

UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTTTT	PPPPPPPPPPPP	
UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTTTT	PPPPPPPPPPPP	
UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTTTT	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	EEEEEEEEEEEEEEEE	TTT	PPPPPPPPPPPP	
UUU	UUU	EEEEEEEEEEEEEEEE	TTT	PPPPPPPPPPPP	
UUU	UUU	EEEEEEEEEEEEEEEE	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	UUU	EEEEEEEEEEEEEEEE	TTT	PPP	
UUUUUUUUUUUUUUUU	UUU	EEEEEEEEEEEEEEEE	TTT	PPP	
UUUUUUUUUUUUUUUU	UUU	EEEEEEEEEEEEEEEE	TTT	PPP	

_s
Va
--
000
000
000
7F1
7F1
7F1
7F1
7F1
7F1
7F1
7F1

```

RRRRRRRR MM MM SSSSSSSS TTTTTTTTTT EEEEEEEEE E SSSSSSSS TTTTTTTTTT 44 44
RRRRRRRR MM MM SSSSSSSS TTTTTTTTTT EEEEEEEEE E SSSSSSSS TTTTTTTTTT 44 44
RR RR RR MMMM MMMM SS TT EE SS TT 44 44
RR RR RR MMMM MMMM SS TT EE SS TT 44 44
RR RR RR MM MM MM SS TT EE SS TT 44 44
RR RR RR MM MM MM SS TT EE SS TT 44 44
RRRRRRRR MM MM SSSSSS SS TT EEEEEEE SSSSSS TT 4444444444
RRRRRRRR MM MM SSSSSS SS TT EEEEEEE SSSSSS TT 4444444444
RR RR MM MM SS TT EE SS TT 44
RR RR MM MM SS TT EE SS TT 44
RR RR MM MM SS TT EE SS TT 44
RR RR MM MM SSSSSSSS TT EEEEEEEEE SSSSSSSS TT 44
RR RR MM MM SSSSSSSS TT EEEEEEEEE SSSSSSSS TT 44

```

```

....
....
....
....

```

```

LL I I I I I SSSSSSSS
LL I I I I I SSSSSSSS
LL I I SS
LL I I SS
LL I I SS
LL I I SSSSSS
LL I I SSSSSS
LL I I SS
LL I I SS
LL I I SS
LL I I SS
LLLLLLLLLL I I I I I SSSSSSSS
LLLLLLLLLL I I I I I SSSSSSSS

```

```

0000 1      .IDENT 'V04-000'
0000 78     $BEGIN RMSTEST4,009.  __RMSTEST,<XAB RMS TEST PROGRAM>,<GBL, LONG>
0000 79
0000 80 :
0000 81
0000 82     .ENABL  DBG
0000 83
0000 84 :
0000 85 : macros:
0000 86 :
0000 87 :
0000 88 :
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     $PUT  RAB=CMDORAB,ERR=REPORT_ERROR
0000 97     BSBW  ERR
0000 98 L:
0000 99     .ENDM  TYPE
0000 100
0000 101 :
0000 102
0000 103     .MACRO  STORE STRING,PRE
0000 104     .SAVE
0000 105     .PSECT  --$RMSNAM
0000 106     $$$TMPX=
0000 107     PRE                                     ; store any carriage control info
0000 108     .ASCII %STRING%
0000 109     $$$TMPX1=-.-$$.TMPX
0000 110     .RESTORE
0000 111     .ENDM  STORE

