

```

1
2      ;***COPYRIGHT 1969, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.***
3
4
5      ;THIS SUB-PROGRAM ASSEMBLED WITH SYSTEM PARAMETER FILE - S,MAC(V414)
6      XLIST
7      LIST
8      ;THIS SUB-PROGRAM ASSEMBLED WITH CONFIGURATION DEPENDENT FEATURE SWITCHES - FT50SB,MAC(
9      V003)
10     XLIST
11     LIST
12     TITLE DTASRN - NEW FORMAT DECTAPE SERVICE FOR TD-10 (PDP-10)
13     SUBTTL T, WACHS/RCC TS 04 JUN 69 V406
14     XP      VDTASX,406*
15
16     ;DEFINE GLOBAL VERSION NUMBER FOR LOADER MAP.
17     IFNDEF ALMACT, <ALMACT=0>
18
19     ENTRY DTASRN
20     ;THIS ENTRY FOR SELECTIVE LOAD BY BUILD
21     INTERNAL DTADSP
22     EXTERNAL TPOPJ,TPOPJ1,DTALOC,DTALC2,DTBOTH,WSYNC,DTACHL
23     EXTERNAL STOIOS,STOTAC,SETACT,CLRACT,OUT,DTASAV,PIOMOD
24     EXTERNAL DTAVAL,DTREQ,DTWAIT,SETIOD,THSDAT,PUNIT
25     EXTERNAL ADVBFE,ADVBF,ADRERR,WAIT1,CPOPJ,CPOPJ1,BADDIR
26     EXTERNAL COMCHK,JOBPD1,DTTURN,PJORN,RELEA9,UADCK1,STREQ
27     EXTERNAL CLOCK,JBADR,DEFVPHY,PION,PIOFF,DITRY
28
29     ;DTALOC=PI LOC FOR DATA - CHANGED BY ROUTINE
30     ;DTALC2=PI LOC+1 FOR DATA
31     ;DTBOTH=10*(PI FOR DATA CHAN) + PI FOR FLAGS CHAN
32     ;DTTURN=300200+DTBOTH (CONO FOR TURN-AROUND)
33
34     000004 BLK=4
35
36
37     000144 DIRBLK=*D100 ;NUMBER OF BLOCK FOR DIRECTORY
38
39     001101 TOPBLK=1101 ;HIGHEST LEGAL BLOCK NUMBER
40     000123 NAMSTR=*D83 ;1ST NAME WORD IN DIRECTORY
41     000003 QUANT=3 ;NUMBER OF BLOCKS CAN READ BEFORE GIVING UP DTC
42     000014 MINDIS=14 ;MINIMUM NUMBER OF BLOCKS TO SEARCH BEFORE DISCONNECTING
43     ;FROM A TAPE
44     000004 SPACE=4 ;NUMBER OF BLOCKS SEPERATING CONTIGUOUS BLKS OF A FILE
45
46     ;DOB MAGIC CELLS
47     000013 FSTBLK=13
48     000014 DL0C=14
49     000015 IBLK=15
50     000016 OBLK=16
51     000017 DISPAD=17
52     000020 DMPLST=20
53     000021 SVDWRD=21

```

53 IFLAGS IN RH OF IOS
54 000100 UDSD=100
55 020000 LSTSAV=20000
56
57

58 IFLAGS IN LH OF IOS
59 000200 NOLINK=200
60 000400 CHNGDR=400
61 001000 RVERSE=1000
62 002000 SINGL=2000
63 004000 DMPMOD=4000
64 010000 RWDIR=10000
65 020000 DMPCLS=20000
66 040000 NOBUF=40000
67 737777 NOBUF=737777
68 100000 REWBIT=100000
69 200000 RUNBIT=200000
70 400000 RECKON=400000
71

IJUST READ OR WRITE 1 BLOCK

IDATA GOING DIRECTLY INTO USER AREA
I-NORUF

ITHIS TAPE IS DEAD-RECKONING.
I (MUST RE SIGN BIT)

72
73 777777 777777 CPBIT=-1
74
75

IFEATURE TEST TO ALLOW IO INTO USER AREA DIRECTLY
IIT WILL ONLY HAPPEN ON DUMP MODE WITH NON-STANDARD BIT
I(UDSD) ON

```

76                                     EXTERN  JIFSEC
77
78                                     INTERN  DTAINI,DTADDR,DTAINI,DTADS,DTADSP
79
80 000000 446441 200000 DTADDB: SIXBIT /DTA0/
81 000001 000074 000200 XWD +060*HUNGST,200
82 000002 000000 000000 0
83 000003 000000 000224 EXP DTADSP
84 000004 001107 154403 XWD 1107,154403
85 000005 000000 000000 EXP 0,0
86 000006 000000 000000
87 000007 000007 000000 XWD PROG,0
88 000010 000007 000000 XWD PROG,0
89 000011 000000 000000 EXP 0,0,0
90 000012 000000 000000
91 000013 000000 000000
92 000014 000000 000022 EXP DTADIR
93 000015 000000 000000 FXP 0,0,0,0
94 000016 000000 000000
95 000017 000000 000000
96 000020 000000 000000
97 000021 000000 000000
98 000022 DTADIR: BLOCK 200
99 000222 DTADS=.-DTADDR
100
101 000222 254000 000247 JRST DTAINI
102 000223 254000 001755 JRST HUNGTP ;HUNG DEVICE
103 000224 254000 000712 DTADSP: JRST UREL
104 000225 254000 000671 JRST UCLS
105 000226 254000 001042 JRST UOUT
106 000227 254000 000772 JRST UIN
107 000230 254000 000501 JRST ENTR
108 000231 254000 000256 JRST LOOK
109 000232 254000 001035 JRST DMPO
110 000233 254000 000766 JRST DMPI
111 000234 254000 000605 JRST SETO
112 000235 254000 000603 JRST SETI
113 000236 254000 000611 JRST GETF
114 000237 254000 000324 JRST RENAM
115 000240 263140 000000 POPJ PDP, ;CLOSE INPUT
116 000241 254000 000650 JRST UTPCLR
117
118 ;FALL INTO MTAPE
    
```

