

```

1          ;***COPYRIGHT 1969, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.***
2
3
4          ;THIS SUR-PROGRAM ASSEMBLED WITH SYSTEM PARAMETER FILE - S,MAC(V414)
5          XLIST
6          LIST
7          ;THIS SUR-PROGRAM ASSEMBLED WITH CONFIGURATION DEPENDENT FEATURE SWITCHES - FT50SB,MAC(
8          V073)
9          XLIST
10         LIST
11         TITLE DTCSRN - NEW FORMAT DECTAPE SERVICE FOR 551 (PDP-6)
12         SURTTL DTAS51 A,WACHS/TW/RCC 01 JUN 69 V012
13         XP      VDTASR,012      ; GLOBAL VERSION NUMBER FOR LOADER STORAGE MAP.
14
15
16         ENTRY DTCSRN      ;ENTRY POINT FOR SELECTIVE LOAD BY BUILD
17
18 000070    EXTERNAL      TROPJ,TPOPJ1,DTCCHL,DCOUT,DCIN,DCON,DCOFF
19          EXTERNAL      STOIDS,STOTAP,SETACT,CLRACT,OUT,DTCSAV,PIOMOD
20
21          EXTERNAL      DTAVAL,DTREQ,SETI00,THSDAT,PUNIT,GETDCDT
22          EXTERNAL      ADVBFE,ADVBFF,ADPERR,WAIT1,CPOPJ,CPOPJ1,RADDIR
23          EXTERNAL      COMCHK,PJOBN,RELEASE,UADCK1,DTTRY
24          EXTERNAL      JBTAOR,DCLOC,DCLOC1,DTCCHN,DCREG,DCAVAL
25
26          000074    RLK=4
27
28
29          000144    DIRBLK=+0100      ;NUMBER OF BLOCK FOR DIRECTORY
30
31          001101    TOPBLK=1101      ;HIGHEST LEGAL BLOCK NUMBER
32          000123    NAMSTH=+083      ;1ST NAME WORD IN DIRECTORY
33          000005    QUANT=5          ;NUMBER OF BLOCKS CAN READ BEFORE GIVING UP DTC
34          000006    SPACE=6          ;NUMBER OF BLOCKS SEPERATING CONTIGUOUS BLKS OF A FILE
35
36
37          ;DDB MAGIC CELLS
38          000213    FSTBLK=13
39          000214    DLCC=14
40          000215    IBLK=15
41          000216    OBLK=16
42          000217    DISPAD=17
43          000220    DMPLST=20
44          000221    SVOWRD=21

```

```
45
46          ;FLAGS IN RH OF IOS
47          200100 UNSD=100
48
49          ;FLAGS IN LH OF IOS
50          000200 NOLINK=200
51          000400 CHNGDR=400
52          001000 RVERSE=1000
53          002000 SINGL=2000
54          004000 DMPMOD=4000
55          010000 RWDIR=10000
56          020000 DMPCLS=20000
57          040000 NORUF=40000
58          737777 NORUF=737777
59
60          777777 777777 CPRINT=-1
61
62
63
64
```

;JUST READ OR WRITE 1 BLOCK

;I/O DIRECTLY TO USER
;-NORUF

;CONDITIONAL ASSEMBLY PARAMETER FOR I/O DIRECTLY
;TO USER. IF -1 THE I/O IN DUMP MODE WITH
;BIT 29 ON IN INIT WILL GO DIRECTLY TO USER
;WITHOUT DIRECT CONSIDERATION OF BLOCK BOUNDRIES


```

106
107
108 000247 602000 000100 ;LOOKUP A DIRECTORY ENTRY
109 000250 254000 000200 LOOK: TRNE IOS,UUSD ;NON-STANDARD?
110 000251 260140 000356 PUSHJ PDP,DSERCH ;YES, LOOKUP OK
111 000252 263140 000000 POPJ PDP, ;NO, FIND DIRECTORY ENTRY
112 000253 550101 000026 HRRZ TAC1,26(TAC) ;NOT THERE
113 000254 350000 000014 AOS UOO ;GET DATE, NO. OF 1K BLOCKS NEEDED
114 000255 202100 000014 MOVEM TAC1,@UOO ;POINT UOO TO WORD 3
115 000256 574040 000001 LOOKA: HLRE TAC,TAC ;INTO USER'S LOOKUP BLOCK
116 000257 271040 000027 ADDI TAC,27 ;GET INDEX
117 000260 336006 000013 SKIPM FSTBLK(DEVDAT) ;TAPE BEING WRITTEN?
118 000261 506046 000016 HRLM TAC,DBLK(DEVDAT) ;NO, SAVE INDEX FOR POSSIBLE OUTPUT
119 000262 542040 000015 HRRM TAC,AC1 ;SAVE INDEX IN CASE LH(IRLK) IS
120 ;CHANGED BY DEAD-RECKONING
121 000263 201100 000000 MOVEI TAC1,0
122 000264 260140 000415 PUSHJ PDP,RLKSRB ;COUNT NUMBER OF BLOCKS IN FILE
123 000265 344600 000271 AOJA UOO,STOWD4 ;LAST BLOCK DONE
124 000266 350000 000002 AOS TAC1 ;COUNT THE BLOCK
125 000267 370003 000001 SOS 1(PDP) ;ADJUST FOR CORRECT RETURN
126 000270 253140 000403 AOBJN PDP,RLKSRB ;LOOK FOR NEXT BLOCK
127
128 ;TAC1 HAS THE NUMBER OF BLOCKS BELONGING TO THE FILE
129 000271 220100 000277 STOWD4: IMUL TAC1,C-177] ;NUMBER OF WORDS IF ALL BLOCKS FULL
130 000272 516100 000014 HRLZM TAC1,@UOO ;STORE IN DIRECTORY WD 4
131 000273 275600 000202 SURI UOO,2 ;POINT UOO TO DIRECTORY WD 2
132 000274 621000 000000 TLZ IOS,IO ;MAKE SURE IO IS OFF
133 000275 661000 000000 TLO IOS,SINGL ;JUST READ 1 RECORD
134 000276 201200 000144 MOVEI BLK,DIRBLK ;NO, FIND FIRST MENTION OF BLOCK
135 000277 260140 000443 PUSHJ PDP,LSTFRE+1 ;NEAR DIRECTORY
136 000300 326200 000303 JUMPX RLK,+3 ;FOUND IF BLK NOT =0
137 000301 260140 000415 LOOKD: PUSHJ PDP,RLKSRB ;FIND FIRST MENTION IN DIRECTORY
138 000302 254000 001652 RST RDIR ;NOT THERE - ERROR
139 000303 260140 001176 PUSHJ PDP,READRF ;GO READ IT
140 000304 260140 000000 PUSHJ PDP,WAIT1 ;WAIT TILL IT'S IN
141 000305 506646 000015 HRLM AC1,IBLK(DEVDAT) ;SAVE INDEX ON INPUT FILE FOR LATER
142 ;TEST ON ENTER - WONT ALLOW ENTER
143 ;TO BE DONE ON LOOKED-UP FILE
144 000306 550206 000015 HRRZ RLK,IBLK(DEVDAT) ;GET FIRST BLOCK OF FILE
145 000307 336006 000013 LOOKC: SKIPM FSTBLK(DEVDAT) ;IF FILE HAS NOT BEEN ENTERED
146 000310 202206 000013 MOVEM RLK,FSTBLK(DEVDAT) ;SAVE IN DDR
147 000311 542200 000014 HRRM RLK,@UOO ;SAVE IN USER'S AREA
148 000312 500000 000014 HLL TAC,@UOO ;GET USER'S EXTENSION
149 000313 502046 000012 HLLM TAC,DEVEXT(DEVDAT) ;SAVE IN DEVICE DATA BLOCK FOR RENAME AND
150 ; AND SUPERSADING SHARED SEGMENTS
151 000314 254000 000250 JRST CPOPJ1 ;AND TAKE GOOD EXIT
    
```

```

152 000315 261140 000014 RENAM: PUSH PDP,UUO ;SAVE LOC OF NEW NAME
153 000316 201606 000011 MOVEI UUO,DEVFIL(DEVDAT) ;SEARCH FOR OLD NAME
154 000317 260140 000360 PUSHJ PDP,DSER1
155 000320 254000 000350 JRST RENER1 ;NOT FOUND - ERROR
156 000321 262140 000014 POP PDP,UUO ;FOUND, RESTORE UUO
157 000322 332020 000014 SKIPE @UUO ;RENAMING TO ZERO?