```

```
0000 113
0000 114 ;
0000 115
0000 116 .MACRO BEGIN TSTNAM
0000 117 STORE <TSTNAM>
0000 118 MOVL #$$ .TMPX, BEG_DESCR+4 ; addr
0000 119 MOVL #$$ .TMPX1, BEG_DESCR ; len
0000 120 BSBW BEGPUT
0000 121 .ENDM BEGIN
0000 122 .MACRO FINISH TSTNAM
0000 123 STORE <TSTNAM>
0000 124 MOVL #$$ .TMPX, FIN_DESCR+4 ; addr
0000 125 MOVL #$$ .TMPX1, FIN_DESCR ; len
0000 126 BSBW FINPUT
0000 127 .ENDM FINISH
0000 128 .MACRO FIELD FLDNAM
0000 129 STORE <FLDNAM>
0000 130 MOVL #$$ .TMPX, FLD_DESCR+4 ; addr
0000 131 MOVL #$$ .TMPX1, FLD_DESCR ; len
0000 132 BSBW FLDPUT
0000 133 .ENDM FIELD
```

```

00000000 135      .PSECT      RMSTEST,GBL,LONG
          0000 136      .ALIGN      LONG
          0000 137 T4START::
          0000 138 T4FAB:: $FAB      FNM=<TST$DISK:T4FILE.DAT;1>,-
          0000 139      ORG=SEQ,-
          0000 140      RFM=VFC,-
          0000 141      RAT=CR,-
          0000 142      FSZ=4,-
          0000 143      MRS=100,-
          0000 144      NAM=NAMBLK,-
          0000 145      DEQ=12
          0C 0 146 FLUSH_FAB::
          0050 147      $FAB      FAC=<PUT,GET>,-
          0050 148      FNM=<TST$DISK:T4FILE.DAT;1>,-
          0050 149      NAM=NAMBLK,-
          0050 150      SHR=<PUT,GET,UPI>,-
          0050 151      XAB=FHCXAB
          00A0 152
          00A0 153 :
          00A0 154 :      attention: in order to assemble this module, t4rab and FLUSH_RAB
          00A0 155 :      have been put into another module, RMSTESTR
          00A0 156 :
          00A0 157
          00A0 158 FHCXAB::
          00A0 159      $XABFHC NXT=ALQXAB
          00CC 160 ALQXAB::
          00CC 161      $XABALL NXT=PROXAB,-
          00CC 162      DEQ=15
          00EC 163 PROXAB::
          00EC 164      $XABPRO
          0144 165 DATXAB::
          0144 166      $XABDAT
          0170 167 RDTXAB::
          0170 168      $XABRDT
          0184 169 TRMXAB::
          0184 170      $XABTRM
          01A8 171      $RMSDEFEND
          00000024 01A8 172      EXTRA=XAB$L_SBN-4      ; 4 bytes of extra (spare) char.
          0000 01A8 173 SAVEPRO:
          01A8 174      .WORD      0      ; word to save pro in
          01AA 175
          01AA 176 :
          01AA 177 ;THESE ARE THE DATA STRUCTURES FOR DATE AND TIME XAB CHECKS
          01AA 178 :
          01AA 179
          20 33 36 39 31 2D 52 41 4D 2D 33 20 01AA 180 CDT:      .ASCII / 3-MAR-1963 03:03:03.03/
          33 30 2E 33 30 3A 33 30 3A 33 30 01B6
          00000017 01C1 181      CDTL=-CDT
          20 34 34 39 31 2D 52 50 41 2D 34 20 01C1 182 RDT:      .ASCII / 4-APR-1944 04:04:04.04/
          34 30 2E 34 30 3A 34 30 3A 34 30 01CD
          00000017 01D8 183      RDTL=-RDT
          20 38 38 39 31 2D 47 55 41 2D 38 20 01D8 184 EDT:      .ASCII / 8-AUG-1988 08:08:08.08/
          38 30 2E 38 30 3A 38 30 3A 38 30 01E4
          00000017 01EF 185      EDTL=-EDT
          20 38 34 39 31 2D 43 45 44 2D 32 31 01EF 186 RDT2:      .ASCII /12-DEC-1948 12:12:12.12/
          32 31 2E 32 31 3A 32 31 3A 32 31 01FB
          00000017 0206 187      RDTL2=-RDT2

```



```

037F 263 ERR=REPORT_ERROR
038E 264
038E 265 ;
038E 266 ;using alq from xab
038E 267 ;
038E 268
30 FC6F' 30 038E 269 BSBW ERR
FD47 CF D1 0391 270 Cmpl ALQXAB+XAB$$_ALQ,#48 ; alq in xab should ret actual alq
15 18 0396 271 BGEQ ALQOK
0398 272 FIELD <ALQ IN XAB (NOT = DESIRED ALLOC ON EXTEND)>
03AD 273 ALQOK:
03AD 274
03AD 275 ;
03AD 276 ;undo damage to xab links
03AD 277 ;
03AD 278
24 AB FCEF CF DE 03AD 279 MOVAL FHCXAB,FAB$$_XAB(R11)
FCEA CF FD15 CF DE 03B3 280 MOVAL ALQXAB,XAB$$_NXT+FHCXAB
FDOF CF FD2E CF DE 03BA 281 MOVAL PROXAB,XAB$$_NXT+ALQXAB
03C1 282 $CONNECT RAB=T4RAB,-
03C1 283 ERR=REPORT_ERROR
32 41 8F 6E 00 30 03D4 284 BSBW ERR
00000000'EF 00000000'EF 2C 03D7 285 MOVCS #0,(SP),#^A/A/,#50,CPYBUF; move 50 a's into cpybuf
00000000'EF 00000000'EF DE 03E2 286 MOVAL CPYBUF,RAB$$_RBF+T4RAB
00000000'EF 32 B0 03ED 287 MOVW #50,RAB$$_RSZ+T4RAB
03F4 288 $PUT RAB=T4RAB,-
03F4 289 ERR=REPORT_ERROR
FBF6' 30 0407 290 BSBW ERR

```



```

040A 292
040A 293
040A 294 : take some time to try out flush
040A 295
040A 296
040A 297 $FLUSH RAB=T4RAB,-
040A 298 ERR=REPORT_ERROR
FBEO' 30 041D 299 BSBW ERR
0404 30 0420 300 BSBW ZERO_XABS
0423 301 $OPEN FAB=FLUSH_FAB,-
0423 302 ERR=REPORT_ERROR
FBC9' 30 0434 303 BSBW ERR
0418 30 0437 304 BSBW CHECK_XABS
043A 305 $CONNECT RAB=FLUSH_RAB,-
043A 306 ERR=REPORT_ERROR
FBB0' 30 044D 307 BSBW ERR
0450 308 $GET RAB=FLUSH_RAB,-
0450 309 ERR=REPORT_ERROR
FB9A' 30 0463 310 BSBW ERR
32 00000000'EF B1 0466 311 CMPW RAB$W_RSZ+FLUSH_RAB,#50 ; got right rec. size
15 13 046D 312 BEQL RSZ_OK
046F 313 FIELD <RSZ IN RAB>
00 41 8F 00000000'EF 32 2D 0484 314 RSZ_OK:
6E 13 0484 315 CMPCS #50,CPYBUF,#^A/A/,#0,(SP); is record ok?
15 13 048E 316
0491 317 BEQL YES
04A6 318 YES: FIELD <RECORD>
04A6 319 $GET FLUSH_RAB ; this should be eof
00000000'8F 50 D1 04B3 320 CMPL RO,#RMS$_EOF
0A 13 04BA 321 BEQL 10$
5A 00000000'EF DE 04BC 322 MOVAL FLUSH_RAB,R10
FB3A' 30 04C3 323 BSBW EOFPUT
FBAA CF D4 04C6 324 10$:
04C6 325 CLRL FAB$L_XAB+FLUSH_FAB ; no xabs on close for now
04CA 326 $DISCONNECT RAB=FLUSH_RAB,- ; clean up after flush
04CA 327 ERR=REPORT_ERROR
FB20' 30 04DD 328 BSBW ERR
04E0 329 $CLOSE FAB=FLUSH_FAB,- ; all done w/ flush test
04E0 330 ERR=REPORT_ERROR
FB0C' 30 04F1 331 BSBW ERR
04F4 332 TYPE <ALL DONE WITH FLUSH TEST>
0523 333
0523 334 :
0523 335 : all done with flush test
0523 336 :
0523 337