112	000242	550100	000014	HRRZ	TAC1,UUO	MTAPE - GET OPERATION
113	000243	302100	000001	CAIE	TAC1,1	REWIND OR
114	000244	306100	000011	CAIN	TAC1,11	REWIND UNLOAD ARE LEGAL
115	000245	364100	001307	SOJA	TAC1,MTAP0	
116	000246	263140	000000	POPJ	PDP,	OTHERS ARE NO-OPS
117						
118						
119				INITIALIZE DTC		
120	000247	732200	000000	DTAINI:	CONO	DTC,0
121	000250	402000	002152		SETZM	DISCON
122	000251	201040	001750		MOVEI	TAC,*D1000
123	000252	231040	000000		IDIVI	TAC,JIFSEC
124	000253	271040	000001		ADDI	TAC,1
125	000254	542040	001473		HRRM	TAC,MSECTP
126	000255	263140	000000		POPJ	PDP,


```

173 000324 261140 000014 RENAM: PUSH PDP,UUO ;SAVE LOC OF NEW NAME
174 000325 201606 000011 MOVEI UUO,DEVFIL(DEVDAT) ;SEARCH FOR OLD NAME
175 000326 260140 000367' PUSHJ PDP,DSER1
176 000327 254000 000357' JRST RENER1 ;NOT FOUND - ERROR
177 000330 262140 000014 POP PDP,UUO ;FOUND, RESTORE UUO
178 000331 332020 000014 SKIPE @UUO ;RENAMING TO ZERO?
179 000332 254000 000344' JRST RENAM2 ;NO, GO DO REAL RENAME
180 000333 402001 000000 SETZM (TAC) ;YES, DELETE NAME IN DIR
181 000334 402001 000026 SETZM 26(TAC) ;DELETE EXTENSION
182 000335 402006 000011 SETZM DEVFIL(DEVDAT) ;ZERO DEVFIL
183 000336 574040 000001 HLRE TAC,TAC ;GET INDEX OF FILE
184 000337 271040 000027 ADDI TAC,27
185 000340 260140 000473' PUSHJ PDP,DELETE ;DELETE ALL BLOCKS OF FILE
186 000341 661000 000400 RENAM1: TLQ IOS,CHNGDR ;DIRFACTORY HAS CHANGED
187 000342 350003 000000 AOS (PDP) ;SET FOR GOOD RETURN
188 000343 254000 000000 JRST STOIOS ;GO TO USER
189
190 ;COME HERE TO RENAME TO A REAL NEW NAME
191 000344 200240 000001 RENAM2: MOVE DAT,TAC ;SAVE LOC OF NAME IN DIRECTORY
192 000345 260140 000365' PUSHJ PDP,DSERCH ;SEARCH FOR NEW NAME
193 000346 334000 000000 SKIPA ;NOT FOUND - GOOD
194 000347 254000 000361' JRST RENER2 ;NAME ALREADY EXISTS - ERROR
195 000350 200060 000014 MOVE TAC,@UUO ;GET NEW NAME
196 000351 202045 000000 MOVEM TAC,(DAT) ;SAVE IN DIR
197 000352 350000 000014 AOS UUO
198 000353 200060 000014 MOVE TAC,@UUO ;EXTENSION
199 000354 502045 000026 HLLM TAC,26(DAT) ;SAVE IN DIR
200 000355 502046 000012 HLLM TAC,DEVEXT(DEVDAT) ;SAVE INN DDB
201 000356 254000 000341' JRST RENAM1 ;GIVE GOOD RETURN TO USER
202
203 000357 262140 000014 RENER1: POP PDP,UUO
204 000360 634040 000001 TDZA TAC,TAC ;RH E+1 =0
205 000361 201040 000004 RENER2: MOVEI TAC,4 ;RH E+1 =4
206 000362 350000 000014 AOS UUO ;POINT TO 2ND WORD
207 000363 542060 000014 HRRM TAC,@UUO ;SET ERRORR CODE
208 000364 263140 000000 POPJ PDP, ;AND TAKE ERROR RETURN
    
```

```

209                                     ;SEARCH DIRECTORY FOR A MATCH
210 000365 201654 000003 DSERCH: MOVEI AC1,3(UUO) ;CHECK VALIDITY OF ADDRESS
211 000366 260140 000000          PUSHJ PDP,UADCK1 ;NEVER RETURN IF ADDR, OUT OF BOUNDS
212 000367 260140 000410' DSER1: PUSHJ PDP,DIRCHK ;ENSURE DIRECTORY IS IN CORE
213 000370 550046 000014          HRRZ TAC,DL0C(DEVDAT) ;LOCATION OF DIRECTORY
214 000371 270040 002365'          ADD TAC,[XWD -26,NAMSTR] ;POINT TO START OF NAMES
215 000372 336120 000014          NMLOOK: SKIPM TAC1,@UUO ;GET NAME
216 000373 254000 000000          JRST TPOPJ ;NULL ARGUMENT - ERROR RETURN
217 000374 202106 000011          MOVEM TAC1,DEVFIL(DEVDAT) ;STORE FOR RENAME AND SUPERSEDING
218                                     ; SHARED SEGMENTS
219 000375 316101 000000          CAMN TAC1,(TAC) ;TEST FOR MATCH
220 000376 344600 000401'          AOJA UUO,NMFOUN ;FOUND NAME, CHECK EXTENSION
221 000377 253040 000375'          AOBJN TAC,.-2 ;TRY NEXT NAME
222 000400 263140 000000          POPJ PDP, ;NOT FOUND
223 000401 510120 000014          NMFOUN: HLLZ TAC1,@UUO ;PICK UP USER'S EXTENSION
224 000402 430101 000026          XOR TAC1,26(TAC) ;TEST AGAINST DIRECTORY EXTENSION
225 000403 607100 777777          TLNN TAC1,-1 ;MATCH?
226 000404 254000 000323'          JRST CPOPJ1 ;YES, RETURN
227 000405 252040 000407'          AOBJP TAC,.+2
228 000406 364600 000372'          SOJA UUO,NMLOOK ;NO, TRY NEXT NAME
229 000407 364600 000000          SOJA UUO,CPOPJ ;NAME NOT FOUND
230
231                                     ;CHECK IF DIRECTORY IS IN CORE, IF NOT, READ IT
232 000410 606000 000100          DIRCHK: TRNN IOS,UOSD ;DONT BOTHER IF NON-STANDARD
233 000411 337006 000004          SKIPG DEVMOD(DEVDAT) ;IS IT IN?
234 000412 263140 000000          POPJ PDP, ;YES, RETURN
235 000413 201200 000144          MOVEI BLK,DIRBLK ;BLOCK NUMBER
236 000414 621000 000020          TLZ IOS,IO
237 000415 260140 000742'          PUSHJ PDP,GETDT ;GET CONTROL
238 000416 661000 010000          TLO IOS,RNDIR ;JUST READ 1 BLOCK
239 000417 260140 001245'          PUSHJ PDP,READBC ;GO READ IT
240 000420 260140 000313'          PUSHJ PDP,WAIT1 ;WAIT TILL IN
241 000421 205100 400000          MOVSI TAC1,DVDIRI ;SET DIRECTORY-IN-CORE BIT
242 000422 436106 000004          ORM TAC1,DEVMOD(DEVDAT)
243 000423 263140 000000          POPJ PDP,
    
```

```

244                                     ;SEARCH DIRECTORY FOR FILE WHOSE INDEX IS IN TAC
245 000424 205240 440500 BLKSRC: MOVSI DAT,440500 ;DAT IS A BLOCK POINTER
246 000425 540246 000014 HRR DAT,DLOC(DEVDAT)
247 000426 201200 000001 MOVEI RLK,1 ;START AT BLOCK 1
248
249 000427 134400 000005 BLKSRA: ILDB TEM,DAT ;INDEX OF NEXT BLOCK
250 000430 316040 000010 CAMN TAC,TEM ;MATCH?
251 000431 254000 000404 JRST CPOPJ1 ;YES, RETURN
252
253 000432 305200 001101 BLKSRA: CAIGE BLK,TOPBLK ;NO, SEARCHED LAST?
254 000433 344200 000427 AOJA BLK,BLKSRA ;NO, TRY NEXT BLOCK
255 000434 263140 000000 POPJ PDP, ;YES, RETURN
256
257                                     ;SET UP POINTER TO DIRECTORY FOR BLOCK IN BLK
258 000435 261140 000004 SETPTR: PUSH PDP,BLK ;SAVE BLK
259 000436 260140 000552 PUSHJ PDP,DRPTR ;SET BLK AS A BYTE POINTER
260 000437 200240 000004 MOVE DAT,BLK ;RETURN IT IN DAT
261 000440 262140 000004 POP PDP,BLK ;RESTORE BLK
262 000441 263140 000000 POPJ PDP, ;AND RETURN

263
264                                     ;GET NEXT AVAILABLE FREE BLOCK
265 000442 261140 000010 NXTFRE: PUSH PDP,TEM
266 000443 260140 000435 PUSHJ PDP,SETPTR ;SET DAT TO A BYTE POINTER
267 000444 201040 000000 MOVEI TAC,0 ;LOOK FOR FREE BLOCKS
268 000445 260140 000427 PUSHJ PDP,BLKSRA ;FIND A ZERO BLOCK
269 000446 201200 000000 MOVEI BLK,0 ;NOT THERE- RETURN 0
270 000447 262140 000010 FREXIT: POP PDP,TEM
271 000450 263140 000000 POPJ PDP,

272
273                                     ;GET PREVIOUS FREE BLOCK
274 000451 201040 000000 LSTFRE: MOVEI TAC,0
275 000452 261140 000010 PUSH PDP,TEM
276 000453 271200 000002 ADDI BLK,2
277 000454 260140 000435 PUSHJ PDP,SETPTR ;SET DAT AS A POINTER
278 000455 275200 000002 SUBI BLK,2
279 000456 260140 000464 PUSHJ PDP,DECPTR ;DECREMENT BYTE POINTER
280 000457 135400 000005 LDB TEM,DAT ;INDEX TO BLOCK
281 000460 316400 000001 CAMN TEM,TAC ;FOUND?
282 000461 254000 000447 JRST FREXIT ;YES, RETURN
283 000462 367200 000456 SOJG BLK,-4 ;TRY AGAIN IF NOT AT START
284 000463 254000 000447 JRST FREXIT ;REACHED START - RETURN BLK=0
285
286                                     ;DECREMENT BYTE POINTER
287 000464 321240 000471 DECPTR: JUMPL DAT, +5
288 000465 270240 002366 ADD DAT,CBYTE (6) 5] ;DECREMENT
289 000466 327240 000407 JUMPG DAT,CPOPJ ;IF POSITIVE - SAME WORD
290 000467 505240 010500 HRLI DAT,010500 ;RESET TO PREVIOUS WORD
291 000470 364240 000466 SOJA DAT,CPOPJ
292 000471 505240 060500 HRLI DAT,060500
293 000472 364240 000470 SOJA DAT,CPOPJ
    
```

```

294                                     ;COME HERE TO DELETE THE FILE WHOSE INDEX IS INN TAC
295 000473 201100 000000 DLETF: MOVEI TAC1,0 ;SET TO DELETE BLOCKS
296 000474 260140 000424 PUSHJ PDP,BLKSRG ;FIND A BLOCK BELONGING TO FILE
297 000475 254000 000472 JRST CPOPJ ;ALL THROUGH
298 000476 137100 000005 DFB TAC1,DAT ;DELETE IT
299 000477 370003 000001 SOS 1(PDP) ;ADJUST PDL FOR RETURN
300 000500 253140 000432 AORJN PDP,BLKSRB ;AND FIND NEXT MATCH
301
302
303
304 000501 602000 000100 ;ENTER A FILE NAME IN DIRECTORY
305 000502 254000 000431 ENTR: TRNE IOS,UUSD ;NON STANDARD?
306 000503 260140 000365 JRST CPOPJ1 ;YES, RETURN
307 000504 254000 000562 PUSHJ PDP,DSERCH ;NO, LOOK FOR MATCH
308 000505 200120 000014 JRST NEWENT ;THIS IS A NEW ENTRY
309 000506 350000 000014 ENTR2: MOVE TAC1,@UUO ;PICK UP 2ND WORD (EXTENSSION)
310 000507 540120 000014 AOS UUO ;POINT TO WORD 3
311 000510 606100 007777 HRR TAC1,@UUO ;ADD DATE
312 000511 434100 000000 TRNN TAC1,7777 ;IS DATE ALREADY THERE?
313 000512 202101 000026 IOR TAC1,THSDAT ;NO, ADD CURRENT DATE
314 000513 335746 000016 MOVEM TAC1,26(TAC) ;INTO DIRECTORY
315 SKIPGE AC3,OBLK(DEVDAT) ;IS THIS A SAVE FILE (UGETF DONE
316 000514 344600 000567 ;BEFORE THE ENTER?)
317 000515 275600 000002 ENTR3: AOJA UUO,SETW04 ;YES, STORE LENGTH IN DIRECTORY
318 000516 200120 000014 SUBI UUO,2 ;NO, POINT TO NAME
319 000517 202101 000000 MOVE TAC1,@UUO ;PICK IT UP
320 000520 574040 000001 MOVEM TAC1,(TAC) ;INTO DIRECTORY
321 000521 271040 000027 HLR TAC,TAC ;COMPUTE INDEX OF FILE
322 000522 554246 000015 ADDI TAC,27
323 000523 274240 000001 HLRZ DAT,IBLK(DEVDAT) ;INDEX OF INPUT FILE
324 000524 322240 000475 SUB DAT,TAC ;WRITING SAME FILE AS READING?
325 000525 506046 000016 JUMPE DAT,CPOPJ ;TAKE ERROR RETURN IF YES
326 HRLM TAC,OBLK(DEVDAT) ;SAVE INDEX IN DDB
327 000526 260140 000473 PUSHJ PDP,DELETE ;DELETE ALL BLOCKS BELONGING TO FILE
328 000527 342740 000577 AOJE AC3,FNTRD ;FIND FIRST FREE BLOCK ON TAPE IF THIS
329 ;IS A SAVE FILE (UGETF DONE)
330 000530 201200 000144 MOVEI BLK,DIRBLK ;NO, GET 1ST BLOCK CLOSE TO
331 000531 661000 001000 TLO IOS,RVERSE ;DIRECTORY, GOING IN REVERSE
332 000532 260140 000644 PUSHJ PDP,U$LSTA
333 000533 303200 001101 CAILE BLK,TOPBLK ;BLOCK LEGAL?
334 000534 263140 000000 POPJ PDP, ;NO, ERROR RETURN
335 000535 202206 000013 ENTRC: MOVEM RLK,FSTBLK(DEVDAT) ;SAVE AS 1ST BLOCK
336 000536 542206 000016 HRRM RLK,OBLK(DEVDAT) ;SAVE IN DDB
337 000537 350000 000014 AOS UUO ;POINT UUO TO WORD 2
338 000540 542220 000014 HRRM RLK,@UUO ;SAVE 1ST BLOCK IN USER'S AREA
339 000541 500060 000014 HLL TAC,@UUO ;GET EXTENSION
340 000542 502046 000012 HLLM TAC,NEVEXT(DEVDAT) ;SAVE EXTENSION IN DDB ALSO
341 000543 661000 000200 TLO IOS,NOLINK
342 000544 350003 000000 AOS (PDP)
    
```

```

343
344 000545 260140 000552' ;MARK DIRECTORY ENTRY POINTED TO BY BLK AS TAKEN
345 000546 554046 000016 MARKDR: PUSHJ PDP,DRPTR ;SET POINTER TO BLOCK IN DIR
346 000547 136040 000004 HLRZ TAC,0BLK(DEVDAT) ;PICK UP INDEX
347 000550 661000 000400 IDPB TAC,BLK ;MARK DIRECTORY
348 000551 254000 000343' TLO IOS,CHNGDR ;DIRECTORY HAS CHANGED
349 JRST STOIOS
350
351 000552 275200 000001 ;;SET POINTER TO CORRECT DIRECTORY ENTRY
352 000553 231200 000007 DRPTR: SUBI BLK,1 ;SET FOR ILDB OR IDPB
353 000554 270206 000014 IDIVI BLK,7 ;COMPUTE WORD, POSITION
354 000555 505200 440500 ADD BLK,0LOC(DEVDAT) ;GET CORRECT ADDRESS
355 000556 322240 000524' HRLI BLK,440500 ;MAKE IT A BYTE POINTER
356 000557 133000 000004 JUMPE DAT,CPOPJ ;CORRECT FOR POSITION IN WORD
357 000560 367240 000557' IBP BLK
358 000561 263140 000000 SOJG DAT,-1
359 POPJ PDP,
360
361 000562 274040 002367' ;HERE FOR NEW FILE NAME ON ENTER
NEWENT: SUB TAC,EXWD 26,26];START AT BEGINNING OF DIRECT.
362 000563 336001 000000 SKIPN (TAC) ;FIND A FREE SLOT
363 000564 344600 000505' AOJA UUO,ENTR2 ;RETURN WITH UUO POINTING TO WRD 2
364 000565 253040 000563' AOBJN TAC,-2
365 000566 263140 000000 POPJ PDP, ;NONE AVAILABLE,
366
367
368
369 000567 574120 000014 ;SET UP LENGTH OF FILE IN DIRECTORY FOR A SAVE FILE
370 000570 213000 000002 SETWD4: HLRE TAC1,@UUO ;GET -LENGTH
371 000571 570420 000014 MOVNS TAC1 ;+LENGTH
372 000572 270100 000010 HRRE TEM,@UUO
373 000573 620100 001777 ADD TAC1,TEM ; +START ADDRESS
374 000574 242100 000002 TRZ TAC1,1777 ;STORE N-1, WHERE N IS NO OF K
375 000575 436101 000026 LSH TAC1,2
376 000576 364600 000515' ORM TAC1,26(TAC) ;INTO 2ND WRD OF DIRECTORY
377 SOJA UUO,ENTRA ;CONTINUE WITH ENTER
378 000577 201040 000000 ENTRD: MOVEI TAC,0 ;GET THE 1ST FREE BLOCK ON TAPE
379 000600 260140 000424' PUSHJ PDP,BLKSRC ;AS THE 1ST LOGICAL BLOCK OF THE FILE
380 000601 263140 000000 POPJ PDP, ;NONE AVAILABLE
381 000602 254000 000535' JRST ENTRC ;CONTINUE WITH ENTER
    
```

```

382                                     ;USETI - SET NEXT INPUT BLOCK TO READ
383 000603 630000 002370' SETI:  TDZ      IOS,[XWD IOEND,IOEND]
384 000604 334240 000006      SKIPA   DAT,DEVDAT
385
386                                     ;USETO - SET NEXT OUTPUT BLOCK TO READ
387 000605 201246 000001' SETO:  MOVEI   DAT,1(DEVDAT)
388 000606 260140 000420'      PUSHJ  PDP,WAIT1      ;WAIT FOR BUFFERS TO FILL (OR EMPTY)
389 000607 542605 000015'      HRRM   UUU,IBLK(DAT)  ;SET BLOCK NUMBER
390 000610 254000 000551'      JRST   STOIOS        ;STOE IOS, POPJ
391
392                                     ;UGETF - GET NEXT FREE BLOCK FOR THIS FILE
393 000611 260140 000606' GETF:  PUSHJ  PDP,WAIT1      ;WAIT TILL BUFFERS EMPTY
394 000612 260140 000410'      PUSHJ  PDP,DIRCHK     ;ENSURE DIR, IN CORE
395 000613 260140 000620'      PUSHJ  PDP,USRFRE     ;GET NEXT AVAILABLE BLOCK
396 000614 336006 000016'      SKIPN  OBLK(DEVDAT)  ;HAS AN ENTER OR LOOKUP BEEN DONE?
397 000615 477206 000016'      SETOR  BLK,OBLK(DEVDAT) ;NO, SET SWITCH SO THAT THE NEXT ENTER
398                                     ;WILL FINF FIRST FREE BLOCK ON TAPE
399 000616 200040 000004'      MOVE   TAC,BLK      ;TELL USER THE BLOCK NUMBER
400 000617 254000 000000'      JRST   STOTAC
401
402
403                                     ;GET NEXT (OR PREVIOUS) FREE BLOCK
404 000620 201400 000004' USRFRE: MOVEI   TEM,SPACE      ;BLOCKS "SPACE" APART
405 000621 135200 000000'      LDB    BLK,PIOMOD     ;EXCEPT DUMP AND SAVMOD FILES
406 000622 301200 000015'      CAIL   BLK,SD        ;OR ONE OF DUMP MODES?
407 000623 201400 000002'      MOVEI  TEM,2        ;YES, WHICH ARE CLOSER
408 000624 550206 000016' USRFRA: HRRZ   BLK,OBLK(DEVDAT) ;CURRENT BLOCK
409 000625 603000 001000'      TLNE  IOS,RVERSE    ;FORWARD?
410 000626 254000 000641'      JRST  USRLST        ;NO
411 000627 271210 000000'      ADDI  BLK,(TEM)     ;YES, FIND NEXT BLOCK AT LEAST N
412 000630 303200 001101'      CAILE  BLK,TOPLBK
413 000631 634200 000004'      TDZA  BLK,BLK
414 000632 260140 000442' CALNXT: PUSHJ  PDP,NXTFRE     ;BLOCKS PAST THIS ONE
415 000633 326200 000610'      JUMPN  BLK,STOIOS    ;RETURN IF FOUND
416 000634 663400 000001'      TLOE  TEM,1        ;FOUND NONE ON THIS PASS
417 000635 254000 000646'      JRST  NOBLKS       ;TAPE IS FULL
418 000636 641000 001000'      TLC   IOS,RVERSE    ;REVERSE DIRECTION
419 000637 541400 000001'      HRRI  TEM,1        ;START LOOKING AT NEXT BLOCK IN OTHER DIRECTION
420 000640 254000 000624'      JRST  USRFRA
421 000641 275210 000000' USRLST: SUBI  BLK,(TEM)     ;LOOK FOR FREE BLOCK N BEFORE
422 000642 337000 000004'      SKIPG BLK
423 000643 634200 000004'      TDZA  BLK,BLK
424 000644 260140 000451' USLSTA: PUSHJ  PDP,LSTFRE     ;THIS ONE
425 000645 254000 000633'      JRST  CALNXT*1
426
427
428                                     ;NO FREE BLOCKS AVAILABLE, GIVE HIGH BLOCK,SET IOBKYL LATER
429 000646 201200 001102' NOBLKS: MOVEI  BLK,TOPLBK*1 ;SET HIGH BLOCK
430 000647 263140 000000'      POPJ  PDP,
    
```

```

431                                     ;UTPCLR UUO
432 000650 602000 000100 UTPCLR: TRNE IOS,UDSD
433 000651 263140 000000 POPJ PDP, ;FORGET IT FOR NON-STANDARD
434 000652 205040 400000 MOVSI TAC,DVDIRIN ;SET DIRECTORY-IN-CORE BIT
435 000653 436046 000004 ORM TAC,DEVMOD(DEV DAT)
436 000654 661000 000400 TLO IOS,CHNGDR ;DIRECTORY HAS CHANGED
437 000655 550046 000014 HRRZ TAC,DLOC(DEV DAT) ;LOC OF DIRECTORY
438 000656 504100 000001 HRL TAC1,TAC
439 000657 541101 000001 HRRI TAC1,1(TAC) ;BLT POINTER
440 000660 402001 000000 SETZM (TAC)
441 000661 251101 000176 BLT TAC1,176(TAC) ;LEAVE LAST WORD IN DIR, ALONE
442 000662 205100 017000 MOVSI TAC1,17000 ;MARK DIRECTORY AS UNAVAILABLE
443 000663 202101 000016 MOVEM TAC1,16(TAC)
444 000664 205100 757000 MOVSI TAC1,757000 ;RESERVE BLOCKS 1 AND 2
445 000665 202101 000000 MOVEM TAC1,(TAC) ;FOR READ IN MODE LOADER
446 000666 205100 777770 MOVSI TAC1,777770 ;MARK BLOCKS 1102-1105 AS
447 000667 456101 000122 ORCAM TAC1,NAMSTR-1(TAC) ;UNAVAILABLE ALSO
448 000670 254000 000633 JRST STOIOS
    
```



```

472                                     ;RELEASE UUD
473 000712 260140 000611' UREL:  PUSHJ  PDP,WAIT1      ;MAKE SURE THE TAPE IS STOPPED
474 000713 260140 001235'         PUSHJ  PDP,NXTCM2     ;CLEAR OUT DUMP-MODE STUFF
475 000714 205040 400000         MOVSI  TAC,DVDIRIN   ;IF NONSTANDARD, WILL CLEAR
476 000715 622000 000100         TRZE  IOS,UDSD     ;CLEAR NON-STANDARD BIT,
477 000716 412046 000004         ANDCAM TAC,DEVMOD(DEV DAT) ;SO DIRECTORY WILL BE READ ANEW
478 000717 337006 000004         SKIPG  DEVMOD(DEV DAT) ;IF DIRECTORY HAS BEEN
479 000720 627000 000400         TLZA  IOS,CHNGDR   ;MODIFIED IT MUSY RE WRITTEN
480 000721 254000 000730'         JRST  UREL2        ;IT HASN'T BEEN CHANGED
481 000722 661000 000020         TLO   IOS,IO
482 000723 260140 000742'         PUSHJ  PDP,GETDT    ;WAIT TILL DTC AVAILABLE
483 000724 661000 010000         TLO   IOS,RWDIR   ;GOING TO WRITE DIRECTORY
484 000725 201200 000144         MOVEI  BLK,DIRBLK  ;BLOCK NUMBER
485 000726 260140 001110'         PUSHJ  PDP,WRTBLK  ;WRITE UT
486 000727 260140 000712'         PUSHJ  PDP,WAIT1    ;WAIT TILL IT HAS BEEN WRITTEN
487
488
489 000730 200046 000007 UREL2:  MOVE   TAC,DEVIAD(DEV DAT) ;BITS 1,2 ARE COUNT OF CHANS
490 000731 603040 200000         TLNE  TAC,200000   ;DEV INITED ON ANOTHER CHANNEL TOO?

491 000732 263140 000003 QUANTL: POPJ   PDP,QUANT      ;YES, DONT ZAP IOS OR ODB
492
493                                     ;SOME BITS IN THE IOS WORD AND THE ODB WILL NORMALLY BE CHANGED ON THE
494                                     ;INTERRUPT LEVEL AFTER THE RELEASE, BUT HNGSTP CAN CAUSE THESE ACTIONS
495                                     ;NEVER TO OCCUR, SO MAKE SURF THEY REALLY HAPPENED
496 000733 621000 077600         TLZ   IOS,77600   ;ZERO IOS BITS
497 000734 402006 000016         SETZM OBLK(DEV DAT) ;AND OBLK
498 000735 402006 000015         SETZM IBLK(DEV DAT) ;ZERO IBLK SO WRITING A FILE AFTER READING
499                                     ;IT WILL WORK (CHECK IS MADE AT ENTER)
500 000736 402006 000013         SETZM FSTBLK(DEV DAT)
501 000737 254000 000673'         JRST  STIOS       ;STORE IOS AND RETURN
502
503