158 000323 254000 000335 JRST RENAM2 ;NO. GO DO REAL RENAME
159 000324 402001 000000 SETZM (TAC) ;YES, DELETE NAME IN DIR
160 000325 402001 000026 SETZM 26(TAC) ;DELETE EXTENSION
161 000326 402006 000011 SETZM DEVFIL(DEVDAT) ;ZERO DEVFIL
162 000327 574040 000001 HLRE TAC,TAC ;GET INDEX OF FILE
163 000330 271040 000027 ADDI TAC,27
164 000331 260140 000464 PUSHJ PDP,DELETE ;DELETE ALL BLOCKS OF FILE
165 000332 661000 000400 RENAM1: TLO IOS,CHNGDR ;DIRECTORY HAS CHANGED
166 000333 350003 000000 AOS (PDP) ;SET FOR GOOD RETURN
167 000334 254000 000000 JRST STOIOS ;GO TO USER
168
169 ;COME HERE TO RENAME TO A REAL NEW NAME
170 000335 200240 000001 RENAM2: MOVE DAT,TAC ;SAVE LOC OF NAME IN DIRECTORY
171 000336 260140 000356 PUSHJ PDP,DSERCH ;SEARCH FOR NEW NAME
172 000337 334000 000000 SKIPE ;NOT FOUND - GOOD
173 000340 254000 000352 JRST RENER2 ;NAME ALREADY EXISTS - ERROR
174 000341 200000 000014 MOVE TAC,@UUO ;GET NEW NAME
175 000342 202045 000020 MOVEM TAC,(DAT) ;SAVE IN DIR
176 000343 350000 000014 AOS UUO
177 000344 200000 000014 MOVE TAC,@UUO ;EXTENSION
178 000345 502045 000026 HLLM TAC,26(DAT) ;SAVE IN DIR
179 000346 502046 000012 HLLM TAC,DEVEXT(DEVDAT) ;SAVE INN DDB
180 000347 254000 000332 JRST RENAM1 ;GIVE GOOD RETURN TO USER
181
182 000350 262140 000014 RENER1: POP PDP,UUO
183 000351 634040 000001 TDZA TAC,TAC ;RH E+1 =0
184 000352 201040 000004 RENER2: MOVEI TAC,4 ;RH F+1 =1
185 000353 350000 000014 AOS UUO ;POINT TO 2ND WORD
186 000354 542060 000014 HRRM TAC,@UUO ;SET ERRORR CODE
187 000355 263140 000000 POPJ PDP, ;LAND TAKE ERROR RETURN
    
```

```

198                                     ;SEARCH DIRECTORY FOR A MATCH
199 000356 201654 000003 DSERCH: MOVEI AC1,3(000) ;CHECK VALIDITY OF ADDRESS
200 000357 260140 007000 PUSHJ PDP,UADCK1 ;NEVER RETURN IF ADDR. OUT OF BOUNDS
201 000360 260140 000401 DSER1: PUSHJ PDP,DIRCHK ;ENSURE DIRECTORY IS IN CORE
202 000361 550046 000014 HRRZ TAC,BLOC(DEV DAT) ;LOCATION OF DIRECTORY
203 000362 270040 002100 ADD TAC,EXWD -26,NAMSTRJ ;POINT TO START OF NAMES
204 000363 336120 000014 NMLOOK: SKIPN TAC1,@JUD ;GET NAME
205 000364 254000 000000 JRST TPOPJ ;NULL ARGUMENT - ERROR RETURN
206 000365 202106 000011 MOVEM TAC1,DEVFIL(DEV DAT) ;STORE FOR RENAME AND SUPERSEDING
207                                     ; SHARED SEGMENTS
208 000366 316101 000000 CAMN TAC1,(TAC) ;TEST FOR MATCH
209 000367 344600 000372 AOJA JUD,NMFOUN ;FOUND NAME, CHECK EXTENSION
210 000370 253040 000366 AORJN TAC,.-2 ;TRY NEXT NAME
211 000371 263140 000000 POPJ PDP, ;NOT FOUND
212 000372 510120 000014 NMFOUN: HLLZ TAC1,@JUD ;PICK UP USER'S EXTENSION
213 000373 430101 000026 XOR TAC1,26(TAC) ;TEST AGAINST DIRECTORY EXTENSION
214 000374 607100 777777 TLNN TAC1,-1 ;MATCH?
215 000375 254000 000314 JRST CPOPJ1 ;YES. RETURN
216 000376 252040 000400 AORJP TAC,.-2
217 000377 364600 000363 SOJA JUD,NMLOOK ;NO. TRY NEXT NAME
218 000400 364600 000000 SOJA JUD,CP0PJ ;NAME NOT FOUND
219
220                                     ;CHECK IF DIRECTORY IS IN CORE. IF NOT, READ IT
221 000401 606000 000100 DIRCHK: TRNN IOS,UOSD ;DONT BOTHER IF NON-STANDARD
222 000402 337000 000004 SKIPG DEVMOD(DEV DAT) ;IS IT IN?
223 000403 263140 000000 POPJ PDP, ;YES. RETURN
224 000404 201200 000144 MOVEI RLK,DIRBLK ;BLOCK NUMBER
225 000405 621000 000020 TLZ IOS,IO
226 000406 260140 000712 PUSHJ PDP,GETDT ;GET CONTROL
227 000407 661000 010000 TLQ IOS,RWDIR ;JUST READ 1 BLOCK
228 000410 260140 001200 PUSHJ PDP,READRC ;GO READ IT
229 000411 260140 000304 PUSHJ PDP,WAIT1 ;WAIT TILL IN
230 000412 205100 400000 MOVSI TAC1,DVDIRI ;SET DIRECTORY-IN-CORE BIT
231 000413 436106 000004 ORN TAC1,DEVMOD(DEV DAT)
232 000414 263140 000000 POPJ PDP,
    
```

```

223                                     ;SEARCH DIRECTORY FOR FILE WHOSE INDEX IS IN TAC
224 000415 205240 440500 RLKSRG: MOVSI DAT,440500 ;DAT IS A BLOCK POINTER
225 000416 540246 000014 HRR DAT,DL0C(DEV0AT)
226 000417 201200 000001 MOVEI RLK,1 ;START AT BLOCK 1
227
228 000420 134400 000005 BLKSRA: ILDB TEM,DAT ;INDEX OF NEXT BLOCK
229 000421 316040 000010 CAMN TAC,TEM ;MATCH?
230 000422 254000 000375 JRST CPOPJ1 ;YES, RETURN
231
232 000423 305200 001171 BLKSRB: CAIGE RLK,TOPBLK ;NO, SEARCHED LAST?