```

```

0523 339
0523 340
0523 341 $DISCONNECT RAB=T4RAB,-
0523 342 ERR=REPORT_ERROR
FAC7' 30 0536 343 BSBW ERR
0539 344 $CLOSE FAB=R11,-
0539 345 ERR=REPORT_ERROR
FAB5' 30 0548 346 BSBW ERR
0548 347
24 AB 17 AB 94 054B 348 CLRFB FAB$B SHR(R11)
FB4E CF DE 054E 349 MOVAL FHCXAB,FAB$L_XAB(R11) ; set up xab links again
02D0 30 0554 350 BSBW ZERO_XABS
0557 351 $OPEN FAB=R11,-
0557 352 ERR=REPORT_ERROR
FA97' 30 0566 353 BSBW ERR
0569 354
15 04 AB 15 E1 0569 355 BBC #FAB$V CBT,FAB$L_FOP(R11),CC
056E 356 FIELD <CBT BIT WAS SET, THEREFORE>
15 04 AB 14 E1 0583 357 CC: BBC #FAB$V CTG,FAB$L_FOP(R11),OK; after extend, not ctg
0588 358 FIELD <CTG BIT WAS SET, THEREFORE>
059D 359
059D 360 ;
059D 361 ; check fhc xab
059D 362 ;
059D 363
02B2 30 059D 364 OK: BSBW CHECK_XABS
05A0 365
05A0 366 ;
05A0 367 ;do another extend, forcing it to get the value from the alq of the fab
05A0 368 ;
05A0 369
10 AB 24 AB D4 05A0 370 CLRL FAB$L_XAB(R11)
OC OC D0 05A3 371 MOVL #12,FAB$L_ALQ(R11)
05A7 372 $EXTEND FAB=R11,-
05A7 373 ERR=REPORT_ERROR
OC FA47' 30 05B6 374 BSBW ERR
10 AB D1 05B9 375 CMPL FAB$L_ALQ(R11),#12 ; alq in fab=12
15 18 05BD 376 BGEQ ALQOKT
05BF 377 FIELD <ALQ IN FAB (NOT = DESIRED ALLOCATION AFTER EXTEND)>
05D4 378 ALQOK1:
05D4 379
05D4 380 ;
05D4 381 ;change protection and uic on close
05D4 382 ;
05D4 383
05D4 384 $XABPRO_STORE XAB=PROXAB,-
05D4 385 PRO=<RWED,RWED,RED,RWED>,-
05D4 386 UIC=<222,55>
24 AB 50 D0 05E9 387 MOVL R0,FAB$L_XAB(R11) ; set up xab
FB85 CF 08 A0 B0 05ED 388 MOVW XAB$W_PRO(R0),SAVEPRO ; for check
FC56 CF OC A0 D0 05F3 389 MOVL XAB$L_UIC(R0),UIC ; ditto
05F9 390 $CLOSE FAB=R11,-
05F9 391 ERR=REPORT_ERROR
24 AB FA95' 30 0608 392 BSBW ERR
FA91 CF DE 060B 393 MOVAL FHCXAB,FAB$L_XAB(R11)
0611 394 $OPEN FAB=R11,- ; check changes after ext
0611 395 ERR=REPORT_ERROR