```

```

504 000740 260140 000000 GETDT0: PUSHJ PDP,SETACT ;WAIT TILL TAPE COMES OUT OF REWIND
505 000741 260140 000000 PUSHJ PDP,WSYNG ;BEFORE DOING ANYTHING ELSE TO IT
506 ;GET DEC TAPE CONTROLLER
507 000742 603000 100000 GETDT: TLNE IOS,REWBIT ;IF TAPE IS REWINDING NOW
508 000743 254000 000740' JRST GETDT0 ;WAIT TILL THRU BEFORE CONTINUING
509 000744 352000 000000 GETDT1: AOSE DTREQ ;CAN I HAVE IT
510 000745 260140 000000 PUSHJ PDP,OTWAIT ;NO. COME BACK LATER
511 000746 135040 000000 LDR TAC,PUNIT ;HAVE CONTROL NOW
512 000747 242040 000011 LSH TAC,11 ;CONNECT TO DTA
513 000750 732201 030000 CONO DTC,30000(TAC) ;SEE IF TAPE IS OK
514 000751 732700 000100 CONSZ DTS,100 ;SELECT ERROR?
515 000752 254000 000763' JRST QUEST ;YES. COMPLAIN
516 000753 603000 000020 TLNE IOS,IO ;NO. TRYING TO WRITE TAPE?
517 000754 732740 004000 CONSO DTS,4000 ;YES. WRITE PROTECTED?
518 000755 334040 000732' SKIPA TAC,QUANTL ;NO. EVERYTHING IS OK
519 000756 254000 000763' JRST QUEST ;YES. COMPLAIN
520 000757 552040 002151' HRRZM TAC,QUANTM ;SET UP NUMBER OF BLOCKS TO KEEP CONTROL FOR
521 000760 732200 010000 CONO DTC,10000 ;DESELECT CONTROL SO FNDBLK TEST
522 ;WILL WORK RIGHT (CONCO DTC,20000)
523 000761 202300 002142' MOVEM DEVDAT,USEWRD ;SAVE ACS NEEDED ON INTERRUPT LEVEL
524 000762 254000 000740' JRST SETACT ;LIGHT IOACT AND RETURN
525
526 ;COME HERE TO COMPLAIN ABOUT UNIT NOT RIGHT (SELECT OR PROTECT ERROR)
527 000763 260140 001747' QUEST: PUSHJ PDP,THRUTD ;GIVE UP CONTROL
528 000764 260140 000000 PUSHJ PDP,HNGSTP ;TYPE "DTAN OK?"
529 ;NOTE -- AC BLK (=ITEM) USED TO BE
530 ; PUSHED HERE, BUT IS NOT ANY LONGER
531 ; BECAUSE HNGSTP NOW PRESERVES IT,
532 ; AND THE STACK IS VERY FULL.
533 000765 254000 000744' JRST GETDT1 ;GET CONTROL AGAIN AND RETRY
    
```

```

534                                     EXTERN  JOBDDT,USRDDT
535                                     ;DUMP MODE INPUT
536 000766                                     DMP1:  IFN      CPBIT, <
537 000766 550706 000015                       HRRZ      AC2,IBLK(DEV DAT)
538                                     >
539 000767 621000 000020                       TLZ      IOS,IO
540 000770 260140 001134'                     PUSHJ    PDP,DMPSET      ;SET UP DUMP-MODE STUFF
541 000771 254000 001024'                     JRST     ZERCOR         ;ZERO USER'S CORE IF SAVE-MODE
542
543
544                                     ;INPUT UO
545 000772 621000 000020 UIN:  TLZ      IOS,IO
546 000773 550206 000015                       HRRZ      BLK,IBLK(DEV DAT) ;BLOCK TO READ
547 000774 260140 000737'                     PUSHJ    PDP,STOIOS
548 000775 602000 000100                       TRNE     IOS,UDSD
549 000776 254000 001243'                     JRST     READRF        ;NON STANDAR?
550 000777 322200 001020'                     JUMPE   BLK,EOF        ;0 MEANS EOF
551 001000 260140 001014'                     PUSHJ    PDP,BLKCHK    ;CHECK LEGALITY OF BLOCK NUMBER
552 001001 607000 004000                       TLNN     IOS,DMPMOD    ;DUMP MODE?

553 001002 302200 000144                       CAIE     BLK,DIRBLK    ;TRYING TO READ DIRECTORY?
554 001003 254000 001243'                     JRST     READRF        ;NO, GO READ
555
556                                     ;READING DIRECTORY - GIVE CORE IMAGE IF IT EXISTS
557 001004 260140 000410'                     PUSHJ    PDP,DIRCHK    ;READ IT IF IT ISN'T IN ALREADY
558 001005 504046 000014                       HRL      TAC,DLOC(DEV DAT) ;LOC OF DIRECTORY
559 001006 201126 000007                       MOVEI    TAC1,@DEVIAD(DEV DAT) ;WHERE USER WANTS IT
560 001007 541042 000001                       HRRI    TAC,1(TAC1)    ;LOC OF DATA
561 001010 251042 000200                       BLT      TAC,200(TAC1) ;GIVE IT TO HIM
562 001011 260140 000000                       PUSHJ    PDP,ADVBF     ;ADVANCE BUFFERS
563 001012 255000 000000
564 001013 263140 000000                       POPJ     PDP,
    
```

```

565                                     ;CHECK VALIDITY OF BLOCK NUMBER
566 001014 307200 001171 BLKCHK: CAIG   RLK, TOPBLK   ;LEGAL?
567 001015 263140 000000          POPJ   PDP,          ;YES. RETURN
568 001016 262140 000001          POP   PDP, TAC
569 001017 664000 040000          TRQA   IOS, IORCTL   ;NO. LIGHT ERROR BIT
570
571                                     ;INPUT BLOCK = 0 - END OF FILE
572 001020 661000 000040 EOF:   TLO   IOS, IOEND   ;LIGHT EOF BIT
573 001021 607000 004000          TLNN   IOS, DMPMOD
574 001022 254000 000774'         JRST   STOIOS
575 001023 254000 002073'         JRST   DMPEOF   ;GIVE UP CONTROL IF DUMP-MODE
576
577
578                                     ;ZERO USER'S CORE ON SAVE-MODE INPUT
579 001024 201047 000000 ZERCOR: MOVEI  TAC, JOBDDT(PROG) ;ZERO CORE
580 001025 505041 000001          HRLI  TAC, 1(TAC)
581 001026 207000 000001          MOVSS TAC          ;BLT POINTER
582 001027 550106 000020          HRRZ  TAC1, DMPPLST(DEVDAT) ;TOP CELL TO ZERO (-175)
583 001030 271107 000000          ADDI  TAC1, (PROG) ;RELOCATE TO USER AREA
584 001031 402001 777777          SETZM -1(TAC)      ;ZERO
585 001032 251042 000000          BLT  TAC, (TAC1)
586 001033 402000 000000          SETZM USRDDT      ;DDT IS KEPT IN PROTECTED PART
587 001034 254000 000772'         JRST  UIN

```

```

588                                     ;DUMP MODE OUTPUT
589 001035 260140 000410' DMPO: PUSHJ PDP,DIRCHK ;MAKE SURE DIRECTORY IS IN CORE
590                                     IFN CPBIT, <
591 001036 550706 000016 HRRZ AC2,0BLK(DEV DAT)
592                                     >
593 001037 661000 000020 TLO IOS,IO
594 001040 260140 001134' PUSHJ PDP,DMPSET ;SET DUMP-MODE POINTERS
595 001041 255000 000000 JFCL
596
597                                     ;OUTPUT UO0
598 001042 661000 000020 UOUT: TLO IOS,IO
599 001043 602000 000100 TRNE IOS,UDSD ;NON STANDARD?
600 001044 254000 001054' JRST UOUT2 ;YES
601 001045 260140 000410' PUSHJ PDP,DIRCHK ;NO. MAKE SURE DIRECTORY IS IN CORE
602 001046 661000 000020 TLO IOS,IO ;IF DIRCHK READ, IO WENT OFF
603 001047 550206 000016 HRRZ BLK,0BLK(DEV DAT)
604 001050 306200 000144 CAIN BLK,DIRBLK ;CHECK IF WRITING DIRECTORY
605 001051 254000 001115' JRST COR2HM ;YES. WRITE CORE IMAGE
606 001052 322200 001125' JUMPE BLK,FAKADV ;DONT WRITE IF NO BLOCK GIVEN

607 001053 260140 001014' PUSHJ PDP,RLKCHK ;CHECK FOR LEGAL BLOCK
608 001054 607000 004000 UOUT2: TLNN IOS,DMPMOD ;ALREADY HAVE CONTROL IF DUMP-MODE
609 001055 260140 000742' PUSHJ PDP,GETOT ;GET DEC TAPE CONTROLLER
610 001056
611 001056 603000 004000 DTOCHK: TLNE IOS,DMPMOD ;DUMP MODE?
612 001057 254000 001175' JRST DMPFIL ;YES. FILL BUFFER FROM LIST
613 001060 205066 000010 MOVSI TAC,@DEV0AD(DEV DAT) ;LOCATION OF BUFFER
614 001061 270040 002371' ADD TAC,[XWD 1,BUF];SET TO STORE IN MONITOR BUFFER
615 001062 251040 002363' BLT TAC,BUF+177 ;GO BLT IT
616 001063 621000 000200 TLZ IOS,NOLINK
    
```

```

617 001064 602000 000100 OUFULL: TRNE IOS,UDSD ;NON-STANDARD?
618 001065 254000 001106' JRST OUTBL2 ;YES, NO FILE-STRUCTURED OPERATIONS
619 001066 574200 002164' HLRE BLK,RUF ;IS IT?
620 001067 321200 001113' JUMPL BLK,LSTBLK ;YES, - LAST BLOCK OF FILE
621 001070 326200 001075' JUMPN BLK,OUTBLK ;IF NON-0 - YES
622 001071 603000 020000 TLNE IOS,DMPCLS ;NO, LAST BLOCK OF A DUMP FILE?
623 ;DMPCLS WILL BE TURNED OFF AT THE INTERRUPT
624 001072 254000 001075' JRST OUTBLK ;YES, LINK MUST STAY 0
625 001073 260140 000620' OUCOMP: PUSHJ POP,USRFRE ;COMPUTE NEXT BLOCK
626 001074 661000 000200 TLO IOS,NOLINK ;THIS BLOCK NOT LINKED
627 001075 506200 002164' OUTBLK: HRLM BLK,RUF ;SAVE LINK IN 1ST WORD OF BLOCK
628 001076 200406 000013 MOVE TEM,FSTBLK(DEVDAT) ;STORE 1ST BLOCK OF FILE IN WORD
629 001077 137400 002372' DPP TEM,[POINT 10,RUF,27]
630 001100 550406 000016 HRRZ TEM,0BLK(DEVDAT) ;BLOCK TO WRITE NOW
631 001101 542206 000016 HHRM BLK,0BLK(DEVDAT) ;BLOCK TO WRITE NEXT
632 001102 200200 000010 MOVE BLK,TEM
633 001103 260140 001014' PUSHJ POP,BLKCHK ;CHECK LEGALITY OF BLOCK
634 001104 260140 000545' PUSHJ POP,MARKDR ;MARK BLOCK TAKEN IN DIRECTORY
635 001105 334200 000010 SKIPA BLK,TEM
636 001106 550206 000016 OUTBL2: HRRZ BLK,0BLK(DEVDAT)
637 001107 260140 001022' PUSHJ POP,STOIOS
638
639
640 001110 260140 001341' WRTBLK: PUSHJ PDP,FNOBLK ;GO SEARCH FOR BLOCK
641 001111 200100 002373' MOVE TAC1,CBLKO DTC,700] ;HERE WE ARE - GO WRITE
642 001112 254000 001247' JRST RDWRT