233 000424 344200 000420 AOJA RLK,BLKSRA ;NO, TRY NEXT BLOCK
234 000425 263140 000000 POPJ PDP, ;YES, RETURN
235
236                                     ;SET UP POINTER TO DIRECTORY FOR BLOCK IN BLK
237 000426 261140 000004 SETPTR: PUSH PDP,BLK ;SAVE BLK
238 000427 260140 000543 PUSHJ PDP,DRPTR ;SET BLK AS A BYTE POINTER
239 000430 200240 000004 MOVE DAT,RLK ;RETURN IT IN DAT
240 000431 262140 000004 POP PDP,BLK ;RESTORE BLK
241 000432 263140 000000 POPJ PDP, ;AND RETURN

242
243                                     ;GET NEXT AVAILABLE FREE BLOCK
244 000433 261140 000010 NXTFRE: PUSH PDP,TEM
245 000434 260140 000426 PUSHJ PDP,SETPTR ;SET DAT TO A BYTE POINTER
246 000435 201040 000000 MOVEI TAC,0 ;LOOK FOR FREE BLOCKS
247 000436 260140 000420 PUSHJ PDP,BLKSRA ;FIND A ZERO BLOCK
248 000437 201200 000000 MOVEI RLK,0 ;NOT THERE- RETURN 0
249 000440 262140 000010 FREXIT: POP PDP,TEM
250 000441 263140 000000 POPJ PDP,

251
252                                     ;GET PREVIOUS FREE BLOCK
253 000442 201040 000000 LSTFRE: MOVEI TAC,0
254 000443 261140 000010 PUSH PDP,TEM
255 000444 271200 000002 ADDI RLK,2
256 000445 260140 000426 PUSHJ PDP,SETPTR ;SET DAT AS A POINTER
257 000446 275200 000002 SURI RLK,2
258 000447 260140 000455 PUSHJ PDP,DECPTR ;DECREMENT BYTE POINTER
259 000450 135400 000005 LDR TEM,DAT ;INDEX TO BLOCK
260 000451 316400 000001 CAMN TEM,TAC ;FOUND?
261 000452 254000 000440 JRST FREXIT ;YES, RETURN
262 000453 367200 000447 SOJG RLK,-4 ;TRY AGAIN IF NOT AT START
263 000454 254000 000440 JRST FREXIT ;REACHED START - RETURN BLK=0
264
265
266 000455 321240 000462 DECPTR: JUMPL DAT,+5 ;DECREMENT
267 000456 270240 000101 ADD DAT,[BYTE (6) 5] ;IF POSITIVE - SAME WORD
268 000457 327240 000400 JUMPR DAT,CPOPJ ;RESET TO PREVIOUS WORD
269 000460 505240 010500 HRLI DAT,010500
270 000461 364240 000457 SOJA DAT,CPOPJ
271 000462 505240 060500 HRLI DAT,060500
272 000463 364240 000461 SOJA DAT,CPOPJ
    
```

```

273                                     ;COME HERE TO DELETE THE FILE WHOSE INDEX IS INN TAC
274 000464 201100 000000 DLETE: MOVEI TAC1,0 ;SET TO DELETE BLOCKS
275 000465 260140 000415' PUSHJ PDP,BLKSRB ;FIND A BLOCK BELONGING TO FILE
276 000466 254000 000463' JRST CPOPJ ;ALL THROUGH
277 000467 137100 000005 DPB TAC1,DAT ;DELETE IT
278 000470 370003 000001 SOS 1(PDP) ;ADJUST PDL FOR RETURN
279 000471 253140 000423' AORJN PDP,BLKSRB ;AND FIND NEXT MATCH
280
281
282                                     ;ENTER A FILE NAME IN DIRECTORY
283 000472 602000 000100 ENTR: TRNE IOS,UDSD ;NON STANDARD?
284 000473 254000 000422' JRST CPOPJ1 ;YES. RETURN
285 000474 260140 000356' PUSHJ PDP,DSERCH ;NO. LOOK FOR MATCH
286 000475 254000 000553' JRST NEWENT ;THIS IS A NEW ENTRY
287 000476 200120 000014 ENTR2: MOVE TAC1,@UUO ;PICK UP 2ND WORD (EXTENSION)
288 000477 350000 000014 AOS UUO ;POINT TO WORD 3
289 000500 540120 000014 HRR TAC1,@UUO ;ADD DATE
290 000501 606100 007777 TRNN TAC1,7777 ;IS DATE ALREADY THERE?
291 000502 434100 000000 IOR TAC1,THSDAT ;NO. ADD CURRENT DATE
292 000503 202101 000026 MOVEM TAC1,26(TAC) ;INTO DIRECTORY
293 000504 335746 000016 SKIPGE AC3,0BLK(DEVDAT) ;IS THIS A SAVE FILE (UGETF DONE
294 ;BEFORE THE ENTER?)
295 000505 344000 000560' ACJA UUO,SETW04 ;YES. STORE LENGTH IN DIRECTORY
296 000506 275600 000002 ENTR3: SUBI UUO,2 ;NO. POINT TO NAME
297 000507 200120 000014 MOVE TAC1,@UUO ;PICK IT UP
298 000510 202101 000000 MOVEM TAC1,(TAC) ;INTO DIRECTORY
299 000511 574040 000001 HLRE TAC,TAC ;COMPUTE INDEX OF FILE
300 000512 271040 000027 ADDI TAC,27
301 000513 554246 000015 HLRZ DAT,IBLK(DEVDAT) ;INDEX OF INPUT FILE
302 000514 274240 000001 SUB DAT,TAC ;WRITING SAME FILE AS READING?
303 000515 322240 000466' JUMPF DAT,CPOPJ ;TAKE ERROR RETURN IF YES
304 000516 506046 000016 HLRL TAC,0BLK(DEVDAT) ;SAVE INDEX IN DDB
305
306 000517 260140 000464' PUSHJ PDP,DELETE ;DELETE ALL BLOCKS BELONGING TO FILE
307 000520 342740 000570' AOJE AC3,ENTRD ;FIND FIRST FREE BLOCK ON TAPE IF THIS
308 ;IS A SAVE FILE (UGETF DONE)
309 000521 201200 000144 MOVEI BLK,DIRBLK ;NO. GET 1ST BLOCK CLOSE TO
310 000522 661000 001000 TLO IOS,RVERSE ;DIRECTORY, GOING IN REVERSE
311 000523 260140 000636' PUSHJ PDP,USLSTA
312 000524 303200 001101 CAILF BLK,TOPBLK ;BLOCK LEGAL?
313 000525 263140 000000 POPJ PDP, ;NO. ERROR RETURN
314 000526 202206 000013 ENTRC: MOVEM BLK,FSTBLK(DEVDAT) ;SAVE AS 1ST BLOCK
315 000527 542006 000016 HRRM BLK,0BLK(DEVDAT) ;SAVE IN DDB
316 000530 350000 000014 AOS UUO ;POINT UUO TO WORD 2
317 000531 542200 000014 HRRM BLK,@UUO ;SAVE 1ST BLOCK IN USER'S AREA
318 000532 502060 000014 HLL TAC,@UUO ;GET EXTENSION
319 000533 502046 000012 HLLM TAC,DEVEXT(DEVDAT) ;SAVE EXTENSION IN DDB ALSO
320 000534 661000 000200 TLO IOS,NOLINK
321 000535 350003 000000 AOS (PDP)
    
```

```

322                                     ;MARK DIRECTORY ENTRY POINTED TO BY BLK AS TAKEN
323 000536 260140 000543' MARKDR: PUSHJ PDP,DRPTF ;SET POINTER TO BLOCK IN DIR
324 000537 554046 000016 HLRZ TAC,@BLK(DEV0AT) ;PICK UP INDEX
325 000540 136040 000024 IDPB TAC,RLK ;MARK DIRECTORY
326 000541 661020 000400 TLO IOS,CHNGDR ;DIRECTORY HAS CHANGED
327 000542 254020 000334' JRST STOIOS
328
329                                     ;;SET POINTER TO CORRECT DIRFCTORY ENTRY
330 000543 275200 000001 DRPTR: SUBI BLK,1 ;SET FOR ILDB OR IDPB
331 000544 231200 000027 IDIVI BLK,7 ;COMPUTE WORD, POSITION
332 000545 270206 000014 ADD BLK,@LOC(DEV0AT) ;GET CORRECT ADDRESS
333 000546 505200 440500 HRLI BLK,440500 ;MAKE IT A BYTE POINTER
334 000547 322240 000515' JUMPE DAT,CPOPJ ;CORRECT FOR POSITION IN WORD
335 000550 133000 000024 IBP BLK
336 000551 367240 000550' SOJG DAT,-1
337 000552 263140 000000 POPJ PDP,
338
339                                     ;HERE FOR NEW FILE NAME ON ENTER
340 000553 274040 002122' NEWENT: SUB TAC,FXWD 26,263;START AT BEGINNING OF DIRECT.