```

```

15 04 AB F9DD' 30 0620 396 BSBW ERR
15 04 AB 15 E1 0623 397 BBC #FABS_V CBT,FABS_L_FOP(R11),NOCBT
15 04 AB 14 E1 0628 398 FIELD <CBT BIT WAS SET, THEREFORE>
063D 399 NOCBT: BBC #FABS_V CTG,FABS_L_FOP(R11),NOCTG ; shouldn't be ctg, after extend
0642 400 FIELD <CTG BIT WAS SET, AFTER 2 EXTENDS, THEREFORE>
0657 401 NOCTG:
0657 402 MOVAL FHCXAB,R9 ; check pertinent fields
00000046 59 FA45 CF DE 0657 403 CMP XABS_L_HBK(R9),#70 ; alq=10+48+12
8F OC A9 D1 065C 404 BGEQ HBKOK
15 18 0664 405 FIELD <HBK IN FHCXAB (AFTER 2ND EXTEND)>
00 28 A9 D1 067B 406 HBKOK: CMPL XABS_L_SBN(R9),#0 ; not ctg anymore
15 13 067F 407 BEQL STILL_OK
0681 408 FIELD <SBN IN FHCXAB (AFTER 2ND EXTEND)>
0696 409 STILL_OK:
01C9 30 0696 410 BSBW CHECK_ALL
00 46 0699 411 .BYTE 70,0 ; values for alq,ctg (not ctg)
025E 30 069B 412 BSBW CHECK_PRO
069E 413 TYPE <DONE WITH 2ND EXTEND, NOW TEST DATES>
24 AB D4 06CD 414 CLRL FABSL_XAB(R11) ; no xabs on this close, for now
06D0 415 MTA: $CLOSE FAB=R11,- ; continue if mta
06D0 416 BSBW ERR=REPORT_ERROR
F91E' 30 06DF 417 BSBW ERR
06E2 418
06E2 419 ;
06E2 420 ;before finishing up, have some fun with the dat and rdt xabs
06E2 421 ;
06E2 422 ;
59 FASE CF DE 06E2 423 MOVAL DATXAB,R9
06E7 424 $BINTIM_S CDTDEC,XABS_Q_CDT(R9)
06F5 425 $BINTIM_S RDTDEC,XABS_Q_RDT(R9)
0703 426 $BINTIM_S EDTDEC,XABS_Q_EDT(R9)
08 A9 00C8 8F B0 0711 427 MOVW #200,XABS_W_RVN(R9)
FB08 CF 00C8 8F B0 0717 428 MOVW #200,CURRVN
24 AB 59 D0 071E 429 MOVL R9,FABS_L_XAB(R11)
04 A9 D4 0722 430 CLRL XABS_L_NXT(R9)
FB1F CF FA98 CF DE 0725 431 MOVAL RDT,CURRDT ; current rdt str
04 AB 00000080 8F C8 072C 432 BISL #FABS_M_RWO,FABS_L_FOP(R11) ; rewind if mta
0734 433 $CREATE FAB=R11,-
0734 434 ERR=REPORT_ERROR
F8BA' 30 0743 435 BSBW ERR
F9FB CF FA26 CF DE 0746 436 MOVAL RDTXAB,XABS_L_NXT+DATXAB
00C6 30 074D 437 BSBW ZERO_DAT_XABS
0750 438 $DISPLAY FAB=R11,-
0750 439 ERR=REPORT_ERROR
F89E' 30 075F 440 BSBW ERR
032B 30 0762 441 BSBW CHECK_DATES
FADF CF FA86 CF DE 0765 442 MOVAL RDT2,CURRDT ; get a new rdt
59 FA00 CF DE 076C 443 MOVAL RDTXAB,R9
0771 444 $BINTIM_S RDT2DEC,XABS_Q_RDT(R9)
08 A9 012C 8F B0 077F 445 MOVW #300,XABS_W_RVN(R9)
FA9A CF 012C 8F B0 0785 446 MOVW #300,CURRVN
24 AB 59 D0 078C 447 MOVL R9,FABS_L_XAB(R11) ; only rdt for close
0790 448 $CLOSE FAB=R11,-
0790 449 ERR=REPORT_ERROR
F85E' 30 079F 450 BSBW ERR
24 AB F99E CF DE 07A2 451 MOVAL DATXAB,FABS_L_XAB(R11) ; only dat for open
F99C CF D4 07A8 452 CLRL XABS_L_NXT+DATXAB

```

	68	10	07AC	453	BSBB	ZERO_DAT_XABS		
			07AE	454	\$FAB_STORE	FAB=R11,-		; can't 'put' to mta
			07AE	455		FAC=GET		
			07B2	456	\$OPEN	FAB=R11,-		
			07B2	457		ERR=REPORT_ERROR		
24 AB	F83C'	30	07C1	458	BSBW	ERR		
	F9AB (F	DE	07C4	459	MOVAL	RDTXAB,FAB\$L_XAB(R11)		; get rdt on display
			07CA	460	\$DISPLAY	FAB=R11,-		
			07CA	461		ERR=REPORT_ERROR		
	F824'	30	07D9	462	BSBW	ERR		
	02B1	30	07DC	463	BSBW	CHECK_DATES		
24 AB	D4	D4	07DF	464	CLRL	FAB\$L_XAB(R11)		; no xabs for this close
			07E2	465	\$CLOSE	FAB=R11,-		
			07E2	466		ERR=REPORT_ERROR		
	F80C'	30	07F1	467	BSBW	ERR		
			07F4	468				
			07F4	469				
			07F4	470	\$FAB_STORE	FAB=R11,-		; restore fac
			07F4	471		FAC=PUT		
04 AB	0000080	8F	CA	07F8	BICL	#FAB\$M_RWO,FAB\$L_FOP(R11)		; and fop
			04	0800	FINISH	<XAB TESTS>		
			04	0815	RET			
				474				

```

0816 476
0816 477
0816 478 ;2 routines to zero out the xabs before checking the results
0816 479 ;
0816 480
0816 481 ZERO_DAT_XABS:
0816 482
0816 483 ;
0816 484 ;zero out the dat and rdt xabs
0816 485 ;
0816 486
F92C CF 26 00 6E 00 2C 0816 487      MOVCS  #0,(SP),#0,#<XAB$C_DATLEN-6>,DATXAB+6
F950 CF 0E 00 6E 00 2C 081E 488      MOVCS  #0,(SP),#0,#<XAB$C_RDTLEN-6>,RDTXAB+6
                                05 0826 489      RSB
0827 490
0827 491 ZERO_XABS:
0827 492
0827 493 ;
0827 494 ;zero out the fhc, all and pro xabs
0827 495 ;
0827 496
F877 CF 26 00 6E 00 2C 0827 497      MOVCS  #0,(SP),#0,#<XAB$C_FHCLEN-6>,FHCXAB+6
F89B CF 1A 00 6E 00 2C 082F 498      MOVCS  #0,(SP),#0,#<XAB$C_ALLEN-6>,ALQXAB+6
0052 8F 00 6E 00 2C 0837 499      MOVCS  #0,(SP),#0,#<XAB$C_PROLEN-6>,PROXAB+6
                                F8B1 CF
                                05 0841 500      RSB