```

```

643                                     ;WRITE LAST BLOCK
644 001113 201200 000000 LSTBLK: MOVEI  BLK,0          ;LINK=0
645 001114 254000 001075' JRST:  OUTBLK          ;GO WRITE LAST BLOCK
646
647                                     ;TRYING TO WRITE DIRECTORY - STORE IN CORE
648 001115 201066 000010 COR2HM: MOVEI  TAC,@DEVOAD(DEV DAT) ;WHERE IT IS
649 001116 505041 000001 HRLI:  TAC,1(TAC)
650 001117 540046 000014 HRR:   TAC,DLOC(DEV DAT)          ;WHERE TO PUT IT
651 001120 201101 000177 MOVEI  TAC,177(TAC)
652 001121 251042 000000 BLT:   TAC,(TAC1)
653 001122 661000 000400 TLO:  IOS,CHNGDR          ;REMEMBER TO WRITE IT OUT
654 001123 205100 400000 MOVSI  TAC1,DVDIRI
655 001124 436106 000004 ORM:   TAC1,DEVMOD(DEV DAT) ;DIR, IS NOW IN CORE
656 001125 623000 004000 FAKADV: TLZE  IOS,DMPMOD          ;DUMP MODE?
657 001126 254000 001747' JRST:  THRU TD          ;YES, GIVE UP CONTROL
658 001127 260140 000000 PUSHJ  PDP,ADVBFE          ;ADVANCE BUFFERS
659 001130 255000 000000 JFCL
660 001131 621000 000200 TLZ:  IOS,NOLINK          ;DIRECTORY BLOCK IS NOT LINKED
661 001132 402006 000016 SETZM  OBLK(DEV DAT)
662 001133 254000 001107' JRST:  STOIOS