341 000554 336001 000000 SKIPN (TAC) ;FIND A FREE SLOT
342 000555 344600 000476' AOJA UO,ENTR? ;RETURN WITH UO POINTING TO WRD 2
343 000556 253040 000554' AORJN TAC,-2
344 000557 263140 000000 POPJ PDP, ;NONE AVAILABLE.
345
346
347                                     ;SET UP LENGTH OF FILE IN DIRECTORY FOR A SAVE FILE
348 000560 574120 000014 SETW04: HLRE TAC1,@UO ;GET -LENGTH
349 000561 213000 000002 MOVNS TAC1 ;+LENGTH
350 000562 570420 000014 HRRE TEM,@UO ; +START ADDRESS
351 000563 270100 000010 ADD TAC1,TEM ;STORE N-1. WHERE N IS NO OF K
352 000564 620100 001777 TRZ TAC1,1777
353 000565 242100 000002 LSH TAC1,2
354 000566 436101 000026 ORM TAC1,26(TAC) ;INTO 2ND WRD OF DIRECTORY
355 000567 364600 000506' SOJA UO,ENTRA ;CONTINUE WITH ENTER
356
357 000570 201040 000000 ENTRD: MOVEI TAC,0 ;GET THE 1ST FREE BLOCK ON TAPE
358 000571 260140 000415' PUSHJ PDP,RLKSRC ;AS THE 1ST LOGICAL BLOCK OF THE FILE
359 000572 263140 000020 POPJ PDP, ;NONE AVAILABLE
360 000573 254020 000526' JRST ENTRC ;CONTINUE WITH ENTER
    
```

```

361                                     ;USETI - SET NEXT INPUT BLOCK TO READ
362 000574 630000 002133' SETI: TDZ IOS,EXWD IOEND,IOENDJ
363 000575 334240 000006 SKIPA DAT,DEVDAT

364                                     ;USETO - SET NEXT OUTPUT BLOCK TO READ
365
366 000576 201246 000001 SETO: MOVEI DAT,1(DEVDAT)
367 000577 260140 000411' PUSHJ POP,WAIT1 ;WAIT FOR BUFFERS TO FILL (OR EMPTY)
368 000600 542605 000015 HRRM BUO,IBLK(DAT) ;SET BLOCK NUMBER
369 000601 254000 000542' JRST STOIOS ;STOE IOS, POPJ

370
371                                     ;UGETH - GET NEXT FREE BLOCK FOR THIS FILE
372 000602 260140 000577' GETF: PUSHJ PDP,WAIT1 ;WAIT TILL BUFFERS EMPTY
373 000603 260140 000401' PUSHJ PDP,DIRCHK ;ENSURE DIR, IN CORE
374 000604 260140 000611' PUSHJ PDP,USRFRE ;GET NEXT AVAILABLE BLOCK
375 000605 336006 000016 SKIPN OBLK(DEVDAT) ;IF AN ENTER HAS NOT BEEN DONE,
376 000606 477276 000016 SETOP BLK,OBLK(DEVDAT) ;SET SWITCH SO THAT ENTER WILL GET
377                                     ;1ST FREE BLOCK ON TAPE (FOR 10DMP)
378 000607 200040 000004 MOVE TAC,BLK ;TELL USER THE BLOCK NUMBER
379 000610 254000 000000 JRST STOTAC

380
381
382                                     ;GET NEXT (OR PREVIOUS) FREE BLOCK
383 000611 201440 000006 USRFRE: MOVEI TEM,SPACE ;BLOCKS "SPACE" APART
384 000612 135200 000000 LDR BLK,PIOMOD ;EXCEPT DUMP-MOSE
385 000613 301200 000015 TAIL BLK,SO
386 000614 201400 000002 MOVEI TEM,2
387 000615 201400 000002 MOVEI TEM,2 ;WHICH ARE CLOSER
388 000616 550276 000016 USRFRA: HRRZ BLK,OBLK(DEVDAT) ;CURRENT BLOCK
389 000617 603000 001200 TLNE IOS,RVERSE ;FORWARD?
390 000620 254000 000633' JRST JSRLST ;NO
391 000621 271210 000000 ADDI BLK,(TEM) ;YES, FIND NEXT BLOCK AT LEAST N
392 000622 303200 001101 CAILE BLK,TOPBLK
393 000623 634220 000004 TDZA BLK,BLK
394 000624 260140 000433' CALNXT: PUSHJ PDP,NXTFRE ;BLOCKS PAST THIS ONE
395 000625 326200 000601' JUMPN BLK,STOIOS ;RETURN IF FOUND
396 000626 663400 000001 TLOE TEM,1 ;FOUND NONE ON THIS PASS
397 000627 254000 000640' JRST NOBLKS ;TAPE IS FULL
398 000630 641000 001000 TLC IOS,RVERSE ;REVERSE DIRECTION
399 000631 541400 000001 HRRJ TEM,1 ;START LOCKING AT NEXT BLOCK IN OTHER DIRECTION
400 000632 254000 000616' JRST USRFRA
401 000633 275210 000000 USRLST: SURJ BLK,(TEM) ;LOOK FOR FREE BLOCK N BEFORE
402 000634 337000 000004 SKIPR BLK
403 000635 634270 000004 TDZA BLK,BLK ;REVERSE IF AT FRONT OF TAPE
404 000636 260140 000442' USLSTA: PUSHJ PDP,LSTFRE ;THIS ONE
405 000637 254000 000625' JRST CALNXT+1

406
407
408                                     ;NO FREE BLOCKS AVAILABLE, GIVE HIGH BLOCK,SET IOBKTL LATER
409 000640 201200 001102 NOBLKS: MOVEI BLK,TOPBLK+1 ;SET HIGH BLOCK
410 000641 263140 000000 POPJ PDP,
    
```

411				;UTPCLR UUO	
412	000642	602000	000100	UTPCLR: TRNE	IOS,UDSD
413	000643	263140	000000		PDP,
414	000644	205040	400000	POPJ	;FORGET IT FOR NON-STANDARD
415	000645	436046	000004	MOVSI	TAC,400000 ;SET DIRECTORY-IN-CORE BIT
416	000646	661000	000400	ORM	TAC,DEVMOD(DEV DAT)
417	000647	550046	000014	TLO	IOS,CHNGDR ;DIRECTORY HAS CHANGED
418	000650	504100	000001	HRRZ	TAC,DLOC(DEV DAT) ;LOC OF DIRECTORY
419	000651	541101	000001	HRL	TAC1,TAC
420	000652	402001	000000	HRPI	TAC1,1(TAC) ;BLT POINTER
421	000653	251101	000176	SETZM	(TAC)
422	000654	205100	017000	BLT	TAC1,176(TAC) ;LEAVE LAST WORD IN DIR, ALONE
423	000655	202101	000016	MOVSI	TAC1,17000 ;MARK DIRECTORY AS UNAVAILABLE
424	000656	205100	757000	MOVEM	TAC1,16(TAC)
425	000657	202101	000000	MOVSI	TAC1,757000 ;RESERVE BLOCKS 1 AND 2
426	000660	205100	777770	MOVEM	TAC1,(TAC) ;FOR READ IN MODE LOADER
427	000661	456101	000122	MOVSI	TAC1,777770 ;MARK BLOCKS 1102-1105 AS
428	000662	254000	000625	ORCAM	TAC1,NAMSTR-1(TAC) ;UNAVAILABLE ALSO
				JRST	STOIOS

```

429                                     ;CLOSE UUU
430 000663 623000 000200 UCLS: TLZE IOS,NOLINK ;IF NO LINK TO FREE BLOCK
431 000664 602000 000100 TRNE IOS,UUSD ;OR IF NON-STD MODE
432 000665 254000 000662' JRST STOIOS ;FORGET IT
433 000666 135040 000612' LDR TAC,PIOMOD ;WRITE LAST BLOCK
434 000667 301040 000016 CAIL TAC,16 ;DUMP MODE?