```

```

0842 502 CHECK_CR:
0842 503
0842 504 ;
0842 505 ;routine to check xabs after create and subsequent displ
0842 506 ;
0842 507
00F7 30 0842 508 BSBW CHECK_FHC
U1 00 0A 00 0845 509 .BYTE 0,10,0,1 ; values for lrl,alq,ffb,sbn
0016 30 0849 510 BSBW CHECK_ALL
01 0A 084C 511 .BYTE 10,1 ; values for alq,ctg ( should be)
00AB 30 084E 512 BSBW CHECK_PRO
05 0851 513 RSB
0852 514
0852 515 CHECK_XABS:
0852 516
0852 517 ;
0852 518 ;general routine to check out all xabs
0852 519 ;
0852 520
00E7 30 0852 521 BSBW CHECK_FHC
00 38 3A 32 0855 522 .BYTE 50,58,56,0 ; values for lrl,alq,ffb,sbn(not ctg)
0006 30 0859 523 BSBW CHECK_ALL
00 3A 085C 524 .BYTE 58,0 ; values for alq,ctg ( not ctg anymore)
009B 30 085E 525 BSBW CHECK_PRO
05 0861 526 RSB
0862 527
0862 528 CHECK_ALL:
0862 529
0862 530 ;
0862 531 ;routine to check out the allocation xab
0862 532 ;
0862 533
59 F866 CF DE 0862 534 MOVAL ALQXAB,R9
OF 14 A9 B1 0867 535 CMPW XABS$W_DEQ(R9),#15
15 13 086B 536 BEQL DEQOK-
086D 537 FIELD <DEQ IN ALL. XAB>
00 16 A9 91 0882 538 DEQOK: CMPB XABS$B_BKZ(R9),#0
15 13 0886 539 BEQL BKZOK
0888 540 FIELD <BKZ IN ALL. XAB>
50 00 BE 9A 089D 541 BKZOK: MOVZBL @(SP),R0
6E D6 08A1 542 INCL (SP)
10 A9 50 D1 08A3 543 CMPL R0,XABS$L_ALQ(R9)
15 15 08A7 544 BLEQ ALQOK2
08A9 545 FIELD <ALQ IN ALL. XAB>
50 00 BE 9A 08BE 546 ALQOK2: MOVZBL @(SP),R0
6E D6 08C2 547 INCL (SP)
1A 50 E9 08C4 548 BLBC R0,NOTCTG
2F 08 A9 07 E0 08C7 549 BBS #XABS$V_CTG,XABS$B_AOP(R9),AOPOK ; should be set
08CC 550 FIELD <CTG CLR IN AOP, -THEREFORE>
15 08 A9 07 E1 08E1 551 NOTCTG: BBC #XABS$V_CTG,XABS$B_AOP(R9),AOPOK ; should be clear
08E6 552 FIELD <CTG SET IN AOP, -THEREFORE>
05 08FB 553 AOPOK: RSB
08FC 554
08FC 555 CHECK_PRO:
08FC 556
08FC 557 ;
08FC 558 ;check the protection xab