```

```

663
664 001134 260140 000742' ;SET UP POINTERS AND STUFF FOR DUMP-MODE
665 001135 661010 004000' DMPSET: PUSHJ PDP,GETDT ;GET CONTROL
666 001136 260140 000000' TLO IOS,DMPMOD ;LIGHT BIT
667 001137 254000 002075' PUSHJ PDP,COMCHK ;CHECK VALIDITY OF LIST
668 001140 331060 000014' JRST SVADR ;ING. GIVE ADDRESS ERROR
669 001141 254000 001147' SKIPL TAC,@UUO ;OK, NULL LIST?
570 JRST DMPTS1 ;YES, RETURN
671 001142 602000 000100' IFN CPBIT, <
672 001143 364240 001151' TRNE IOS,UUSD ;NO. NON-STD MODE?
673 > SOJA DAT,TOUSFR ;YES. GO ELSEWHERE
674 001144 370000 000014' DMPST2: SOS UUO ;NO. SAVE START OF LIST (-1)
675 001145 202600 000020' MOVEM UUO,DMPLST(DEFDAT)
676 001146 254000 000502' JRST CPOPJ1
677
678 001147 262140 000001' DMPTS1: POP PDP,TAC
679 001150 254000 001747' JRST THRU TD
680
681
682
683 IFN CPBIT, <
684 001151 322700 001172' ;HERE TO START DUMP-MODE INTO USER AREA DIRECTLY
685 001152 240240 777771' TOUSFR: JUMPE AC2,NORLK0 ;CANT READ BLK 0 IN NON-STD DUMP MODE
686 001153 350000 000005' ASH DAT,-7 ;NUMBER OF WRDS IN LIST /200
687 001154 202240 002157' AOS DAT
688 001155 201620 000014' MOVEM DAT,BLKCNT ;SAVE TO UPDATE POSITION
689 001156 202600 002161' TOUSR1: MOVEM UUO,@UUO ;REAL ADDRESS OF LIST
690 001157 202600 002160' MOVEM UUO,SVPNTR ;SAVE IT
691 001160 271047 000000' ADDI TAC,(PROG) ;RELOCATE ADDRESS OF 1ST IOWN
692 001161 202040 002145' MOVEM TAC,PNTR ;AND SAVE IT
693 001162 200040 002374' MOVE TAC,[JSR DMPADV] ;SET UP LOC FOR WHEN
694 001164 552340 002144' MOVEM TAC,DALC2 ;IOWN IS EXHAUSTED
695 001165 661000 040000' HRRZM PROG,ADRPRG ;SAVE JUST ADDRESS OF PROG
696 001166 607000 000020' TLO IOS,NORUF ;INDICATE DIRECTLY TO USER
697 001167 254000 001146' TLNN IOS,IO
698 001170 262140 000001' JRST CPOPJ1 ;READING - CONTINUE
699 001171 254000 001106' POP PDP,TAC ;WRITING - THIS WILL SAVE LOTS OF TIME
700 JRST OUTBL2
701
702
703 001172 660000 400000' NORLK0: TRO IOS,IOIMPM
704 001173 262140 000001' POP PDP,TAC
705 001174 254000 001747' JRST THRU TD ;RETURN TO UUOCON WITH ERROR BIT SET
706 >
    
```

```

707                                     ;FILL OUTPUT BUFFER FROM LIST
708 001175 205100 777601 DMPFIL: MOVSI TAC1,-177
709                                     IFE CPBIT, <
710                                     TRNE IOS,UDSD
711                                     SUB TAC1,ONEONE ;200 DATA WORDS IF NON-STANDARD
712 >
713 001176 260140 001224' DMPFLB: PUSHJ PDP,NXTCOM ;GET NEXT COMMAND
714 001177 254000 001205' JRST DMPOTH ;END OF LIST
715 001200 200401 000000 DMPFLA: MOVE TEM,(TAC) ;GET NEXT WORD
716 001201 202402 002165' MOVEM TEM,BUF+1(TAC1) ;INTO BUFFER
717 001202 252100 001215' AORJP TAC1,DMPOVR ;BUFFER FULL IF GOES
718 001203 253040 001200' AORJN TAC,-3 ;GET NEXT WORD FROM COMMAND
719 001204 254000 001176' JRST DMPFLB ;GET NEXT COMMAND
720
721 001205 DMPOTH:
722 IFE CPBIT, <
723 TRNN IOS,UDSD
724 >
725 001205 552100 002164' HRRZM TAC1,BUF ;LIST RAN OUT SAVE WORD COUNT
726 001206 402002 002165' SETZM BUF+1(TAC1) ;ZERO REST OF BUFFER
727 001207 551102 002166' HRRZI TAC1,BUF+2(TAC1)
728 001210 303100 002363' CAILE TAC1,BUF+177 ;JUST ZERO 1 WORD IF AT TOP
729 001211 254000 001064' JRST OUFULL
730 001212 505102 777777' HRLI TAC1,-1(TAC1)
731 001213 251100 002363' BLT TAC1,BUF+177 ;****TEST IF TOP OF BUFFER
732 001214 254000 001064' JRST OUFULL ;NOW WRITE BUFFER
733
734 ;BUFFER FULL BEFORE END OF COMMAND
735 001215 253040 001220' DMPOVR: AOBJN TAC,+3 ;WAS THAT LAST WORD OF COMMAND?
736 001216 260140 001224' PUSHJ PDP,NXTCOM ;YES, GET NEXT
737 001217 201040 000000 MOVEI TAC,0
738 001220 202046 000021 MOVEM TAC,SVDWRD(DEVDAT) ;NO, SAVE REMAINDER OF COMMAND
739 001221 201040 000177 DMPOVA: MOVEI TAC,177
740 IFE CPBIT, <
741 TRNN IOS,UDSD
742 >
743 001222 202040 002164' MOVEM TAC,BUF ;WD CNT =177
744 001223 254000 001064' JRST OUFULL ;GO WRITE PART OF STUFF
745
746
747 ;GET NEXT COMMAND FROM LIST
748 001224 336006 000020 NXTCOM: SKIPN DMPLST(DEVDAT) ;END OF COMMANDS?
749 001225 254000 001235' JRST NXTCM2 ;YES, RETURN
750 001226 354046 000020 AOSA TAC,DMPLST(DEVDAT) ;GET NEXT COMMAND
751 001227 542046 000020 NXTCM1: HRRM TAC,DMPLST(DEVDAT) ;STORE GO-TO ADDRESS
752 001230 200060 000001 MOVE TAC,@TAC ;GET COMMAND
753 001231 322040 001235' JUMPE TAC,NXTCM2 ;END OF LIST
754 001232 327040 001227' JUMPG TAC,NXTCM1 ;GO-TO WORD
755 001233 271047 000000 ADDI TAC,(PROG) ;REAL COMMAND - ADD RELOCATION
756 001234 344040 001167' AOJA TAC,CPOPJ1 ;AND RETURN
757
    
```

```
758                                     ;END OF DUMP-MODE LIST
759 001235 402026 000021 NXTCM2: SETZM  SVDWRD(DFVDAT)  ;ZERO POINTERS
760 001236 402006 000020          SETZM  DMPLST(DFVDAT)
761 001237 263140 000000          POPJ    PDP,
762
763
764 001240 332046 000021 DMPFLC: SKIPF  TAC,SVDWRD(DFVDAT)  ;IS THERE ANOTHER COMMAND
765 001241 254000 001200'         JRST  DMPFLA  ;YES, GET IT
766 001242 254000 002032'         JRST  DMPH2   ;NO, THROUGH
```

```

767          000320 DTC=320
768          000324 DTS=324
769
770          ;IO INTERFACE
771 001243 607000 004000 READRF: TLNN IOS,OMPMOD ;HAVE CONTROL IF DUMP-MODE
772                                     ; UNLESS ON A LOOKUP, IN WHICH
773                                     ; CASE ALWAYS GET DT CONTROL
774 001244 260140 000742' RDBLUK: PUSHJ PDP,GETDT ;GET DT CONTROL
775
776 001245 260140 001341' READBC: PUSHJ PDP,FNDBLK ;SEARCH FOR RIGHT BLOCK
777 001246 200100 002375' MOVE TAC1,[BLK] DTC,300] ;FOUND IT - START READING
778
779          ;HERE WITH BLK=BLOCK NUMBER, TAC1=FUNCTION, START SEARCH
780 001247 502100 001275' RDWRT: HLLM TAC1,IOWD ;BLKI OR BLKO
781 001250 505040 777600 HRLI TAC,-200
782 001251 607000 010000 TLNN IOS,RWDIR ;WRITING (READING) DIRECT?
783 001252 334040 002140' SKIPA TAC,BFPNTR ;NO. INTO BUF
784 001253 540046 000014 HRR TAC,DLOC(DEVDAT) ;YES. LOC OF DIRECTORY
785 001254 732300 100000 CONSZ DTC,100000 ;IN REVERSE?

786 001255 254000 001271' JRST IORVRS ;YES
787 001256 370000 000001 SOS TAC ;ADDRESS -1
788 001257 200400 001275' MOVE TEM,IOWD ;GET IOWD
789 001260 202400 000000 IOG0: MOVEM TEM,DALOC ;SET UP INTERRUPT LOCATION
790 IFN CPBIT, <
791 001261 603000 040000 TLNE IOS,NORUF ;IF DIRECTLY TO USER
792 001262 254000 001266' JRST IOG02 ;DTALC2, PNTR ALREADY SET UP
793 >
794 001263 202040 002145' MOVEM TAC,PNTR ;POINTER
795 001264 200040 002376' MOVE TAC,[JSR DTATHR] ;SET UP INTERRUPT LOC+1
796 001265 202040 001163' MOVEM TAC,DALC2
797 001266 732202 000000 IOG02: CONO DTC,DTBOTH(TAC1) ;START READ OR WRITE
798 001267 732600 770000 CONO DTS,770000 ;ENABLE FOR ALL INTERRUPTS
799 001270 263140 000000 POPJ PDP, ;DISMISS INTERRUPT
    
```

```

800
801 001271 271040 000176 ;HERE IF TAPE IS GOING IN REVERSE WHEN BLOCK NUMBER FOUND
      IORVRS: ADDI TAC,176 ;START AT TOP OF BUFFER
802 001272 200400 002377 MOVE TEM,[JSR RVERS]
803 001273 254000 001260 JRST IOGO ;COMPLICATED STUFF FOR EACH WORD
804
805
806 ;HERE FOR ANY DATA WORD WITH TAPE IN REVERSE
807 001274 000000 000000 RVERS: 0
808 001275 732000 002145 IOWD: BLKI DTC,PNTR ;READ (WRITE) A WORD
809 ; NOTE -- THIS LOCATION IS IMPURE,
810 ; MODIFIED TO A BLKI OR BLKO,
811 001276 254000 001302 JRST RVTHRU ;POINTER RAN OUT
812 001277 370000 002145 SOS PNTR ;POINTER HAS TO BACK UP
813 001300 370000 002145 SOS PNTR
814 001301 254500 001274 JEN @RVERS ;DISMISS THE INTERRUPT
815
816 ;HERE WHEN POINTER RUNS OUT IN REVERSE
817 001302 732000 770001 RVTHRU: CONO DTS,770001 ;FUNCTION STOP
818 001303 254500 001274 JEN @RVERS ;DISMISS
819
820 ;HERE WHEN POINTER RUNS OUT FORWARD
821 001304 000000 000000 DTATHR: 0
822 001305 732000 770001 CONO DTS,770001 ;FUNCTION STOP
823 001306 254500 001304 JEN @DTATHR ;DISMISS
    
```

```

824                                     ;HERE TO PERFORM A REWIND MTAPE FOR DTA
825 001307 260140 000742' MTAP0: PUSHJ PDP,GETDT      ;GET THE CONTROL FOR THE TAPE
826 001310 242100 000015      LSH      TAC1,15      ;TAC1=0 FOR REW; 10 FOR REW UNLD
827 001311 630000 002400'      TDZ      IOS,[XWD IO,IOACT] ;SET UP IOS
828 001312 661002 101000      TLO      IOS,REWBIT+RVERSE(TAC1)
829 001313 201200 000000      MOVEI    BLK,0      ;SEARCH FOR BLOCK 0
830 001314 260140 001341'      PUSHJ    PDP,FNOBLK   ;GO FIND BLOCK
831
832
833                                     ;CONTROL COMES HERE ON THE INTERRUPT CHANNEL WHEN THE BLOCK IS FOUND
834 001315 603000 200000      TLNE    IOS,RUNBIT   ;REWIND UNLOAD?
835 001316 254000 001321'      JRST    .+3        ;YFS. CONTINUE
836 001317 732600 770001      CONO    DTS,770001   ;NO, THROUGH WITH TAPE. - FNCTN STOP
837 001320 263140 000000      POPJ    PDP,        ;DISMISS INTERRUPT
838 001321 200040 001325'      MOVE    TAC,RUNWD   ;USE THE CLOCK TO DESELECT THE TAPE
839 001322 264000 001334'      JSR     CLKREQ      ;FOR A WHILE - IT WILL GO OFF THE END
840 001323 732200 410000      CONO    DTC,410000   ;DESELECT THE TAPE SO IT CAN GO OFF
841 001324 263140 000000      POPJ    PDP,        ;DISMISS THE INTERRUPT
842 001325 001326' 000200      RUNWD:  XWD      RUNCLK,200

843
844                                     ;COME HERE ON THE CLOCK LEVEL FOR REWIND AND UNLOAD
845 001326 200300 002142'      RUNCLK: MOVE    DEVDAT,USEWRD
846 001327 135040 000746'      LDR     TAC,PUNIT   ;GIVE CONO TO READ BLOCK NOS IN FORWARD
847 001330 242040 000011      LSH    TAC,11      ;DIRECTION - THIS WILL CAUSE FULL REEL
848 001331 732201 230200      CONO    DTC,230200(TAC);TO STOP FLAPPING WHEN CONO STOP IS DONE
849 001332 200006 000002      MOVE    IOS,DEVIOS(DEVDAT) ;RESFT IOS
850 001333 254000 002127'      JRST    REWDUN     ;DECREASE DTREG, STOP TAPE
851
852
853                                     ;HERE TO PUT C(TAC) IN THE CLOCK QUEUE
854 001334 000000 000000      CLKRFQ: 0
855 001335 700600 000000      CONO    PI,PIOFF   ;TURN OFF PI
856 001336 136040 000000      IDPB   TAC,CLOCK
857 001337 700600 000000      CONO    PI,PION
858 001340 254020 001334'      JRST    2CLKREQ
    
```

```

859          EXTERN JOB
860          EXTERN HNGSTP
861          ;COME HERE TO START READING BLOCK NUMBERS
862 001341 552200 002150' FNDBLK: WRRZM  BLK,BLOCK  ;BLOCK WE'RE LOOKING FOR
863 001342 262146 000017          POP      PDP,DISPAD(DEVDAT) ;WHERE TO GO WHEN RIGHT BLOCK FOUND
864 001343 402000 002153'          SETZM  ERRCNT
865 001344 135040 001327' FNDBL2: LDB   TAC,PUNIT  ;GET UNIT NUMBER
866 001345 242040 000011          LSH    TAC,11    ;POSITION IT
867 001346 732300 002000          CONSZ  DTC,20000 ;TAPE SELECTED?
868 001347 254000 001372'          JRST  FNDBL4   ;YES
869 001350 660040 230000          TR0   TAC,230000 ;NO, SET TO SELECT TAPE FORWARD
870 001351 603000 041000          TLNE  IOS,RVERSE+NOBUF ;REVERSE OR DIRECT TO USER?
871 001352 640040 300000          TR0   TAC,300000 ;YES, SET FOR REVERSE
872 001353 660040 001266' FNDBL3: TR0   TAC,DTROTH  ;ADD PI ASSIGNMENT
873 001354 200400 002401'          MOVE  TEM,[JSR SRCH] ;SET UP PI LOCS FOR SEARCH
874 001355 202400 001260'          MOVEM TEM,DTALOC  ;INTO PI INTERRUPTCELL
875 001356 732201 000200          CONO  DTC,200(TAC) ;START SEARCHING BLOCKS
876 001357 732600 670000          CONO  DTS,670000  ;ENABLE FOR ALL BUT JOB DONE
877 001360 476000 002155'          SETOM IOWRIT    ;SET SWITCH NON - 0 IF USING
878 001361 607000 010000          TLNN  IOS,RWDIR  ;MONITOR BUFFER FOR OUTPUT
879 001362 607000 000020          TLNN  IOS,I0    ;SO REKON WONT GIVE CONTROL (AND
880 001363 402000 002155'          SETZM IOWRIT    ;BUFFER) AWAY
881 001364 621000 400004          TLZ   IOS,REKON+I0FST ;TURN OFF DEAD-REKON BIT
882 001365 202300 002142'          MOVEM DEVDAT,USEWRD
883 001366 260140 001133'          PUSHJ POP,STOIOS  ;SAVE IOS
884          IFN CPRIT, <
885 001367 621000 737777          TLZ   IOS,NOBUFC  ;SET WORD NON-ZERO IF NORUF ON
886 001370 556000 002162'          WLRZM IOS,DIRCTN
887          >
888 001371 263140 000000          POPJ  PDP,      ;AND LEAVE
889
890 001372 732300 200000          FNDBL4: CONSZ  DTC,200000 ;DIRECTION TEST
891 001373 664040 200000          TROA  TAC,200000 ;FORWARD
892 001374 660040 100000          TRO   TAC,100000 ;REVERSE
893 001375 254000 001353'          JRST  FNDBL3   ;START SEARCH
    
```

```

894                                     ;INTERRUPT HERE TO READ A BLOCK NUMBER
895 001376 000000 000000 SRCH: 0
896 001377 202040 002146' MOVEM TAC,TEMP ;SAVE WORKING AC
897 001400 732640 000001 CONI DTS,TAC ;GET STATUS BITS
898 001401 603040 002000 TLNE TAC,2070 ;IS CONTROL STILL ACTIVE?
899 001402 254000 001450' JRST SRCHD ;YES, MUST BE IN "PSEUDO END-ZONE"
900 001403 732040 000001 DATAI DTC,TAC ;NO, READ A BLOCK NUMBER
901 001404 621040 777777 TLZ TAC,-1 ;INSURANCE
902 001405 303040 001101 CAILE TAC,TOPBLK ;LEGAL BLOCK NUMBER?
903 001406 254000 001434' JRST SRCHB ;NO, SET ERROR SWITCH
904 001407 274040 002150' SUB TAC,BLOCK ;NOW-WANTED
905
906 001410 332000 002162' SKIPE DIRCTN ;MUST BE GOING FORWARD IF DUMP-MODE
907 001411 254000 001442' JRST SRCHC ;INDIRECTLY TO USER
908
909 001412 322040 001614' JUMPE TAC,FOUND ;=0 IF WE'RE THERE
910 001413 732300 100000 SRCHA: CONS? DTC,100000 ;IF TAPE IS IN REVERSE
911 001414 213000 000001 MOVNS TAC ;SWITCH TURN-AROUND TEST
912 001415 323040 001420' JUMPLE TAC, +3 ;TEST FOR DISCONNECT IF DIRECTION IS CORRECT

913
914 001416 732200 000000 CONO DTC,DTURN ;***
915 001417 254000 001431' JRST SRCHXT ;TURN AROUND
916 001420 217000 000001 MOVMS TAC ;AND GO AWAY
917 001421 303040 000014 CAILE TAC,MINDIS ;WORTH WHILE TO DISCONNECT TAPE?
918 001422 254000 001452' JRST REKON ;YES, GO DISCONNECT
919 001423 333000 002152' SKIPLD DISCON ;NO, IS THERE A DISCON. TAPE
920 ;WHICH HAS TIMED OUT?
921 001424 307040 000002 CAIG TAC,2 ;YES, WILL THIS SEARCH TAKE LONG?
922 001425 334000 000000 SKIPA ;NO
923 001426 732600 670002 CONO DTS,670002 ;YES, STOP DISCONNECTED TAPE
924 001427 332040 002156' SKIPE TAC,ALMSWT ;READING BLOCK NOS OF AN ALMOST ACTIVE DTA?
925 001430 264000 001334' JSR CLKREQ ;YES, GET ITS JOB BACK INTO MEMORY
926 001431 200040 002146' SRCHXT: MOVE TAC,TEMP ;RESTORE TAC
927 001432 402000 002156' SETZM ALMSWT
928 001433 254520 001376' JEN @SRCH ;AND DISMISS THE INTERRUPT
    
```

```

929                                     ;HERE IF AN ILLEGAL BLOCK WAS READ FROM THE TAPE
930 001434 350040 002153' SRCHR: AOS TAC,ERRCNT ;HUMP ERROR COUNT
931 001435 307040 000000 CAIG TAC,DTTRY ;TRIED ENOUGH?
932 001436 254000 001431' JRST SRCHXT ;NO. READ ANOTHER BLOCK NUMBER
933 001437 476000 002150' SETCM BLOCK ;YES. BLOCK = -1 AS AN ERROR SWITCH
934 001440 732600 770001 CONO DTS,770001 ;FUNCTION STOP
935 001441 254000 001431' JRST SRCHXT ;GO AWAY
936
937
938                                     ;FN OPRT, <
939 001442 732300 100000 SRDHC: CONSZ DTC,100000 ;GOING FORWARD?
940 001443 645040 400000 TLCA TAC,400000 ;NO. SWITCH TURN AROUND TEST, ENSURE FORWARD
941 001444 322040 001614' JUMPE TAC,FOUND ;GO IF FOUND FORWARD
942 001445 331000 000001 SKIPL TAC ;TURN AROUND?
943 001446 732200 001416' CONO DTC,DTURN ;YES
944 001447 254000 001431' JRST SRCHXT ;READ ANOTHER BLOCK NUMBER
945
946                                     >
947
948                                     ;COME HERE IF CONTROL IS STILL ACTIVE AFTER READING A BLOCK NUMBER
949                                     ;THIS MEANS THAT TAPE IS IN THE "PSEUDO END-ZONE" - EXTRA FILLERS
950                                     ;INSERTED FOR THE BENEFIT OF THE PDP-9
951 001450 732200 001446' SRCHD: CONO DTC,DTURN ;TURN TAPE AROUND
952 001451 254000 001431' JRST SRCHXT ;AND EXIT THE INTERRUPT
    
```

```

952 ;COME HERE WHEN A TAPE IS IN A LONG SEARCH
953 ;IF THIS TAPE HAS NOT USED THE MONITOR BUFFER (READING), AND
954 ;IF NO OTHER TAPES ARE DISCONNECTED, THIS ONE WILL BE.
955 ;AND A CLOCK REQUEST WILL BE ENTERED FOR AN ESTIMATED TIME TO BLDCK.
956
957 001452 336000 002155' RECON: SKIPN IOWRIT ;MONITOR BUFFER FULL?
958 001453 335000 002152' SKIPGE DISCON ;ANOTHER TAPE DISCONNECTED?
959 001454 254000 001431' JRST SRCHXT ;YES, FORGET IT
960 001455 202100 002154' MOVEM TAC1,FNDTMP ;SAVE AN AC
961 001456 200100 002147' MOVE TAC1,DISTNC
962 001457 274100 000001' SUP TAC1,TAC ;CHECK SEQUENCE
963 001460 366100 001525' SOJN TAC1,TRYLTR ;JUMP IF NOT SEQUENTIAL BLOCKS
964 001461 250300 002142' EXCH DEVDAT,USEWRD ;NO. DISCONNECT FROM THIS ONE
965 001462 205100 000000' MOVSI TAC1,ALMACT ;SET TAPE AS ALMOST ACTIVE - SWAPPABLE AND
966 ;SHUFFLARLE EVEN THOUGH ACTIVE
967 001463 436106 000002' ORM TAC1,DEVVIO(DEV DAT)
968 001464 135100 000000' LDR TAC1,PJORN ;ASSOCIATED JOB NUMBER
969 001465 542100 002163' WRRM TAC1,MONR2 ;SAVE IN WD 2 OF MONITOR BUFFER
970 001466 506300 002163' HRLM DEVDAT,MONB2 ;SAVE ADDRESS OF DTA

971 001467 732300 200000' CONSZ DTC,200000
972 001470 221040 000062' IMULT TAC,62 ;COMPUTE A TIME TO BLOCK
973 001471 732300 100000' CONSZ DTC,100000 ;BASED ON 50 MSEC/BLOCK FORWARD
974 001472 221040 000050' IMULT TAC,50 ;AND 40 MSEC/BLOCK IN REVERSE
975 001473 231040 000000' MSECT: IDIVI TAC,-, ;DIVIDE BY NO. MILLISECS/TIC
976 001474 303040 003300' CALF TAC,3300 ;IF THE COMPUTED TIME IS TOO HIGH
977 001475 201040 000030' MCVSI TAC,30 ;LOOK AGAIN IN 1 1/2 SECONDS
978 001476 205100 400000' MOVSI TAC1,RECKON ;LIGHT A BIT TO INDICATE THE TAPE IS
979 001477 436106 000002' IORM TAC1,DEVVIO(DEV DAT) ;DEAD RECKONING
980 001500 200100 002154' MOVE TAC1,FNDTMP ;RESTORE TAC1
981 001501 542040 001533' WRRM TAC,TIMREQ ;SET UP A TIME REQUEST
982 001502 135040 001344' LDR TAC,PUNIT ;6 BITS OF INFORMATION
983 001503 336000 000001' SKIPN TAC ;UNIT 0 = 8
984 001504 201040 000010' MOVEI TAC,10
985 001505 242040 000014' LSH TAC,14 ;IS THE UNIT NUMBER
986 001506 270040 001533' ADD TAC,TIMREQ ;REQUEST THE MONITOR TO WAKE
987 001507 264000 001334' JSR CLKREQ ;IN N TICKS
988 001510 200040 002150' MOVE TAC,BLOCK ;SAVE THE BLOCK NO.
989 001511 732340 200000' CONSO DTC,200000 ;AND THE DIRECTION
990 001512 660040 200000' TRO TAC,200000
991 001513 506046 000015' HRLM TAC,IBLK(DEV DAT) ;IN THE DDR
992 001514 732200 410000' CONO DTC,410000 ;DESELECT THE CONTROL
993 ;BIT 400000 IS TO PREVENT ERROR FLAGS FROM COMING IN LATER -
994 ; IT DOES NOT CAUSE THE DESELECTED TAPE TO STOP THOUGH
995 001515 201300 000000' MOVEI DEVDAT,0
996 001516 250300 002142' EXCH DEVDAT,USEWRD ;RESET DEVDAT
997 001517 333000 002152' SKIPLE DISCON ;SWITCHING DISCONNECTED TAPES?
998 001520 254000 001530' JRST XTICK ;YES, COME BACK LATER
999 001521 371000 002744' SOSL DTREQ ;NO. COUNT DOWN DTREQ
1000 001522 476000 000000' SETOM DTAVAL
1001 001523 476000 002152' SETOM DISCON ;ONLY 1 RECKON AT A TIME
1002 001524 254000 001431' JRST SRCHXT ;AND EXIT THE INTERRUPT
    
```

```

1003 001525 202040 002147' TRYLTR: MOVEM TAC,DISTNC
1004 001526 200100 002154' MOVE TAC1,FNDTMP IRESTORE TAC1
1005 001527 254000 001431' JRST SRCHXT
1006
1007 ;HERE WHEN SWITCHING DISCONNECTED TAPES
1008 ;TOO MUCH TO DO ON THE INTERRUPT LEVEL, SO COME BACK ON
1009 ;NEXT CLOCK TICK
1010 001530 200040 001534' NXTICK: MOVE TAC,TIMRQ2 ;COME BACK IN A JIFFY
1011 001531 264000 001334' JSR CLKREQ
1012 001532 254000 001431' JRST SRCHXT
1013
1014 001533 001535' 000000 TIMREQ: XWD BACK,0
1015 001534 001602' 000001 TIMRQ2: XWD BACKA,1
1016
1017 IFN ALMACT,<
1018 EXTERN PRIIN
1019 ;COME HERE ON THE CLOCK LEVEL TO BRING THE JOB ASSOCIATED WITH
1020 ;AN "ALMOST ACTIVE" TAPE BACK INTO CORE
1021 ALMREQ: XWD ,+1,1
1022 MOVE DEVDAT,USEWRD ;DOB LOC
1023 LDR ITEM,PJOBN ;JOB NUMBR
1024 PUSHJ PDP,PRIIN ;TELL SWAPPER TO GET IT
1025 JFCL
1026 MOVSI TAC,ALMACT
1027 ANDCAM TAC,DEVIOS(DEVDAT) ;ZAP BIT SO IT WONT BE SWAPPED OUT AGAIN
1028 POPJ PDP,
1029
1030
1031 ;HERE WHEN SWAPPER DOES GET THE JOB BACK INTO CORE
1032 SWPBAK: MOVE IOS,DEVIOS(DEVDAT)
1033 SETZM MONB2 ;CLEAR JOB NO, FROM MON BUFFER
1034 TLNN IOS,IOFST ;WAS JOB SWAPPED OUT WHEN DATA FINISHED?
1035 POPJ PDP, ;NO. TRANSFER COMPLETED
1036 MOVE TAC,JBSTSTS(ITEM) ;YES, WAS IO ABORTED?
1037 TRNN TAC,ALBORT
1038 JRST FAKINT ;NO. NOW TRANSFER DATA, ADVANCE BUFFERS
1039 JRST THRUTP ;YES. FORGET REST OF OPERATION
1040
1041 >