435 000670 254000 000721' JRST CLSDMP ;YES, CLOSE DUMP MODE
436 000671 201066 000010 MOVEI TAC,@DEV0AD(DEV DAT) ;LOC OF PUFFER
437 000672 200101 000001 MOVE TAC1,1(TAC) ;LINK WORD
438 000673 667100 777777 TLOW TAC1,-1 ;LINK=-1 IF NOT SPECIFIED
439 000674 202101 000001 MOVEM TAC1,1(TAC) ;LINK = -1... EOF
440 000675 202006 000002 MOVEM IOS,DEVIOS(DEV DAT) ;SAVE IOS
441 000676 254000 000000 JRST OUT ;GO TO WRITE RECORD
442
443
444                                     ;RELEASE UUU
445 000677 402006 000013 URFL: SETZM FSTBLK(DEV DAT) ;ZERO FSTBLK
446 000700 260140 001170' PUSHJ PDP,NXTCM2 ;CLEAR OUT DUMP-MODE STUFF
447 000701 620000 000100 TRZ IOS,UUSD ;CLEAR NON-STD BIT
448 000702 337006 000004 SKIPL DEVMOD(DEV DAT) ;IF DIRECTORY HAS BEEN
449 000703 627000 000400 TLZN IOS,CHNGDR ;MODIFIED IT MUSY BE WRITTEN
450 000704 263140 000000 POPJ PDP, ;NOT TOUCHED
451 000705 661000 010020 TLO IOS,RWDIR+IO ;GOING TO WRITE IT
452 000706 260140 000712' PUSHJ PDP,GETDT ;WAIT TILL DTC AVAILABLE
453 000707 201200 001144 MOVEI RLK,DIRBLK ;BLOCK NUMBER
454 000710 260140 001045' PUSHJ PDP,WRTBLK ;WRITE UT
455 000711 254000 000602' JRST WAIT1 ;DONT RETURN TO USER TILL DONE
456
457
458                                     ;GET DEC TAPE CONTROLLER
459 000712 260140 000000 GETDT: PUSHJ PDP,GETDCDT ;GET DATA CONTROL, DECTAPE CONTROL
460 000713 352000 000000 AOSE DTREQ ;ARGUMENT FOR GETDCD
461 000714 201040 000005 MOVEI TAC,QUANT ;HAVE IT NOW
462 000715 202040 001664' MOVEM TAC,QUANTM ;KEEP IT FOR QUANT BLOCKS
463 000716 202340 001657' MOVEM PROG,USEPRG
464 000717 202300 001656' MOVEM DEV DAT,USEWRN ;SAVE ACS NEEDED ON INTERRUPT LEVEL
465 000720 254000 000000 JRST SETACT ;LIGHT IOACT AND RETURN
466
467
468                                     ;HERE TO CLOSE A DUMP MODDE FILE
469 000721 661000 000721' CLSDMP: TLO IOS,CLSDMP ;SET SWITCH
470 000722 260140 000712' PUSHJ PDP,GETDT ;GET CONTROL
471 000723 402000 001677' SETZM BUF ;ENSURE LINK, WORDCOUNT=0
472 000724 254000 001022' JRST OUFULL ;GO WRITE LAST RECORD
    
```

```

473
474
475 000725          EXTERN  JOBBOOT,USRDDT
476 000725 550716 000015  ;DUMP MODE INPUT
477          DMPI:  IFN    CPBIT, <
          >      HRRZ   AC2,IBLK(DEVNAT)
478 000726 260140 001070'  PUSHJ  PDP,DMPSET      ;SET UP DUMP-MODE STUFF
479 000727 254000 000762'  JRST   ZERCOR        ;ZERO USER'S CORE IF SAVE-MODE
480
481
482
483 000730 621000 000020  ;INPUT UDD
484 000731 550206 000015  UIN:   TLZ    IOS,IO
485 000732 260140 000665'  HRRZ   RLK,IBLK(DEVNAT) ;BLOCK TO READ
486 000733 602000 000100'  PLSNJ  PDP,STOIOS
487 000734 254000 001176'  TRNE   IOS,UOSD      ;NON STANDAR?
488 000735 322200 000756'  JRST   READRF
489 000736 260140 000752'  JUMPF  RLK,EOF       ;0 MEANS EOF
490 000737 607000 004000'  PUSHJ  PDP,RLKCHK    ;CHECK LEGALITY OF BLOCK NUMBR
491 000740 302200 000144'  TLNN   IOS,DMPMOD    ;DUMP MODE?
          CAIE   RLK,DIRBLK ;TRYING TO READ DIRECTORY?
492 000741 254000 001176'  JRST   READRF       ;NO. GO READ
493
494
495 000742 260140 000401'  ;READING DIRECTORY - GIVE COPE IMAGE IF IT EXISTS
496 000743 504046 000014'  PUSHJ  PDP,DIRCHK    ;READ IT IF IT ISN'T IN ALREADY
497 000744 201126 000007'  HRL    TAC,0LOC(DEVNAT) ;LOC OF DIRECTORY
498 000745 541042 000001'  MOVEI  TAC1,@DEVIAD(DEVNAT) ;WHERE USER WANTS IT
499 000746 251032 000200'  HARI   TAC,1(TAC1)    ;LOC OF DATA
500 000747 260140 000000'  BLT    TAC,200(TAC1)  ;GIVE IT TO HIM
501 000750 255000 000000'  PUSHJ  PDP,ADVBF     ;ADVANCE BUFFERS
502 000751 263140 000000'  JFCL
          POPJ   PDP,
    
```

```

503                                     ;CHECK VALIDITY OF BLOCK NUMBER
504 000752 307200 001101 RLKCHK: CAIG   BLK, TOPBLK   ;LEGAL?