```

```

      08FC 559 ;
      08FC 560
08 A9 F7EC CF DE 08FC 561 MOVAL PROXAB,R9
      F8A3 CF B1 0901 562 CMPW SAVEPRO,XAB$W_PRO(R9) ; cmp to saved value
      15 13 0907 563 BEQL PROCK
      0909 564 FIELD <PROT FIELD IN PROT XAB>
0C A9 F92D CF D1 091E 565 PROOK: CMPL UIC,XAB$L_UIC(R9)
      15 13 0924 566 BEQL UICOK
      0926 567 FIELD <UIC FIELD IN PROT. XAB>
      05 093B 568 UICOK: RSB
      093C 569
```

```

093C 571 CHECK_FHC:
093C 572
093C 573 :
093C 574 : check fhc xab carefully
093C 575 :
093C 576
59 F760 CF DE 093C 577 MOVAL FHCXAB,R9 ; r9 is ptr to xab thru-out cmp's
03 08 A9 91 0941 578 EXTC:
15 13 0941 579 CMPB XAB$B_RFO(R9),#FAB$C_VFC; check rec. format & org.
09 A9 02 93 0945 580 BEQL RFOC
15 12 0947 581 FIELD <RFO IN FHC XAB>
095C 582 RFOC: BITB #FAB$M_CR,XAB$B_ATR(R9) ; check rat field
0960 583 BNEQ ATRC
0962 584 FIELD <ATR IN FHC XAB>
50 00 BE 9A 0977 585 ATRC: MOVZBL @(SP),R0
6E D6 097B 586 INCL (SP)
50 0A A9 B1 097D 587 CMPW XAB$W_LRL(R9),R0 ; check longest record len
15 13 0981 588 BEQL LRLC
0983 589 FIELD <LRL IN FHC XAB>
50 00 BE 9A 0998 590 LRLC: MOVZBL @(SP),R0
6E D6 099C 591 INCL (SP)
50 0C A9 D1 099E 592 CMPL XAB$L_HBK(R9),R0 ; check alq
15 18 09A2 593 BGEQ HBKC
09A4 594 FIELD <HBK IN FHC XAB>
01 10 A9 D1 09B9 595 HBKC: CMPL XAB$L_EBK(R9),#1 ; check end block
15 13 09BD 596 BEQL EBKC
09BF 597 FIELD <EBK IN FHC XAB>
50 00 BE 9A 09D4 598 EBKC: MOVZBL @(SP),R0
6E D6 09D8 599 INCL (SP)
50 14 A9 B1 09DA 600 CMPW XAB$W_FFBC(R9),R0 ; check first free byte
15 13 09DE 601 BEQL FFBC ; its len of rec + fsz + 2
09E0 602 FIELD <FFB IN FHC XAB>
00 16 A9 91 09F5 603 FFBC: CMPB XAB$B_BKZ(R9),#0 ; check bucket size
15 13 09F9 604 BEQL BKZC
09FB 605 FIELD <BKZ IN FHC XAB>
04 17 A9 91 0A10 606 BKZC: CMPB XAB$B_HSZ(R9),#4 ; check fixed area size
15 13 0A14 607 BEQL HSZC
0A16 608 FIELD <HSZ IN FHC XAB>
0064 8F 18 A9 B1 0A2B 609 HSZC: CMPW XAB$W_MRZ(R9),#100 ; check max. rec size
15 13 0A31 610 BEQL MRZC
0A33 611 FIELD <MRZ IN FHC XAB>
0F 1A A9 B1 0A48 612 MRZC: CMPW XAB$W_DXQC(R9),#15 ; check def ext. qty
15 13 0A4C 613 BEQL DXQC
0A4E 614 FIELD <DXQ IN FHC XAB>
50 00 BE 9A 0A63 615 DXQC: MOVZBL @(SP),R0
6E D6 0A67 616 INCL (SP)
08 50 E9 0A69 617 BLBC R0,10$
00 28 A9 D1 0A6C 618 CMPL XAB$L_SBN(R9),#0 ; make sure non-zero lbn
0A70 619
0A70 620 :
0A70 621 ;since it's ctg
0A70 622 :
0A70 623
0A70 624 BNEQ FHC_OK
00 28 A9 D1 0A72 625 BRB 20$
15 13 0A74 626 10$: CMPL XAB$L_SBN(R9),#0 ; make sure zero lbn
0A78 627 BEQL FHC_OR ; since it isn't contig.

```


RMSTEST4
009

XAB RMS TEST PROGRAM

D 10

16-SEP-1984 01:47:44 VAX/VMS Macro V04-00
5-SEP-1984 04:21:52 [UETP.SRC]RMSTEST4.MAR;1

Page 15
(13)

RM
VC

05 0A7A 628 20\$: FIELD <SBN IN FHC XAB>
0A8F 629 FHC_OK: RSB

```

0A90 631
0A90 632 CHECK_DATES:
0A90 633
0A90 634 ;
0A90 635 ;routine to check edt and cdt in dat xab
0A90 636 ;and rdt and rvn in both dat and rdt xab's
0A90 637 ;
0A90 638
05 40 AB F793 CF OB B0 0A90 639 MOVW #DATLEN,LEN ; default is check date only
          00000000'8F E0 0A95 640 BBS #DEVSV SQD,FAB$$_DEV(R11),10$
          F785 CF 14 B0 0A9E 641 MOVW #TIMLEN,LEN ; if not mta, check date and time
          59 F69D CF DE 0AA3 642 10$: MOVAL DATXAB,R9
F766 CF F6E9 CF F76A CF 29 0AA8 643 $ASCTIM_S ,CMPDATDEC,XAB$$_CDT(R9)
          15 13 0ABA 644 CMPC3 LEN,CDT,CMPDAT
          0AC4 645 BEQL CDTOK
          0AC6 646 FIELD <CDT IN DAT XAB>
          0ADB 647 CDTOK:
          0ADB 648 $ASCTIM_S ,CMPDATDEC,XAB$$_EDT(R9)
F735 CF F6E6 CF OB 29 0AED 649 CMPC3 #DATLEN,EDT,CMPDAT ; only check date
          15 13 0AF5 650 BEQL EDTOK
          0AF7 651 FIELD <EDT IN DAT XAB>
          0B0C 652 EDTOK:
01 40 AB 00000000'8F E1 0B0C 653 BBC #DEVSV SQD,FAB$$_DEV(R11),10$
          05 0B15 654 RSB ; that's it if mta
F6F8 CF F71C DF F6FC CF 29 0B16 655 10$: $ASCTIM_S ,CMPDATDEC,XAB$$_RDT(R9)
          15 13 0B28 656 CMPC3 LEN,@CURRDT,CMPDAT
          0B32 657 BEQL RDTOK
          0B34 658 FIELD <RDT IN DAT XAB>
          0B49 659 RDTOK:
F6C4 CF F6E8 DF F6C8 CF 29 0B49 660 $ASCTIM_S ,CMPDATDEC,XAB$$_RDT+RDTXAB
          15 13 0B5C 661 CMPC3 LEN,@CURRDT,CMPDAT
          0B66 662 BEQL RDTOK1
          0B68 663 FIELD <RDT IN RDT XAB>
          0B7D 664 RDTOK1:
          0B7D 665 CMPW CURRVN,XAB$$_RVN(R9)
          15 13 0B83 666 BEQL RVNOK
          0B85 667 FIELD <RVN IN DAT XAB>
          15 13 0B9A 668 RVNOK: CMPW CURRVN,XAB$$_RVN+RDTXAB
          0BA1 669 BEQL RVNOK1
          0BA3 670 FIELD <RVN IN RDT XAB>
          05 0BB8 671 RVNOK1: RSB
          0BB9 672 .END