```

```

1042                                     EXTERN CIPWTM
1043                                     ;COME HERE FROM THE MONITOR WHEN THE ESTIMATED TIME-TO BLOCK
1044                                     ;HAS EXPIRED
1045 001535 332000 002142' BACK: SKIPE USEWRD ;IS CONTROL AVAILABLE?
1046 001536 254000 001572' JRST FLAGIT ;NO. SET FLAG
1047 001537 350000 001521' AOS DTREQ ;YES. DTREQ WILL BE COUNTED DOWN LATER
1048 001540 402000 001522' SETZM DTAVL
1049 001541 402000 002152' SETZM DISCON ;NO TAPE IS NOW DISCONNECTED
1050 001542 620040 000010 BACK2: TRZ TAC,10
1051 001543 242040 000011 LSH TAC,11 ;UNIT
1052 001544 261140 000001 PUSH PDP,TAC ;SAVE IT
1053 001545 242040 000003 LSH TAC,3
1054 001546 270040 000000' ADD TAC,DTADDN ;CONVERT TO SIXBIT NAME
1055 001547 260140 000000' PUSHJ PDP,DEVPHY ;SFT UP DEVDAT FOR IT
1056 001550 254200 001550' HALT . ;*****
1057 001551 202300 002142' MOVEM DEVDAT,USEWRD ;RESET USEWRD
1058 001552 554106 000015 HLRZ TAC1,IBLK(DEVDAT) ;BLOCK NEEDED
1059 001553 137100 002402' DPR TAC1,[POINT 10,BLOCK,35]
1060 001554 550040 001336' HRRZ TAC,CLOCK

1061 001555 505100 000556' HRLI TAC1,CPOPJ ;FIND CLOCK-QUEUE REFERENCE TO BACKC
1062 001556 306040 000000 BACK3: CAIN TAC,CIPWTM
1063 001557 254000 001564' JRST BACKD ;NOT THERE, TAPE MUST HAVE BEEN STOPPED
1064 001560 554401 000000 HLRZ TEM,(TAC) ;ADDRESS OF CLOCK REQUEST
1065 001561 302400 001605' CAIE TEM,BACKC ;THIS THE ONE?
1066 001562 364040 001556' SOJA TAC,BACK3 ;NO. KEEP LOOKING
1067 001563 502101 000000 HLLM TAC1,(TAC) ;YES. MAKE A NO-OP OUT OF IT
1068 001564 262140 000001 BACKD: POP PDP,TAC ;UNIT
1069 001565 660040 270000 TR0 TAC,270000 ;SET SELECT, DELAY INHIBIT
1070 001566 602100 200000 TRNE TAC1,200000 ;SET DIRECTION
1071 001567 640040 300000 TRC TAC,300000 ;REVERSE
1072 001570 200006 000002 MOVE IOS,DEVIOS(DEVDAT) ;RESTORE IOS
1073 IFN ALMACT, < ;UNTIL REST OF SCHEDR, SWAPPER IS FIXED - IGNORE
1074 MOVE TAC1,ALMREQ ;INDICATE CONNECTING TO AN ALMOST
1075 MOVEM TAC1,ALMSWT ;ACTIVE TAPE, SO JOB WILL BE BROUGHT
1076 ;BACK IN IF TAPE CLOSE TO TARGET BLOCK
1077 >
1078 001571 254000 001353' JRST FNDBL3 ;READ NEXT BLOCK NUMBER
    
```

```

1079                                     ;HERE WHEN TIME IS UP AND ANOTHER TAPE HAS THE CONTROL
1080 001572 202040 002152' FLAGIT: MOVEM TAC,DISCON ;DISCON POSITIVE IF NEED THE CONTROL
1081 001573 200040 001576' MOVE TAC,,+3 ;SET A CLOCK REQUEST TO STOP TAPE
1082 001574 264000 001334' JSR CLKREQ
1083 001575 263140 000000 POPJ PDP, ;RETURN, WILL COME IN ON INTERRUPT LEVEL
1084 001576 001605' 000010 XWD BACKC,10
1085
1086                                     ;HERE WHEN DTASER CAN GIVE UP THE CONTROL
1087 001577 732200 400000 BACKB: CONO DTG,400000 ;STOP CURRENT TAPE
1088 001600 260140 001751' PUSHJ PDP,THRUTA ;GIVE IT UP
1089 001601 634040 000001 TDZA TAC,TAC ;DISCON WILL GO TO 0
1090
1091 001602 476000 000001 BACKA: SETOM TAC ;DISCON WILL GO TO -1
1092 001603 250040 002152' EXCH TAC,DISCON ;UNIT TO CONNECT TO
1093 001604 254000 001542' JRST BACK? ;GO RECONNECT
1094
1095
1096                                     ;IF WE GET HERE THERE IS A SLOW TAPE ON THE SELECTED DRIVE
1097 001605 700600 001335' BACKC: CONO PI,PIOFF ;CANT RISK CHANGING THE INTERRUPT ENABLES HERE
1098 001606 732640 000001 CONI OTS,TAC ;READ IN ENABLE FLAGS
1099 001607 557000 000001 HLRZS TAC ;FLAGS INTO R,H.
1100 001610 405040 770000 ANDI TAC,770000 ;ONLY ENABLE FLAGS
1101 001611 732601 000002 CONO OTS,2(TAC) ;STOP THE DESELECTED TAPE
1102 001612 700600 001337' CONO PI,PION
1103 001613 263140 000000 POPJ PDP, ;BEFORE RUNNING OFF END OF THE REEL