505 000753 263140 002000        POPJ   PDP,         ;YES, RETURN
506 000754 262140 000001        POP   PDP, TAC
507 000755 664000 040000        TROA   IOS, IOBKTL   ;NO, LIGHT ERROR BIT
508
509                                     ;INPUT BLOCK = 0 - END OF FILE
510 000756 661000 000040 EOF:    TLO    IOS, IOEND   ;LIGHT EOF BIT
511 000757 607000 004000        TLNN   IOS, DMPMOD
512 000760 254000 000732'      JRST   ST0IOS
513 000761 254000 001612'      JRST   DMPEOF   ;GIVE UP CONTROL IF DUMP-MODE
514
515
516                                     ;ZERO USER'S CORE ON SAVE-MODE INPUT
517 000762 201047 000000 ZERCOR: MOVEI   TAC, JORDDT(PROG) ;ZERO CORE
518 000763 505041 000001        HRLI   TAC, 1(TAC)
519 000764 207000 000001        MOVSS  TAC          ;BLT POINTER
520 000765 550106 000020        HRZ    TAC1, DMPPLST(DEVDAT) ;TOP CELL TO ZERO (-175)
521 000766 271107 000000        ADDI   TAC1, (PROG)   ;RELOCATE TO USER AREA
522 000767 402001 777777        SETZM -1(TAC)       ;ZERO
523 000770 251042 000000        RLT   TAC, (TAC1)
524 000771 402000 000000        SETZM USRDDT      ;DDT IS KEPT IN PROTECTED PART
525 000772 254000 000730'      JRST   IIN
  
```

```

526                                     ;DUMP MODE OUTPUT
527 000773 260140 000401' DMP0:  PUSHJ  PDP,DIRCHK  ;MAKE SURE DIRECTORY IS IN CORE
528                                     IFN    CPBIT, <
529 000774 550706 000016      HRRZ   AC2,0BLK(DEV DAT)
530                                     >
531 000775 260140 001070'      PUSHJ  PDP,DMPSET  ;SET DUMP-MODE POINTERS
532 000776 255000 000000      JFCL
533
534                                     ;OUTPUT (I/O
535 000777 661000 000020      UOUT:  TLO    IOS,IO
536 001000 602000 000100      TRNE   IOS,UDSD  ;NON STANDARD?
537 001001 254000 001010'      JRST  UOUT2    ;YES
538 001002 260140 000401'      PUSHJ  PDP,DIRCHK  ;NO. MAKE SURE DIRECTORY IS IN CORE
539 001003 550206 000016      HRRZ   BLK,0BLK(DEV DAT)
540 001004 306200 000144      CAIN   BLK,DIRBLK ;CHECK IF WRITING DIRECTORY
541 001005 254000 001052'      JRST  COR2HM   ;YES, WRITE CORE IMAGE
542 001006 322200 001062'      JUMPE  BLK,FAKADV ;DONT WRITE IF NO BLOCK GIVEN
543 001007 260140 000752'      PUSHJ  PDP,RLKCHK ;CHECK FOR LEGAL BLOCK
544 001010 607000 004000      UOUT2: TLNN   IOS,DMPMOD ;ALREADY HAVE CONTROL IF DUMP-MODE

545 001011 260140 000712'      PUSHJ  PDP,GETDT  ;GET DEC TAPE CONTROLLER
546 001012                                     FILBUF:
547 001012 603000 004000      DTOCHK: TLNE   IOS,DMPMOD  ;DUMP MODE?
548 001013 254000 001130'      JRST  DMPFIL   ;YES, FILL BUFFER FROM LIST
549 001014 205066 000010      MOVSI  TAC,@DEV0AD(DEV DAT) ;LOCATION OF BUFFER
550 001015 270040 002104'      ADD    TAC,CXWD 1,BUF];SET TO STORE IN MONITOR BUFFER
551 001016 251000 002076'      BLT   TAC,RUF+177 ;GO BLT IT
552 001017 336000 001677'      SKIPN RUF      ;GIVE UP TAPE IF
553 001020 254000 001445'      JRST  THRUTP  ;NO BUFFER TO OUTPUT
554 001021 621000 000200      TLZ   IOS,VOLINK ;ASSUME A LINK IS GIVEN
    
```

555	001022	602000	000100	OUFULL:	TRNE	IOS,UQSD		;NON-STANDARD?
556	001023	254000	001043'		JRST	OUTBL2		;YES, NO FILE-STRUCTURED OPERATIONS
557	001024	574200	001677'		HLRE	RLK,RUF		;IS IT?
558	001025	321200	001050'		JUMPL	RLK,LSTBLK		;YES, - LAST BLOCK OF FILE
559	001026	326200	001033'		JUMPN	PLK,OUTBLK		;IF NON-0 - YES
560	001027	623000	020000		TLZE	IOS,DMPCLS		;NO. LAST BLOCK OF A DUMP FILE?
561	001030	254000	001033'		JRST	OUTBLK		;YES, LINK MUST STAY 0
562	001031	260140	000611'	OUCOMP:	PUSHJ	PDP,USRFRE		;COMPUTE NEXT BLOCK
563	001032	661000	000200		TLO	IOS,NOLINK		;THIS BLOCK NOT LINKED
564	001033	506200	001677'	OUTBLK:	HRLM	RLK,RUF		;SAVE LINK IN 1ST WORD OF BLOCK
565	001034	200406	000013		MOVE	TEM,FSTBLK(DEVDAT)		;STORE 1ST BLOCK OF FILE IN WORD
566	001035	137400	002105'		DPB	TEM,[POINT 10,RUF,27]		
567	001036	550406	000016		HRRZ	TEM,0BLK(DEVDAT)		;BLOCK TO WRITE NOW
568	001037	542206	000016		HRRM	RLK,0BLK(DEVDAT)		;BLOCK TO WRITE NEXT
569	001040	200200	000010		MOVE	RLK,TEM		
570	001041	260140	000536'		PUSHJ	PDP,MARKOR		;MARK BLOCK TAKEN IN DIRECTORY
571	001042	334200	000010		SKIPA	RLK,TEM		
572	001043	550206	000016	OUTBL2:	HRRZ	RLK,0BLK(DEVDAT)		
573	001044	260140	000760'		PUSHJ	PDP,STOIOS		
574								
575								
576	001045	260140	001250'	WRTBLK:	PUSHJ	PDP,FNOBLK		;GO SEARCH FOR BLOCK
577	001046	200100	002106'		MOVE	TAC1,CRLKO DC,700]		;HERE WE ARE - GO WRITE
578	001047	254000	001202'		JRST	RDWRT		

```

579                                     ;WRITE LAST BLOCK
580 001050 201200 000000 LSTBLK: MOVEI  RLK,0          ;LINK=0
581 001051 254000 001033' JRST  OUTBLK          ;GO WRITE LAST BLOCK
582
583                                     ;TRYING TO WRITE DIRECTORY - STORE IN CORE
584 001052 201066 000010 COR2HM: MOVEI  TAC,0DEVDAD(DEVDAT) ;WHERE IT IS
585 001053 505041 000001 HRLI  TAC,1(TAC)
586 001054 540046 000014 HRR  TAC,DLOC(DEVDAT)          ;WHERE TO PUT IT
587 001055 201101 000177 MOVEI  TAC,177(TAC)
588 001056 251042 000000 RLT  TAC,(TAC)
589 001057 661000 000400 TLO  IOS,CHNGDR          ;REMEMBER TO WRITE IT OUT
590 001062 205100 400000 MOVSI  TAC,400000
591 001061 436106 000024 ORM  TAC,DEVMOD(DEVDAT) ;DIR. IS NOW IN CORE
592 001062 627000 004000 FAKADV: TL7N  IOS,0MPMOD
593 001063 260140 000000 PUSHJ PDP,ADVBFE          ;ADVANCE BUFFERS
594 001064 255000 000000 JFCL
595 001065 621000 000200 TLZ  IOS,NOLINK          ;DIRECTORY BLOCK IS NOT LINKED
596 001066 402006 000016 SETZM OBLK(DEVDAT)
597 001067 254000 001044' JRST  STOIOS
    
```

```

598                                     ;SET UP POINTERS AND STUFF FOR DUMP-MODE
599 001070 661000 004000 DMPSET: TLO   IOS,DMPMOD   ;NO. LIGHT BIT
600 001071 260140 000712' PUSHJ  PDP,GETDT   ;GET CONTROL
601 001072 505600 000007' HRLI  UUU,PROG
602 001073 260140 000000' PUSHJ  PDP,COMCHK   ;CHECK VALIDITY OF LIST
603 001074 254000 001614' JRST  SVADER      ;NG. GIVE ADDRESS ERROR
604 001075 336060 000014' SKIPN TAC,@UUU    ;OK. NULL LIST?