```

```

$$PSECT_EP      = 00000000
$$TAB           = 00000184 R D 01
$$TABEND        = 000001A8 R D 01
$$TMP           = 00000001
$$TMP1          = 00000002
$$TMP2          = 0000005B
$$TMP5          = 00000002
$$TMPX          = 00000352 R D 04
$$TMPX1         = 0000000E
$$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            = 00000001 D
ALQOK           = 000003AD R D 01
ALQOK1          = 000005D4 R D 01
ALQOK2          = 000008BE R D 01
ALQXAB          = 000000CC RG D 01
AOPOK           = 000008FB R D 01
ATRC            = 00000977 R D 01
BEGPUT          = ***** X 01
BEG_DESCR       = ***** X 01
BKZC            = 00000A10 R D 01
BKZOK           = 0000089D R D 01
CC              = 00000583 R D 01
CDT             = 000001AA R D 01
CDTDEC          = 00000206 R D 01
CDTL            = 00000017 D
CDTOK           = 00000ADB R D 01
CHECK_ALL       = 00000862 R D 01
CHECK_CR        = 00000842 R D 01
CHECK_DATES     = 00000A90 R D 01
CHECK_FHC       = 0000093C R D 01
CHECK_PRO       = 000008FC R D 01
CHECK_XABS      = 00000852 R D 01
CMDORAB        = ***** X 01
CMPDAT          = 0000022A R D 01
CMPDATDEC       = 00000243 R D 01
CPYBUF         = ***** X 01
CURRDT          = 0000024B R D 01
CURRVN          = 00000226 R D 01
DATLEN          = 0000000B D
DATXAB          = 00000144 RG D 01
DEQOK           = 00000882 R D 01
DEVSV_SQD      = ***** X 01
DXQC            = 00000A63 R D 01
EBKC            = 000009D4 R D 01
EDT             = 000001D8 R D 01
EDTDEC          = 00000216 R D 01
EDTL            = 00000017 D
EDTOK           = 0000080C R D 01
EOFPUT          = ***** X 01

```

```

ERR             ***** X 01
EXTC            00000941 R D 01
EXTRA           = 00000024 D
FABS_B_FAC      = 00000016 D
FABS_B_FNS      = 00000034 D
FABS_B_SHR      = 00000017 D
FABS_C_BID      = 00000003 D
FABS_C_BLN      = 00000050 D
FABS_C_SEQ      = 00000000 D
FABS_C_VAR      = 00000002 D
FABS_C_VFC      = 00000003 D
FABS_L_ALQ      = 00000010 D
FABS_L_DEV      = 00000040 D
FABS_L_FNA      = 0000002C D
FABS_L_FOP      = 00000004 D
FABS_L_XAB      = 00000024 D
FABS_M_CR       = 00000002 D
FABS_M_RWO      = 00000080 D
FABS_V_CBT      = 00000015 D
FABS_V_CHAN_MODE = 00000002 D
FABS_V_CR       = 00000001 D
FABS_V_CTG      = 00000014 D
FABS_V_FILE_MODE = 00000004 D
FABS_V_GET      = 00000001 D
FABS_V_LNM_MODE = 00000000 D
FABS_V_PUT      = 00000000 D
FABS_V_SUP      = 00000002 D
FABS_V_UPI      = 00000006 D
FABS_W_GBC      = 00000048 D
FFBC            000009F5 R D 01
FHCXAB          000000A0 RG D 01
FHC_OK          00000A8F R D 01
FINPUT          ***** X 01
FIN_DESCR       ***** X 01
FLDPUT          ***** X 01
FLD_DESCR       ***** X 01
FLUSH_FAB       00000050 RG D 01
FLUSH_RAB       ***** X 01
HBKC            000009B9 R D 01
HBKOK           0000067B R D 01
HSZC            00000A2B R D 01
LEN             00000228 R D 01
LRLC            00000998 R D 01
MRZC            00000A48 R D 01
MTA             000006D0 R D 01
NAMBLK          ***** X 01
NOCBT           0000063D R D 01
NOCTG           00000657 R D 01
NOTCTG          000008E1 R D 01
OK              0000059D R D 01
OK1             0000030F R D 01
OK2             00000329 R D 01
PROOK           0000091E R D 01
PROXAB          000000EC RG D 01
RABS_L_RBF      ***** X 01
RABS_W_RSZ      ***** X 01
RDT             000001C1 R D 01