```

```

1174
1175 001614 202100 002154' ;HERE WHEN CORRECT BLOCK NUMBER IS FOUND
1176 001615 202400 001274' FOUND: MOVEM TAC1,FNDTMP ;SAVE THOSE ACS THAT WILL BE USED
1177 001616 250300 002142' MOVEM TEM,RVERS
1178 001617 250006 000002' FXCH DEVDAT,USEWRD
1179 001620 250140 001631' FXCH IOS,DEVIOS(DEVDAT)
1180 001621 260166 000017' FXCH PDP,FNDPDP
1181 001622 250140 001631' PUSHJ PDP,@DISPAD(DEVDAT) ;GO TO DISPATCH LOCATION
1182 001623 250006 000002' EXCH PDP,FNDPDP ;RESTORE ACS
1183 001624 250300 002142' FXCH IOS,DEVIOS(DEVDAT)
1184 001625 200040 002146' FXCH DEVDAT,USEWRD
1185 001626 200100 002154' MOVE TAC,TEMP
1186 001627 200400 001274' MOVE TAC1,FNDTMP
1187 001630 254520 001376' MOVE TEM,RVERS
1188 JEN @SRCH ;EXIT THE INTERRUPT
1189 001631 777776 001631' FNDPDP: XWD -2,.
1190 001632 000000 000000
1191
1192
1193
1194 ;IFN CPBIT, <
1195 ;HERE WHEN DUMP-MODE POINTER RUNS OUT
1196 001633 000000 000000' DMPADV: 0
1197 001634 202040 002146' MOVEM TAC,TEMP
1198 001635 354040 002161' AOSA TAC,USPNTR ;ADVANCE LOC OF POINTER
1199 001636 542040 002161' DMPAV1: HRRM TAC,USPNTR
1200 001637 336060 000001' SKIPN TAC,@TAC ;END OF LIST?
1201 001640 254000 001646' JRST DMPAV3 ;YES. STOP TAPE
1202 001641 270040 002144' ADD TAC,ADRPRG ;ADD RELOCATION
1203 001642 327040 001636' JUMPG TAC,DMPAV1
1204 001643 202040 002145' MOVEM TAC,PNTR ;NEW POINTER
1205 001644 200040 002146' DMPAV2: MOVE TAC,TEMP ;RESTORE TAC
1206 001645 254520 001633' JEN @DMPADV
1207
1208
1209
1210
1211
1212
1213 001646 732600 770001' DMPAV3: CONO DTS,770001 ;GIVE FUNCTION STOP
1214 001647 254000 001644' JRST DMPAV2 ;RESTORE TAC AND EXIT THE INTERRUPT
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
    
```

```

1142 ;INTERRUPT HERE FOR FLAG CHANNEL
1143 001650 732740 770000 DTAIN1: CONSO RTS,770000 ;INTERRUPT FOR DTA?
1144 001651 254000 001651' JRST . ;NO
1145 001652 732700 020000 CONSZ RTS,20000 ;END ZONE?
1146 001653 335000 002150' SKIPGE BLOCK ;YES, BAD BLOCK NUMBER?
1147 001654 334000 000000 SKIPA ;YES
1148 001655 254000 002005' JRST TURN ;NO, TURN TAPE AROUND
1149 001656 264000 000000 JSR RTASAV ;SAVE ACS
1150 001657 200300 002142' FAKINT: MOVE DEVDAT,USEWRD ;RESTORE DEVDAT
1151 001660 200006 000002 MOVF IOS,DEVIOS(DEVDAT) ;AND IOS
1152 001661 135200 001464' LDR ITEM,PJORN ;JOB NUMBER
1153 IFN ALMACT,<
1154 PUSHJ PDP,PRIIM ;IS JOB SWAPPED OR SHUFFLED?
1155 JRST SWPDLY ;YES, CANT MOVE BUFFERS NOW
1156 >
1157 001662 200344 000000 MOVE PROG,JBTADR(ITEM) ;ADDRESS
1158 001663 623000 100000 DTAIN1: TLZE IOS,REWBIT ;NO, FROM A REWIND MTAPE?
1159 001664 254000 002127' JRST REWDUN ;YES, DONE
1160 001665 623000 000004 TLZE IOS,IOFST ;COMING FROM SWAP DELAY ROUTINE?
1161 001666 254000 001674' JRST DTAIN2 ;YES, TD10 HAS BEEN ZAPPED
1162 001667 732700 100000 CONSZ RTS,100000 ;JOB DONE?
1163 001670 732700 670000 CONSZ RTS,670000 ;AND NO ERRORS?
1164 001671 254000 002077' JRST FRRS ;NO, TOUGH LUCK
1165 001672 335000 002150' SKIPGE BLOCK ;BAD BLOCK NUMBER ON TAPE?
1166 001673 254000 002120' JRST DERR ;YES, TURN ON IODERR
1167 001674 603000 004000 DTAIN2: TLNE IOS,DMPMOD ;DUMP MODE?
1168 001675 254000 002050' JRST DMPTHR ;YES, GO ELSEWHERE
1169 001676 607000 002000 TLNN IOS,SINGL
1170 001677 254000 001702' JRST ,+3
1171 001700 135040 002372' LDR TAC,CPOINT 10,BUF,27] ;GET 1ST BLOCK NO. IF READ
1172 ;CAME FROM LOOKUP
1173 001701 202046 000015 MOVEM TAC,IBLK(DEVDAT) ;STORE IN DDB
1174 001702 623000 000001 TLZE IOS,IOW ;NO, IN IO WAIT?
1175 001703 260140 000000 PUSHJ PDP,SETIOD ;YES, TAKE OUT OF WAIT
1176 001704 623000 032000 TLZE IOS,SINGL+RWDIR+DMPCLS ;DIRECTORY OPERATION OR
1177 ;CLOSING DUMP FILE?
1178 001705 254000 001744' JRST THRUTP ;YES, LEAVE
1179 001706 603000 000020 TLNE IOS,IO ;WRITING?
1180 001707 254000 001772' JRST OUTHRU ;YES
    
```

```

1181                                     EXTERN  JBTSTS,PJOBN
1182
1183
1184                                     ;HERE ON END OF AN INPUT BLOCK
1185 001710 561100 000177 HRROI  TAC1,177      ;MASK OUT 1ST-BLK DATA
1186 001711 606000 000100 TRNN  IOS,UOSD      ;UNLESS IN NON-STD
1187 001712 406100 002164' ANDM  TAC1,BUF
1188 001713 201066 000007 MOVEI TAC,@DEVIAD(DEV DAT) ;WHERE TO STORE BLOCK
1189 001714 270040 002403' ADD  TAC,[XWD BUF,1];FROM BUF TO THERE
1190 001715 201101 000177 MOVEI TAC1,177(TAC)
1191 001716 251042 000000 BLT  TAC,(TAC1)      ;TRANSFER IT
1192 001717 554200 002164' HLRZ  BLK,BUF      ;NEXT BLOCK TO READ
1193 001720 602000 000100 TRNE  IOS,UOSD      ;IF NON-STD
1194 001721 354206 000015 AQSA  BLK,IBLK(DEV DAT);RFAD SEQUENTIAL BLOCKS
1195 001722 542206 000015 HRRM  BLK,IBLK(DEV DAT) ;SAVE IN DDR
1196 001723 602000 700000 TRNE  IOS,IOTER+IODERR+IOIMPM
1197 001724 254000 001737' JRST  THRUIN
1198 001725 260140 001011' PUSHJ PDP,ADVBF  ;GET NEXT BUFFER
1199 001726 254000 001737' JRST  THRUIN      ;EMPTY BUF NOT AVAILABLE

1200 001727 333000 002152' SKIPL  DISCON      ;TAPE TIMED OUT?
1201 001730 254000 001577' JRST  RACKB      ;YES, GO RECONNECT
1202 001731 332000 000004 SKIPE BLK          ;EXIT IF EOF OR BLOCK TOO LARGE
1203 001732 303200 001101 CATLF BLK,TOPLBK   ;THE ERROR WILL BE CAUGHT ON THE
1204 001733 254000 001737' JRST  THRUIN      ;UO LEVEL NEXT TIME AROUND
1205 001734 375000 002151' SOSGE QUANTM     ;HAS TAPE HAD CHAN LONG ENOUGH?
1206 001735 337000 001537' SKIPG  DTREQ      ;YES, ANYONE ELSE WANT IT?
1207 001736 254000 001245' JRST  READBC      ;NO, READ NEXT BLOCK
1208
1209
1210 001737 550046 000016 THRUIN; HRRZ  TAC,0BLK(DEV DAT) ;TAPE ALSO BEING WRITTEN?
1211 001740 326040 001744' JUMPN  TAC,THRUTP   ;YES, DONT CHANGE REVERSE BIT
1212 001741 621000 001000 TLZ  IOS,RVERSE   ;NO, SET IOS BIT TO CORRECT DIRECTION
1213 001742 732300 100000 CONSZ DTG,100000
1214 001743 661000 001000 TLO  IOS,RVERSE
    
```

```

1215 ;HERE WHEN TAPE IS DONE
1216 001744 333040 002152' THRUPT: SKIPLE TAC,DISCON ;TAPE TIMED OUT?
1217 001745 254070 001577' JRST RACKR ;YES
1218 001746 732270 400000' CONG CFC,400000 ;STOP TAPE
1219 001747 371070 001735' THRUPT: SOSL DTREQ ;BUMP COUNT DOWN
1220 001750 476000 001540' SETOM DTAVAL ;TELL SCHEDULER
1221 001751 732270 010000' THRUPT: CONO CFC,100000 ;DESELECT CONTROL
1222 001752 621000 364000' TLZ IOS,DMPMOD+N0BUF+DMPCLS+PEWSIT+RUNBIT
1223 001753 402000 002142' SETZM USEWRD ;INDICATE CONTROL NOW FREE
1224 001754 254000 002000' JRST CLRACT ;RESET IOACT AND RETURN
1225
1226 ;COME HERE IF A TAPE IS HUNG
1227 001755 325000 001744' HUNGTP: JUMPG IOS,THRUPT ;GIVE UP CONTROL IF NOT SEAD-RECKONING
1228 001756 254000 001234' JRST CPOPJ1 ;IGNORE IT IF DEAD RECKONING -
1229 ;WHEN THE TAPE TIMES OUT FNOBLK WILL RESET THE HUNG TIME, AND
1230 ;IF IT IS STILL HUNG AT ITS END THE ERROR MESSAGE WILL OCCUR
1231
1232
1233 INTERN FTSWAP
1234 EXTERN SHFWAT
1235 ;HERE ON END OF BLOCK WHEN THE JOB IS STILL SWAPPED OUT
1236 ;CONTROL GETS HERE IF SWP OR SHF IS ON IN JBTSTS
1237 ;IF THE JOB IS CURRENTLY SWAPPED OR SHUFFLED WE MUST DELAY
1238 ;SO TEST JOB NO AGAINST SHFWAT (IF SHUFFLE) OR FORCE (IF SWAP) TO DETERMINE
1239 ;IF JOB IS REALLY IN THAT STATE, OR IF MONITOR HAS LIT THE BIT
1240 ;IN ANTICIPATION OF PUTTING THE JOB IN THAT STATE
1241 001757 SWPDLY:
1242
1243 IFN FTSWAP, <
1244 EXTERN FORCE
1245 001757 603100 002000' TLNE TAC1,SWP ;SWAPPED JOB?
1246 001760 334040 000000' SKIPA TAC,FORCE ;YES,
1247
1248 >
1248 001761 200040 000000' MOVE TAC,SHFWAT ;NO, SHUFFLED?
1249 001762 316200 000001' CAMN ITEM,TAC ;JOB REALLY SWAPPED OR SHUFFLED?
1250 001763 254070 001663' JRST DTAIN1 ;NO, FINISH THIS DATA OPERATION
1251 001764 732270 400000' CONO CFC,400000 ;YES, STOP TAPE (WE MUST DELAY)
1252 001765 661070 000004' TLO IOS,IOFST ;INDICATE JOB WAS DELAYED (FOR FAKINT)
1253 001766 200240 002424' MOVE TAC,CXWD FAKINT,1) ;IF JOB IS CURRENTLY BEING SHUFFLED
1254 001767 607040 002000' TLNA TAC,SWP ;COME BACK ON NEXT CLOCK INTERRUPT, AT WHICH TIME
1255 001770 264070 001334' JSR CLKREQ ;THE BLT MUST HAVE FINISHED
1256 001771 254000 001366' JRST STOPS ;AND GO AWAY
    
```

```

1257                                     ;HERE ON END OF OUTPUT BLOCK
1258 001772 260140 001127' OUTHRU: PUSHJ PDP,ADVBFE ;GET NEXT BUFFER
1259 001773 254000 001744' JRST THRUTP ;NOT FULL
1260 001774 333000 002152' SKIPL DISCON
1261 001775 254000 001577' JRST BACKR
1262 001776 550206 000016 WRRZ BLK,0BLK(DEV DAT) ;NEXT BLOCK TO WRITE
1263 001777 303200 001101 CAILE BLK,TOPBLK ;LEGAL?
1264 002000 254000 001744' JRST THRUTP ;NO. CATCH ERROR ON UO0 LEVEL
1265 002001 375000 002151' SOSGF QUANTM ;YES. HAD CHAN LONG ENOUGH?
1266 002002 337000 001747' SKIPG DTRE0 ;AND SOMEONE ELSE WANT IT?
1267 002003 254000 001056' JRST FILBUF ;NO. GO WRITE NEXT BLOCK
1268 002004 254000 001744' JRST THRUTP ;YES. GIVE UP TAPE
1269
1270
1271                                     ;TURN TAPE AROUND AFTER END-ZONE INTERRUPT
1272 002005 732300 000500' TURN: CONSZ DTC,500 ;READING BLOCK NUMBERS?
1273 002006 254000 002011' JRST DIREOF ;NO. END ZONE WHILE READING DATA
1274 002007 732200 001450' CONO DTC,DTURN ;YES. TURN AROUND
1275 002010 254520 000000' JEN 0DTACHL ;DISMISS INTERRUPT

1276
1277                                     ;COME HERE ON AN END ZONE INTERRUPT WHILE READING DATA
1278                                     ;THIS CAN ONLY HAPPEN IN MODE 116,117
1279                                     ;LIGHT IODEND (IT IS A PREMATURE EOF) AND LEAVE
1280 002011 264000 001656' DIREOF: JSR DTASAV ;SAVE ACS
1281 002012 200300 002142' MOVE DEV DAT,USEWRD ;RESTORE DEV DAT
1282 002013 200006 000002' MOVE IOS,DEV IOS(DEV DAT) ;AND IOS
1283 002014 254000 002073' JRST DMPEOF ;LIGHT IODEND AND RETURN

```

```

1284                                     ;COME HERE ON END OF DUMP MODE BLOCK
1285 002015 333000 002152' SVDMTH: SKIPLE DISCON ;HAS A TAPE TIMED OUT
1286 002016 732600 000002' CONO DTS,2 ;YES, STOP IT
1287                                     IFN CPBIT, <
1288 002017 603000 040000' TLNE IOS,NOBUF ;DIRECTLY TO USER?
1289 002020 254000 002042' JRST USDMTH ;YES, ALMOST THROUGH
1290
1291 002021 205100 777601' >
1292                                     MOVSI TAC1,-177
1293                                     IFE CPBIT, <
1294                                     TRNE IOS,UDSD
1295                                     SUB TAC1,ONEONE ;SET UP TAC1 WITH COUNT
1296 002022 607000 000020' >
1297 002023 254000 002037' JRST SVDMIN ;INPUT FILE
1298 002024 550200 000016' HRRZ BLK,0BLK(DEV DAT) ;OUTPUT FILE, NEXT BLOCK
1299 002025 322200 002031' JUMPF RLK,DMPH A ;LAST BLOCK
1300                                     IFE CPBIT, <
1301                                     TRNE IOS,UDSD ;IF NON-STD MODE
1302                                     AOSA 0BLK(DEV DAT) ;WRITE CONSECUTIVE BLOCKS
1303 >
1304 002026 307200 001101' CAIG BLK,TOPLK ;NOT LAST, LEGAL BLOCK NUMBER?
1305 002027 263140 000000' POPJ PDP, ;YES, RETURN
1306
1307 002030 660000 040000' TRO IOS,IORKTL ;BLOCK TOO LARGE
1308 002031 262140 000001' DMPH A: POP POP,TAC ;REMOVE THE RETURN ADDRESS FROM
1309                                     ;CALL TO SVDMTH, SINCE
1310                                     ;NO MORE I/O WILL BE DONE
1311 002032 402000 000021' DMPH2: SETZM SVDWRD(DEV DAT) ;ZERO DUMP-MODE STUFF
1312 002033 402000 000020' SETZM DMPH3: DMPH3: TLZE IOS,IOW
1313 002034 623000 000001' DMPH3: TLZE IOS,IOW ;IN IO WAIT?
1314 002035 260140 001703' PUSHJ POP,SETIO ;YES, RESTART JOB
1315 002036 254000 001744' JRST THRUTP
1316
1317                                     ;HERE ON END OF SAVE MODE INPUT BLOCK
1318 002037 554200 002164' SVDMIN: HLRZ RLK,RUF ;NEXT BLOCK NUMBER
1319                                     IFE CPBIT, <
1320                                     TRNE IOS,UDSD ;NON-STANDARD?
1321                                     AOSA 0BLK(DEV DAT) ;YES, READ CONSECUTIVE BLOCKS
1322 >
1323 002040 542200 000015' HRRM BLK,IBLK(DEV DAT) ;SAVE IN ODB
1324 002041 254000 001756' JRST CPOPJ1
1325
1326                                     IFN CPBIT, <
1327                                     ;HERE WHEN THROUGH DUMP-MODE DIRECTLY TO USER
1328 SVDMTH: MOVEI TAC1,IBLK(DEV DAT)
1329 002042 201100 000015' MOVEI TAC1,IBLK(DEV DAT)
1330 002043 603000 000020' TLNE IOS,IO
1331 002044 201100 000016' MOVEI TAC1,0RLK(DEV DAT) ;SET TAC1 TO RIGHT BLOCK NUMBER
1332 002045 200040 002157' MOVE TAC,BLKCNT ;UPDATE BLOCK COUNTER
1333 002046 272042 000000' ADDM TAC,(TAC1)
1334 002047 254000 002031' JRST DMPH A ;THROUGH
1335 >
    
```

```

1336                                     ;HERE WHEN THROUGH DUMP MODE BLOCK
1337 002050 260140 002015' DMPTH2: PUSHJ PDP,SVDMTH ;END OF BLOCK HOUSEKEEPING
1338                                     ; RETURN ONLY IF MORE I/O
1339                                     ; WILL BE DONE
1340 002051 254000 001240' JRST DMPFLC ;FILL BUFFER FOR NEXT OUTPUT
1341
1342                                     ;HERE WHEN THROUGH READING A DUMP-MODE BLOCK
1343 002052 332046 000021' DMIFIL: SKIPE TAC,SVDMTH(DEVDAT) ;PARTIAL COMMAND?
1344 002053 254000 002056' JRST DMIFLB ;YES, CONTINUE
1345 002054 260140 001224' DMIFLA: PUSHJ PDP,NXTCOM ;NO, GET NEXT COMMAND
1346 002055 254000 002032' JRST DMPTH2 ;END OF LIST - THROUGH
1347 002056 200402 002165' DMIFLB: MOVE TEM,BUF+1(TAC1) ;NEXT DATA WORD
1348 002057 202401 000000' MOVEM TEM,(TAC) ;GIVE TO USER
1349 002060 252100 002063' AOBJP TAC1,.*3 ;IF BUFFER IS FULL
1350 002061 253040 002056' AOBJN TAC,DMIFLB ;GFT NEXT WORD
1351 002062 254000 002054' JRST DMIFLA ;GFT NEXT COMMAND
1352
1353 002063 253040 002066' AOBJN TAC,.*3 ;BUFFER ID FULL, IS COUNT EXACTLY 177?
1354 002064 260140 001224' PUSHJ PDP,NXTCOM ;THAT COM, IS DONE, GET NEXT
1355
1355 002065 254000 002032' JRST DMPTH2 ;END OF LIST - THROUGH
1356 002066 202046 000021' MOVEM TAC,SVDMTH(DEVDAT) ;SAVE PARTIAL COMMAND FOR NEXT TIME
1357 002067 322200 002073' JUMPE BLK,DMPEOF ;IF EOF - LIGHT BIT
1358 002070 307200 001101' RDNXT: CAIG BLK,TOPBLK ;BLOCK LEGAL?
1359 002071 254000 001245' JRST READBC ;GO READ BLOCK NUMBER
1360 002072 664000 040000' TROA IOS,IOBKTL ;LIGHT ERROR BIT
1361
1362
1363                                     ;EOF BEFORE ALL DATA IS IN - DUMP MODE
1364 002073 660000 002000' DMPEOF: TRO IOS,IOEND ;LIGHT EOF BIT
1365 002074 254000 002032' JRST DMPTH2 ;GIVE UP TAPE
  
