RFA4(R10),L^RFABUF+4(R0)
          OB86 522 BEQL RFA_OK101
          OB88 523 FIELD <RFA>
```



```

56  F451 CF 3C  OBCD 524 RFA_OK101:
53  56 C0  OBCD 525      MOVZWL PSTEP,R6
FA4D CF 04 80  OBD2 526      ADDL2  R6,R3
      03 13  OBD5 527      ADDB2  #4,COUNT
      FF31 31  OBDA 528      BEQL   10$
      05 05  OBDC 529      BRW    GNEXT
      OBD 530 10$: RSB
      OBE0 531
      OBE0 532
      OBE0 533 : this is the error subroutine
      OBE0 534 : operation - 0 create
      OBE0 535 : 1 connect
      OBE0 536 : 2 put
      OBE0 537 : 3 find
      OBE0 538 : 4 get
      OBE0 539 : 5 update
      OBE0 540 : 6 delete
      OBE0 541 : 7 close
      OBE0 542 : 8 rewind
      OBE0 543 :
      OBE0 544
      OBE0 545      .END

```


Variable	Value	Mode	Unit
\$\$DTPTMP	= 00000000		D
\$\$PSECT_EP	= 00000000		
\$\$SPTMP	= 00000000		D
\$\$STMP	= 00000001		D
\$\$TAB	= 00000308	R	D 03
\$\$TABEND	= 00000354	R	D 03
\$\$TMP	= 00000000		D
\$\$TMP1	= 00000002		D
\$\$TMP2	= 0000005A		
\$\$TMPX	= 00000071	R	D 05
\$\$TMPX1	= 00000003		D
\$\$RMSTEST	= 0000001E		
\$\$RMS_PBUGCHK	= 00000010		
\$\$RMS_TBUGCHK	= 00000008		
\$\$RMS_UMODE	= 00000004		
..AFLG	= 00000000		D
..FLG	= 00000002		D
..MOD	= 00000001		D
..TYP	= 00000003		
.LEN	= 00000001		D
BASLEN	00000020	R	D 01
BEGPUT	*****	X	00
BEG_DESCR	*****	X	00
CHKALL	00000A59	R	D 01
CHKIDX	00000AF4	R	D 01
CHK_DEL	000009D7	R	D 01
COUNT	00000627	R	D 01
DAT_OK	000008A5	R	D 01
DELBUF	00000000	R	D 01
DEL_OK	000009F3	R	D 01
DNEXT	0000093A	R	D 01
D_OK	00000B4A	R	D 01
D_OK1	00000B68	R	D 01
EOF	00000A18	R	D 01
EOFPUT	*****	X	00
ERR	*****	X	00
FAB\$B_FNS	= 00000034		D
FAB\$C_BID	= 00000003		D
FAB\$C_BLN	= 00000050		D
FAB\$C_IDX	= 00000020		D
FAB\$C_VAR	= 00000002		D
FAB\$L_ALQ	= 00000010		D
FAB\$L_FNA	= 0000002C		D
FAB\$L_FOP	= 00000004		D
FAB\$M_DLT	= 00008000		D
FAB\$M_NAM	= 01000000		D
FAB\$V_CHAN_MODE	= 00000002		D
FAB\$V_DEL	= 00000002		D
FAB\$V_FILE_MODE	= 00000004		D
FAB\$V_GET	= 00000001		D
FAB\$V_LNM_MODE	= 00000000		D
FAB\$V_NAM	= 00000018		D
FAB\$V_PUT	= 00000000		D
FAB\$V_UPD	= 00000003		D
FAB\$W_GBC	= 00000048		D
FAB1	00000000	R	D 03
FINPUT	*****	X	00

FIN DESCR
FIRST
FLDPUT
FLD DESCR
GNEXT
GNEXTD
KNEXT
LAST
MASBUF
NAM\$B-ESS
NAM\$B-NOP
NAM\$B-RSS
NAM\$C-BID
NAM\$C-BLN
NAM\$L-ESA
NAM\$L-RSA
NAMBLK
OPERATION
PNEXT
PSTEP
RAB\$B-KRF
RAB\$B-RAC
RAB\$C-BID
RAB\$C-BLN
RAB\$C-KEY
RAB\$C-RFA
RAB\$C-SEQ
RAB\$L-CTX
RAB\$L-RFA0
RAB\$L-ROP
RAB\$V-KGE
RAB\$W-RFA4
RAB\$W-RS2
RAB\$W-US2
RAB1
RECNUM
REPORT-ERROR
RFABUF
RFA-OK
RFA-OK1
RFA-OK100
RFA-OK101
RMS\$ EOF
RMT\$INDEX
RS2
START
STEP
SYSS\$CLOSE
SYSS\$CONNECT
SYSS\$CREATE
SYSS\$DELETE
SYSS\$FIND
SYSS\$GET
SYSS\$PUT
SYSS\$REWIND
SYSS\$UPDATE
UNEXT

*****		X	00
00000626	R	D	01
*****		X	00
*****		X	00
00000B10	R	D	01
0000098C	R	D	01
0000099D	R	D	01
00000628	R	D	01
0000062D	R	D	01
= 0000000A		D	
= 00000008		D	
= 00000002		D	
= 00000002		D	
= 00000060		D	
= 0000000C		D	
= 00000004		D	
00000050	R	D	03
00000629	R	D	01
000007C5	R	D	01
00000022	R	D	01
= 00000035		D	
= 0000001E		D	
= 00000001		D	
= 00000044		D	
= 00000001		D	
= 00000002		D	
= 00000000		D	
= 00000018		D	
= 00000010		D	
= 00000004		D	
= 00000015		D	
= 00000014		D	
= 00000022		D	
= 00000020		D	
000000B0	R	D	03
00000663	R	D	01
*****		X	00
00000024	R	D	01
000008F6	R	D	01
00000913	R	D	01
00000BAE	R	D	01
00000BCD	R	D	01
= 0001827A		D	
00000769	RG	D	01
00000B89	R	D	01
00000769	R	D	01
00000624	R	D	01
*****	GX		01
*****	GX		01
*****	GX		01
*****	GX		01
*****	GX		01
*****	GX		01
*****	GX		01
*****	GX		01
0000086F	R	D	01

RMSTEST8
Symbol table

INDEXED FILE TEST PROGRAM ;

N 14

16-SEP-1984 01:51:38 VAX/VMS Macro V04-00
5-SEP-1984 04:22:11 [UETP.SRC]RMSTEST8.MAR;1

Page 11
(1)