605 001076 254000 001104' JRST  DMPTS1     ;YES. RETURN
606                                     IFN  CPBIT, <
607 001077 602000 000100' TRNE  IOS,UDSD   ;NON-STD DUMP MODE?
608 001100 364240 001106' SOJA  DAT,TOUSER ;YES. GO DO DIRECT I/O
609                                     >
610 001101 370000 000014' DMPST2: SOS     UUU      ;NO. SAVE START OF LIST (-1)
611 001102 202606 000020' MOVEM UUU,DMPLST(DEVDAT)
612 001103 254000 000473' JRST  CPOPJ1
613
614 001104 262140 000001' DMPTS1: POP     PDP,TAC
615 001105 254000 001457' JRST  THRUTD
616
617
618                                     IFN  CPBIT, <
619                                     ;HERE TO START DUMP-MODE INTO USER AREA DIRECTLY
620 001106 322700 001636' TOUSER: JUMPE  AC2,NOBLK0 ;CANT READ BLK 0 IN NON-STD DUMP MODE
621 001107 240240 777771' ASH   DAT,-7      ;NUMBER OF WRDS IN LIST /200
622 001110 350000 000005' AOS   DAT
623 001111 202240 001672' MOVEM DAT,BLKCNT  ;SAVE TO UPDATE POSITION
624 001112 201620 000014' MOVEI UUU,@UUU    ;REAL ADDRESS OF LIST
625 001113 202600 001673' MOVEM UUU,USPNTR ;SAVE IT
626 001114 202600 001674' MOVEM UUU,SVPNTR
627 001115 271047 000000' ADDI  TAC,(PROG)  ;RELOCATE ADDRESS OF 1ST IOWD
628 001116 202040 001660' MOVEM TAC,PNTR   ;AND SAVE IT
629 001117 200040 002107' MOVE  TAC,[JSR DMPADV] ;SET UP LOC FOR WHEN
630 001120 202040 000000' MOVEM TAC,DCLOC1 ;IOWD IS EXHAUSTED
631 001121 552340 001675' HRRZM PRG,ADRPRG ;SAVE JUST ADDRESS OF PRG
632 001122 661000 040000' TLO   IOS,NORUF  ;INDICATE DIRECTLY TO USER
633 001123 256023 000001' XCT   @1(PDP)    ;TURN ON/OFF IO
634 001124 607000 000020' TLNN  IOS,IO
635 001125 254000 001103' JRST  CPOPJ1     ;READING - CONTINUE
636 001126 262140 000001' POP   PDP,TAC    ;WRITING - THIS WILL SAVE LOTS OF TIME
637 001127 254000 001043' JRST  OUTBL2
638                                     >
    
```

```

639                                     ;FILL OUTPUT BUFFER FROM LIST
640 001130 205100 777601 DMPFIL: MOVSI TAC1,-177
641                                     IFE CPBIT, <
642                                     TRNE IOS,UDSD
643                                     SUB TAC1,ONEONE ;200 DATA WORDS IF NON-STANDARD
644                                     >
645 001131 260140 001157' DMPFLB: PUSHJ PDP,NXTCOM ;GET NEXT COMMAND
646 001132 254000 001140' JRST DMPOTH ;END OF LIST
647 001133 200401 000000 DMPFLA: MOVE TEM,(TAC) ;GET NEXT WORD
648 001134 202402 001700' MOVEM TEM,BUF+1(TAC1) ;INTO BUFFER
649 001135 252100 001150' AOBJP TAC1,DMPOVR ;BUFFER FULL IF GOES
650 001136 253040 001133' AOBJN TAC,-3 ;GET NEXT WORD FROM COMMAND
651 001137 254000 001131' JRST DMPFLB ;GET NEXT COMMAND
652
653 001140 DMPOTH:
654                                     IFE CPBIT, <
655                                     TRNN IOS,UDSD
656                                     >
657 001140 552100 001677' HRRZM TAC1,BUF ;LIST RAN OUT SAVE WORD COUNT

658 001141 402002 001700' SETZM BUF+1(TAC1) ;ZERO REST OF BUFFER
659 001142 551102 001701' HRRZI TAC1,BUF+2(TAC1)
660 001143 303100 002076' CAILE TAC1,BUF+177 ;JUST ZERO 1 WORD IF AT TOP
661 001144 254000 001022' JRST OUFULL
662 001145 505102 777777 HRLI TAC1,-1(TAC1)
663 001146 251100 002076' BLT TAC1,BUF+177 ;****TEST IF TOP OF BUFFER
664 001147 254000 001022' JRST OUFULL ;NOW WRITE BUFFER
665
666                                     ;BUFFER FULL BEFORE END OF COMMAND
667 001150 253040 001153' DMPOVR: AOBJN TAC,+.3 ;WAS THAT LAST WORD OF COMMAND?
668 001151 260140 001157' PUSHJ PDP,NXTCOM ;YES. GET NEXT
669 001152 201040 000000 MOVEI TAC,0
670 001153 202046 000021 MOVEM TAC,SVDWRD(DEVDAT) ;NO, SAVE REMAINDER OF COMMAND
671 001154 201040 000177 DMPOVA: MOVEI TAC,177
672                                     IFE CPBIT, <
673                                     TRNN IOS,UDSD
674                                     >
675 001155 202040 001677' MOVEM TAC,BUF ;WD CNT =177
676 001156 254000 001022' JRST OUFULL ;GO WRITE PART OF STUFF
677
678
679                                     ;GET NEXT COMMAND FROM LIST
680 001157 336006 000020 NXTCOM: SKIPN DMPLST(DEVDAT) ;END OF COMMANDS?
681 001160 254000 001170' JRST NXTCM2 ;YES. RETURN
682 001161 354046 000020 AOSA TAC,DMPLST(DEVDAT) ;GET NEXT COMMAND
683 001162 542046 000020 NXTCM1: HRRM TAC,DMPLST(DEVDAT) ;STORE GO-TO ADDRESS
684 001163 200060 000001 MOVE TAC,@TAC ;GET COMMAND
685 001164 322040 001170' JUMPE TAC,NXTCM2 ;END OF LIST
686 001165 327040 001162' JUMPG TAC,NXTCM1 ;GO-TO WORD
687 001166 271047 000000 ADDI TAC,(PROG) ;REAL COMMAND - ADD RELOCATION
688 001167 344040 001125' AUJA TAC,CPOPJ1 ;AND RETURN
689
690                                     ;END OF DUMP-MODE LIST
691 001170 402006 000021 NXTCM2: SETZM SVDWRD(DEVDAT) ;ZERO POINTERS
    
```

692	001171	402006	000020	SETZM	DMPPLST(DEVDAT)	
693	001172	263140	000000	POPJ	PDP,	
694						
695						
696	001173	332046	000021	DMPFLC: SKIPE	TAC,SVOWRD(DEVDAT)	IS THERE ANOTHER COMMAND
697	001174	254000	001133'	JRST	DMPFLA	YES, GET IT
698	001175	254000	001551'	JRST	DMPH2	NO, THROUGH

```

699          000210 DTC=210
700          000214 DTS=214
701
702          ;IO INTERFACE
703 001176 607000 004000 READBF: TLNN IOS,DMPMOD ;HAVE CONTROL IF DUMP-MODE
704 001177 260140 000712' PUSHJ PDP,GETDT ;GET DT CONTROL
705
706 001200 260140 001250' READBC: PUSHJ PDP,FNDBLK ;SEARCH FOR RIGHT BLOCK
707 001201 200100 002110' MOVE TAC1,[BLKI DC,300] ;FOUND IT - START READING
708
709          ;HERE WITH BLK=BLOCK NUMBER, TAC1=FUNCTION. START SEARCH
710 001202 261140 000001 RDWRT: PUSH PDP,TAC
711 001203 502100 001227' HLLM TAC1,IOWD ;BLKI OR BLKO
712 001204 505040 777600 HRLI TAC,-200
713 001205 607000 010000 TLNN IOS,RWDIR ;WRITING (READING) DIRECT?