```

RDT2	C00001EF	R	D	01	XAB\$C_TRM	=	0000001F	D		
RDT2DEC	0000021E	R	D	01	XAB\$C_TRMLEN	=	00000024	D		
RDTDEC	0000020E	R	D	01	XAB\$S_ACLBUF	=	00000018	D		
RDTL	= 00000017		D		XAB\$S_ACLCTX	=	00000020	D		
RDTL2	= 00000017		D		XAB\$S_ALQ	=	00000010	D		
RDTOK	00000B49	R	D	01	XAB\$S_EBK	=	00000010	D		
RD1OK1	00000B7D	R	D	01	XAB\$S_HBK	=	0000000C	D		
RDTXAB	00000170	RG	D	01	XAB\$S_ITMLST	=	00000008	D		
REPORT_ERROR	*****	X		01	XAB\$S_LOC	=	0000000C	D		
RFOC	0000095C	R	D	01	XAB\$S_NXT	=	00000004	D		
RIGHT	000002F5	R	D	01	XAB\$S_SBN	=	00000028	D		
RMSS_EOF	*****	X		01	XAB\$S_UIC	=	0000000C	D		
RMSTEST_4A	00000253	RG	D	01	XAB\$Q_CDT	=	00000014	D		
RSZ_OK	00000484	R	D	01	XAB\$Q_EDT	=	0000001C	D		
RVNOK	00000B9A	R	D	01	XAB\$Q_RDT	=	0000000C	D		
RVNOK1	00000BB8	R	D	01	XAB\$V_CTG	=	00000007	D		
SAVEPRO	000001A8	R	D	01	XAB\$W_ACLSIZ	=	0000001C	D		
STILL_OK	00000696	R	D	01	XAB\$W_DEQ	=	00000014	D		
SYSSASCTIM	*****	GX		01	XAB\$W_DYQ	=	0000001A	D		
SYSSBINTIM	*****	GX		01	XAB\$W_FFB	=	00000014	D		
SYSSCLOSE	*****	GX		01	XAB\$W_GRP	=	0000000E	D		
SYSSCONNECT	*****	GX		01	XAB\$W_ITMLST_LEN	=	0000000C	D		
SYSSCREATE	*****	GX		01	XAB\$W_LRL	=	0000000A	D		
SYSSDISCONNECT	*****	GX		01	XAB\$W_MBM	=	0000000C	D		
SYSSDISPLAY	*****	GX		01	XAB\$W_MRZ	=	00000018	D		
SYSSEXTEND	*****	GX		01	XAB\$W_PRO	=	00000008	D		
SYSSFLUSH	*****	GX		01	XAB\$W_RF10	=	00000018	D		
SYSSGET	*****	GX		01	XAB\$W_RF12	=	0000001A	D		
SYSSOPEN	*****	GX		01	XAB\$W_RF14	=	0000001C	D		
SYSSPUT	*****	GX		01	XAB\$W_RVN	=	00000008	D		
T4FAB	00000000	RG	D	01	XAB\$W_VOL	=	0000000A	D		
T4RAB	*****	X		01	YES	000004A6	R	D	01	
T4START	00000000	RG	D	01	ZERO_DAT_XABS	00000816	R	D	01	
TIMLEN	= 00000014		D		ZERO_XABS	00000827	R	D	01	
TRMXAB	00000184	RG	D	01						
UIC	0000024F	R	D	01						
UICOK	0000093B	R	D	01						
VERBOSITY	*****	X		01						
XAB\$B_AID	= 00000017		D							
XAB\$B_AOP	= 00000008		D							
XAB\$B_ATR	= 00000009		D							
XAB\$B_BKZ	= 00000016		D							
XAB\$B_HSZ	= 00000017		D							
XAB\$B_MTACC	= 0000000A		D							
XAB\$B_PROT_MODE	= 00000010		D							
XAB\$B_PROT_OPT	= 0000000B		D							
XAB\$B_RFO	= 00000008		D							
XAB\$C_ALL	= 00000014		D							
XAB\$C_ALLLEN	= 00000020		D							
XAB\$C_DAT	= 00000012		D							
XAB\$C_DATLEN	= 0000002C		D							
XAB\$C_FHC	= 0000001D		D							
XAB\$C_FHCLEN	= 0000002C		D							
XAB\$C_PRO	= 00000013		D							
XAB\$C_PROLEN	= 00000058		D							
XAB\$C_RDT	= 0000001E		D							
XAB\$C_RDTLEN	= 00000014		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	00000BB9 (3001.)	01 (1.)	NOPIC USR CON REL GBL NOSHR EXE RD WRT NOVEC LONG
SABSS	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SRMSNAM	0000002A (42.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
SRMSNAM	00000360 (864.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.09	00:00:00.43
Command processing	105	00:00:00.63	00:00:02.75
Pass 1	362	00:00:17.15	00:00:38.17
Symbol table sort	0	00:00:00.53	00:00:01.16
Pass 2	148	00:00:03.68	00:00:08.25
Symbol table output	24	00:00:00.15	00:00:00.31
Psect synopsis output	3	00:00:00.02	00:00:00.30
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	673	00:00:22.25	00:00:51.37

The working set limit was 1350 pages.
80832 bytes (158 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 460 non-local and 9 local symbols.
672 source lines were read in Pass 1, producing 53 object records in Pass 2.
67 pages of virtual memory were used to define 50 macros.

! Macro library statistics !

Macro library name	Macros defined
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	50
TOTALS (all libraries)	50

1077 GETS were required to define 50 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:RMSTEST4/OBJ=OBJ\$:RMSTEST4 MSRC\$:RMSTEST4/UPDATE=(ENH\$:RMSTEST4)+EXECML\$/LIB

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