```
VERBOSITY          ***** X 00
XAB$B_DAN          = 0000000A D
XAB$B_DTP          = 00000013 D
XAB$B_FLG          = 00000012 D
XAB$B_IAN          = 00000008 D
XAB$B_LAN          = 00000009 D
XAB$B_NUL          = 00000015 D
XAB$B_PROLOG       = 00000048 D
XAB$B_REF          = 00000017 D
XAB$B_SIZ0         = 0000002E D
XAB$B_SIZ1         = 0000002F D
XAB$B_SIZ2         = 00000030 D
XAB$B_SIZ3         = 00000031 D
XAB$B_SIZ4         = 00000032 D
XAB$B_SIZ5         = 00000033 D
XAB$B_SIZ6         = 00000034 D
XAB$B_SIZ7         = 00000035 D
XAB$C_KEY          = 00000015 D
XAB$C_KEYLEN       = 0000004C D
XAB$C_STG          = 00000000 D
XAB$C_KNM          = 00000038 D
XAB$C_NXT          = 00000004 D
XAB$V_CHG          = 00000001 D
XAB$V_DUP          = 00000000 D
XAB$W_DFL          = 0000001C D
XAB$W_IFL          = 0000001A D
XAB$W_POS0         = 0000001E D
XAB$W_POS1         = 00000020 D
XAB$W_POS2         = 00000022 D
XAB$W_POS3         = 00000024 D
XAB$W_POS4         = 00000026 D
XAB$W_POS5         = 00000028 D
XAB$W_POS6         = 0000002A D
XAB$W_POS7         = 0000002C D
XAB0               000000F4 R D 03
XAB1               00000140 R D 03
XAB2               0000018C R D 03
XAB3               000001D8 R D 03
XAB4               00000224 R D 03
XAB5               00000270 R D 03
XAB6               000002BC R D 03
XAB7               00000308 R D 03
```

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

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
RMSTEST	00000BE0 (3040.)	01 (1.)	NOPIC USR CON REL GBL NOSHR EXE RD WRT NOVEC LONG
XAB\$B	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
DATA	00000354 (852.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
\$RMSNAM	00000014 (20.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
--\$RMSNAM	00000074 (116.)	05 (5.)	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	00:00:01.10
Command processing	105	00:00:00.52	00:00:02.05
Pass 1	403	00:00:19.93	00:00:53.16
Symbol table sort	8	00:00:01.03	00:00:02.17
Pass 2	124	00:00:03.96	00:00:09.57
Symbol table output	19	00:00:00.15	00:00:00.56
Psect synopsis output	3	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	693	00:00:25.71	00:01:08.65

The working set limit was 1350 pages.
95037 bytes (186 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 861 non-local and 5 local symbols.
545 source lines were read in Pass 1, producing 67 object records in Pass 2.
58 pages of virtual memory were used to define 44 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	34
TOTALS (all libraries)	34

1333 GETS were required to define 34 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:RMSTEST8/OBJ=OBJ\$:RMSTEST8 MSRC\$:RMSTEST8/UPDATE=(ENH\$:RMSTEST8)+EXECMLS/LIB

0409

**DIGITAL
CONFIDE**

EQUIPMENT
INITIAL AND

CORPORATION
PROPRIETARY