```

1366						
1367						
1368	002075	267140	002032	SVADPR: PUSHJ	PDP,DMPTH2	:GIVE UP CONTROL
1369	002076	254070	002000	JRST	ADRERR	:TYPE ERROR MESSAGE
1370						
1371	002077	350040	002153	COME HERE ON ERROR		
1372				ERRS: AOS	TAC,ERRCNT	:BUMP COUNT
1373	002100	607000	040000	IFN CPBIT, <		
1374	002101	254000	002107	TLNN	IOS,NOBUF	:I/O DIRECT TO USER?
1375	002102	200100	002160	JRST	ERRS1	:NO
1376	002103	202100	002161	MOVE	TAC1,SVPNTR	:YES, RESET POINTERS
1377	002104	200102	000000	MOVE	TAC1,USPNTR	
1378	002105	270100	002144	MOVE	TAC1,(TAC1)	:RESET PNTR
1379	002106	202100	002145	ADD	TAC1,ADRPRG	
1380				MOVE	TAC1,PNTR	
1381	002107	732740	040000	>		
1382	002110	303040	001435	ERRS1: CONSO	DTS,40000	:IF ILLEGAL OP - DONT RETRY
1383	002111	254000	002113	CALL	TAC,DITRY	:ENOUGH REREADS?
1384	002112	254000	001344	JRST	PERMER	:YES, PERMANENT ERROR
1385				JRST	FNOBL2	:NO. TRY AGAIN
1386						
1387	002113	331000	002150	:PERMANENT ERROR		
1388	002114	732700	400000	PERMER: SKIPL	BLOCK	:IF BAD BLOCK # LIGHT IODTER
1389	002115	660000	100000	CONSZ	DTS,400000	
1390	002116	732740	010000	TRO	IOS,IODTER	:PARITY
1391	002117	732700	200000	CONSO	DTS,10000	
1392	002120	660000	200000	CONSZ	DTS,200000	
1393	002121	732700	040000	TRO	IOS,IODERR	:MISSED DATA
1394	002122	660000	400000	CONSZ	DTS,40000	
1395	002123	603000	004000	TRO	IOS,IOIMPM	:ILLEGAL OP
1396	002124	254000	002050	TLNE	IOS,DMPMOD	:DUMP MODE?
1397	002125	607000	012020	JRST	DMPTH2	:YES, NOT THROUGH YET
1398	002126	254000	001674	TLNN	IOS,IO+RWDIR+SINGL	
1399	002127	623000	000021	JRST	DTAIN2	
1400	002130	260140	002035	REWDUN: TLZE	IOS,IO*	
1401	002131	627000	012000	PUSHJ	PDP,SETIAD	:NO. TAKE OUT OF IO WAIT
1402	002132	254000	001744	TLZN	IOS,RWDIR+SINGL	:DIRECTORY OPERATION?
1403	002133	260140	001744	JRST	THRUTP	:NO. RETURN TO USER
1404	002134	205040	400000	PUSHJ	PDP,THRUTP	:YES, STOP TAPE
1405	002135	412046	000000	MOVSI	TAC,RVDIRIN	:CLEAR DIRECTORY IN CORE BIT
1406	002136	135200	001661	ANDCAM	TAC,REVMOD(DFVDT)	
1407	002137	254000	000000	RDDIR: LDR	ITEM,PJORN	:NUMBER OF OFFENDING JOB
				JRST	RDDIR	:GO PRINT ERROR MESSAGE

```

1408
1409 002140 777670 002164' BFPNTR: TOWD INTERN DTABUF
1410 002141 000001 000001 ONEONE: XWD 200, RUF+1
1411 002142 000000 000000 USFWRD: 1,1
1412 002143 000000 000000 USFPRG: 0
1413 002144 000000 000000 ADRPRG: 0
1414 002145 000000 000000 PNTR: 0
1415 002146 000000 000000 TEMP: 0
1416 002147 000000 000000 DISTNC: 0
1417 002150 000000 000000 BLOCK: 0
1418 002151 000000 000000 QUANTM: 0
1419 002152 000000 000000 DISCON: 0
1420 002153 000000 000000 ERRCNT: 0
1421 002154 000000 000000 FNDTMP: 0
1422 002155 000000 000000 TOWRIT: 0
1423 002156 000000 000000 ALMSWT: 0
1424
1425 002157 000000 000000 BLKCNT: 0 IFN CPRIT, <
1426 002160 000000 000000 SVPNTR: 0

1427 002161 000000 000000 USPNTN: 0
1428 002162 000000 000000 DIRCTN: 0
1429
1430 >
1431 002163 !THIS IS THE MONITOR BUFFER
1432 DTABUF:
1433 IFN ALMACT, <
1434 XWD SWPBAK, 0 !RM WILL BE LINK TO NEXT MONITOR BUFFER
1435 002163 000000 000000
1436 002164
1437 002364
1438 002364 777777 777601
1439 002365 777752 000123
1440 002366 050000 000000
1441 002367 000026 000026
1442 002370 000040 020000
1443 002371 000001 002164'
1444 002372 101200 002164'
1445 002373 732100 000700
1446 002374 264000 001633'
1447 002375 732000 000300
1448 002376 264000 001324'
1449 002377 264000 001274'
1450 002400 000020 010000
1451 002401 264000 001376'
1452 002402 001200 002150'
1453 002403 002164' 000001
1454 002404 001657' 000001
    
```

NO ERRORS DETECTED

PROGRAM BREAK IS 002405

AC1	000015	INT	AC2	000016	INT	AC3	000017	INT
ADRERR	002076'	EXT	ADRPRG	002144'		ADVBE	001772'	FXT
ADVRF	001725'	EXT	ALMACT	000000		ALMSWT	002156'	
BACK	001535'		BACK2	001542'		BACK3	001556'	
BACKA	001672'		BACKR	001577'		BACKC	001605'	
BACKD	001564'		BADDIR	002137'	EXT	BDDIR	002136'	
RFPNTR	002140'		BLK	000074		BLKCHK	001014'	
BLKCNT	002157'		BLKSRA	000427'		BLKSRB	000432'	
BLKSPC	000424'		RLOCK	002150'		RUF	002164'	
CALNXT	000632'		CHNGDR	000400		CIPWTM	001556'	FXT
CLKREQ	001334'		CLOCK	001554'	EXT	CLRACT	001754'	FXT
CLSDMP	000705'		COMCHK	001136'	EXT	COR2HM	001115'	
CPRT	777777		CPOPJ	001555'	EXT	CPOPJ1	002041'	FXT
DAT	000005	INT	DECPTR	000464'		DERR	002120'	
DEVDAT	000006	INT	DEVEXT	000012	INT	DEVFIL	000011	INT
DEVIAD	000007	INT	DEVIOS	000002	INT	DEVMOD	000004	INT
DEVQAD	000010	INT	DEVPHY	001547'	FXT	DIRBLK	000144	
DIRCHK	000410'		DIRCTN	002162'		DIREOF	002011'	
DISCON	002152'		DISPAD	000017		DISTNC	002147'	
DLTF	000473'		DLCC	000014		DMIFIL	002052'	
DMIFLA	002054'		DMIFLB	002056'		DMPADV	001633'	
DMPAV1	001636'		DMPAV2	001644'		DMPAV3	001646'	
DMPCLS	020000		DMPEOF	002073'		DMPFIL	001175'	
DMPFLA	001200'		DMPFLB	001176'		DMPFLC	001240'	
DMPI	000766'		DMPPLT	000020		DMPMOD	004000	
DMPD	001035'		DMPOTH	001205'		DMPOVA	001221'	
DMPQVR	001215'		DMPSET	001134'		DMPST2	001144'	
DMPTH2	002032'		DMPTH3	002034'		DMPTHA	002031'	
DMPTHR	002050'		DMPTS1	001147'		DRPTR	000552'	
DSER1	000367'		DSERCH	000365'		DTABUF	002163'	INT
DTACHL	002010'	EXT	DTADDB	000000'	INT	DTADDS	000222	INT
DTADTR	000022'		DTADSP	000224'	INT	DTAEND	002364'	
DTAIN1	001663'		DTAIN2	001674'		DTAINI	000247'	INT
DTAINT	001650'	INT	DTALC2	001265'	EXT	DTALOC	001355'	FXT
DTASAV	002011'	EXT	DTASRN	000000'	INT	DTATHR	001374'	
DTAVAL	001750'	EXT	DTBOTH	001353'	EXT	OTC	000320	
DTOCHK	001056'		DTREN	002002'	FXT	OTS	000324	
DTTRY	002110'	EXT	DTTURN	002007'	FXT	DTWAIT	000745'	FXT
DVDIRI	400000	INT	ENTR	000501'		ENTR2	000505'	
ENTRA	000515'		ENTRC	000535'		ENTRD	000577'	
FOF	001020'		FRRCNT	002153'		ERRS	002077'	
FRRS1	002107'		FAKADV	001125'		FAKINT	001657'	
FILBUF	001056'		FLAGIT	001572'		FNDBL2	001344'	
FNDBL3	001353'		FNDBL4	001372'		FNDBLK	001341'	
FNDPDP	001631'		FNDTMP	002154'		FORCE	001760'	FXT
FOUND	001614'		FREXIT	000447'		FSTBLK	000013	
FTCCL	777777		FTDISK	777777		FTLOGI	777777	
FTRC10	777777		FTSWAP	777777	INT	GETDT	000742'	
GETDT0	000740'		GETDT1	000744'		GETF	000611'	
HNGSTP	000764'	EXT	HUNGST	000001	INT	HUNGTP	001755'	
IBLK	000015		IO	000020	INT	IOACT	010000	INT
IORCTL	040000	INT	IODEND	020000	INT	IODERR	200000	INT
IODTER	100000	INT	IOEND	000040	INT	IOFST	000004	INT

DTASRN - NEW FORMAT DECTAPE SERVICE FOR TD-10 (PDP-10) MACRO,V36 19:05 4-JUN-69 PAGE 55-1
 SYMBOL TABLE

IOGO	001260'	IOGO2	001266'	IOIMPM	400000	INT
IORVRS	001271'	IOS	000000	IOW	000001	INT
IOWD	001275'	IOWRIT	002155'	ITEM	000004	INT
JBTADR	001662' EXT	JBTSTS	000000	JIFSEC	000252'	FXT
JOB	000000	JOBDDT	001024'	JORPD1	000000	FXT
JORSAV	000000	LOOK	000256'	LOOKA	000265'	
LOOKC	000316'	LOOKD	000310'	LOOKE	000312'	
LSTBLK	001113'	LSTFRE	000451'	LSTSAV	020000	
MARKDR	000545'	MINDIS	000014	MONB2	002163'	
MSFCPT	001473'	MTAP0	001307'	NAMSTR	000123	
NEWENT	000562'	NMFOUN	000401'	NMLOOK	000372'	
NORLKO	001172'	NORLKS	000646'	NOBUF	040000	
NORUFC	737777	NOLINK	000200	NXTCM1	001227'	
NXTCM2	001235'	NXTCOM	001224'	NXTFRE	000442'	
NXTICK	001530'	OBLK	000016	ONEONE	002141'	
OUCCMP	001073'	OUFULL	001064'	OUT	000704'	FXT
OUTBL2	001106'	OUTBLK	001075'	OUTHRU	001772'	
PDP	000003	PERMER	002113'	PIOFF	001605'	FXT
PIOMOD	000674' EXT	PION	001612' EXT	PJOBN	002136'	FXT
PNTR	002145'	PROG	000007	PUNIT	001502'	FXT
QUANT	000003	QUANTL	000732'	QUANTM	002151'	
QUEST	000763'	RDBLUK	001244'	RDNXT	002070'	
RDRWT	001247'	READRC	001245'	READBF	001243'	
RECKON	400000	REKON	001452'	RELEA9	000000	FXT
RENAM	000324'	RENAM1	000341'	RENAM2	000344'	
RENER1	000357'	RENER2	000361'	REWBIT	100000	
REWDUN	002127'	RUNBIT	200000	RUNCLK	001326'	
RUNWD	001325'	RVERS	001274'	RVERSE	001000	
RVTHRU	001302'	RWDIR	010000	SD	000015	INT
SETACT	000762' EXT	SETI	000603'	SETIOD	002130'	FXT
SETO	000605'	SETPTR	000435'	SETWD4	000567'	
SHFWAT	001761' EXT	SINGL	002000	SPACE	000004	
SRCH	001376'	SRCHA	001413'	SRCHR	021434'	
SRCHC	001442'	SRCHD	001450'	SRCHXT	021431'	
STOIOS	001771' EXT	STOTAC	000617' EXT	STOWN4	000302'	
STREQ	000000	SVADER	002075'	SVDMIN	002037'	
SVDMTH	002015'	SVOWRD	000001	SVPNTR	002160'	
SWP	002000	SWPDLY	001757'	TAC	000001	INT
TAC1	000002	TEM	000010	TEMP	002146'	
THRUIN	001737'	THRUTA	001751'	THRU1D	001747'	
THRUTP	001744'	THSDAT	000511' EXT	TIMREQ	001533'	
TIMR02	001534'	TOPBLK	001101	TOUSER	001151'	
TOUSR1	001156'	TPOPJ	000373' FXT	TPOPJ1	000000	FXT
TRYLTR	001525'	TURN	002005'	UADCK1	000366'	FXT
UCLS	000671'	UOSD	000100	UIN	000772'	
UOUT	001042'	UOUT2	001054'	UREL	000712'	
UREL2	000730'	USDMTH	002042'	USEPRG	002143'	
USEWRD	002142'	USLSTA	000644'	USPNTR	002161'	
USRDDT	001033' EXT	USRFRA	000624'	USRFRE	000620'	
USRLST	000641'	UTPCLR	000650'	UUD	000014	INT
VDTASX	000406	WAIT1	000727' EXT	WRIBLK	001110'	
WSYNC	000741' EXT	ZERCOR	001024'			

CODES	6#		
DISARL	6#		
ENABLE	6#		
NOSCHE	6#		
NOSHUF	6#		
QUEUES	6#		
SCHEDU	6#		
SHUFFL	6#		
STARTD	6#		
XP	6#	6	13