714 001206 334040 001654' SKIPA TAC,BFPNTR ;NO. INTO BUF
715 001207 540046 000014 HRR TAC,DLOC(DEV DAT) ;YES, LOC OF DIRECTORY
716 001210 370000 000001 SOS TAC ;NO, SET FOR FORWARD READ
717 001211 IOGO: IFN CPBIT, <

718 001211 607000 040000 TLNN IOS,NOBUF ;POINTER ALREADY SET UP IF DIRECT IO
719 >
720 001212 202040 001660' MOVEM TAC,PNTR ;POINTER
721 001213 434100 001667' OR TAC1,UNIT ;UNIT AND PI CHAN
722 001214 202100 001670' MOVEM TAC1,COMAND ;SAVE COMMAND FOR READ OR WRITE
723 001215 262140 000001 POP PDP,TAC ;RESTORE SEARCH COMMAND
724 001216 720200 000000 CONO DC,DCIN ;SET DATA CONTROL TO READ
725 001217 700600 000000 CONO PI,DCON ;TURN ON DC PI CHANNEL
726 001220 505040 000037 HRLI TAC,37
727 001221 700600 000000 CONO PI,PIOFF
728 001222 721201 000000 CONO DTC,(TAC) ;START TAPE MOVING
729 001223 540040 001357' HLRM TAC,DTCINT
730 001224 700600 000000 CONO PI,PION
731 001225 263140 000000 POPJ PDP, ;DISMISS INTERRUPT
732
733
734          ;HERE FOR ANY DATA WORD WITH TAPE IN REVERSE
735 001226 000000 000000 RVERS: 0
736 001227 720000 001660' IOWD: BLKI DC,PNTR ;READ (WRITE) A WORD
737 001230 254000 001234' JRST RVTHRU ;POINTER RAN OUT
738 001231 370000 001660' SOS PNTR ;POINTER HAS TO BACK UP
739 001232 370000 001660' SOS PNTR
740 001233 254520 001226' JEN @RVERS ;DISMISS THE INTERRUPT
    
```

```

741                                     ;HERE WHEN POINTER RUNS OUT IN REVERSE
742 001234 264000 001241' RVTHRU: JSP      SHUTDN      ;SHUT DOWN DATA CONTROL
743 001235 254520 001226'          JEN      @RVERS      ;DISMISS
744
745                                     ;HERE WHEN POINTER RUNS OUT FORWARD
746 001236 000000 000000 DTATHR: ?
747 001237 264000 001241'          JSP      SHUTDN      ;SHUT DOWN DATA CONTROL
748 001240 254520 001236'          JEN      @DTATHR     ;DISMISS
749
750
751 001241 000000 000000 SHUTDN: ?
752 001242 720300 017000          CONSZ   DC,17000      ;DATA MISSED?
753 001243 476000 001071'          SETOM   ERRFLG      ;YES, SET SWITCH
754 001244 721340 000400          CONSQ   CTC,400      ;READING?
755 001245 720200 000000          CONO    DC,0        ;YES, TURN OFF DC
756 001246 700600 000000          CONC   PI,DCOFF     ;TURN OFF DC PI
757 001247 254020 001241'          JRST   @SHUTDN
    
```

```

758                                     ;COME HERE TO START READING BLOCK NUMBERS
759                                     EXTERN PION,PIOFF
760
761 001250 552200 001663' FNDBLK: HRRZM  RLK,BLOCK      ;BLOCK WE'RE LOOKING FOR
762 001251 402000 001665'          SETZM  ERRCNT
763 001252 402000 001671' FNDBL2: SETZM  ERRFLG      ;RESET DATA MISSED FLAG
764 001253 135040 000000          LDB   TAC,PUNIT  ;GET UNIT NUMBER
765 001254 242040 000003          LSH   TAC,3      ;POSITION IT
766 001255 660040 000000          TRD   TAC,DTCCHN   ;ADD PI CHANNEL
767 001256 202040 001667'          MOVEM  TAC,UNIT  ;SAVE
768 001257 200400 002111'          MOVE  TEM,[JSR SRCW] ;SET UP INTERRUPT CELLS
769 001260 202400 000000          MOVEM  TEM,DCLOC
770 001261 200400 002112'          MOVE  TEM,[JSR DTATHR]
771                                     IFN  CPBIT, <
772 001262 607000 040000          TLNN  IOS,NOBUF  ;DCLOC1 ALREADY SET UP IF DIRECT IO
773                                     >
774 001263 202400 001120'          MOVEM  TEM,DCLOC1 ;FOR FORWARD DATA OPERATIONS
775 001264 721300 020000          CONSZ  DTC,20000 ;IS TAPE ALREADY MOVING?
776 001265 254000 001277'          JRST  FNDBL4    ;YES
777 001266 660040 323200          TRD   TAC,323200 ;NO. SET READ BLK NOS., START DELAY
778 001267 607000 001000          TLNN  IOS,RVERSE ;MOVE TAPE BACKWARDS?
779 001270 660040 010000          TRD   TAC,10000  ;YES
780 001271                                     FNDBL3: IFN  CPBIT, <
781 001271 261140 000000          PUSH  PDP,IOS
782 001272 621000 737777          TLZ   IOS,NORUFC ;SET UP DIRCTN NON-0 IF DIRECT IO
783 001273 512000 001676'          HLLZM IOS,DIRCTN
784 001274 262140 000000          POP   PDP,IOS
785                                     >
786 001275 202006 000002          MOVEM  IOS,DEVIOS(DFVDAT) ;RESTORE IOS IN CORE
787 001276 263140 000000          POPJ  PDP,      ;AND EXIT
788
789
790                                     ;HERE TO INITIATE READ OF A TAPE THAT IS MOVING ALREADY
791 001277 721300 010000          FNDBL4: CONSZ  DTC,10000 ;GOING BACKWARDS?
792 001300 660040 010000          TRD   TAC,10000  ;YES
793 001301 660040 320200          TRD   TAC,320200 ;SET TO READ BLOCKS, NO DELAY
794 001302 254000 001271'          JRST  FNDBL3    ;GO TELL CONTROL
    
```

```

795 ;INTERRUPT HERE TO READ A BLOCK NUMBER
796 001303 000000 000000 SRCH: 0
797 001304 202040 001661' MOVEM TAC,TEMP ;SAVE TAC
798 001305 720040 000001 DATA DC,TAC ;READ A BLOCK NUMBER
799 001306 274040 001663' SUB TAC,BLOCK ;PRESENT BLOCK - TARGET BLOCK
800 IFN CPBIT, <
801 001307 332000 001676' SKIPE DIRCTN ;IF DIRECT IO
802 001310 721340 010000 CONSO DTC,10000 ;MUST READ DATA FORWARD
803 >
804 001311 322040 001340' JUMPE TAC,FOUND ;THERE IF ZERO
805 001312 540040 001667' HRR TAC,UNIT ;NO. SET UP READ OF NEXT BLOCK
806 001313 721300 010000 CONSZ DTC,10000 ;GOING IN REVERSE
807 001314 650040 002113' TDC TAC,[XWD 400000,10000] ;YES, SWITCH TURN-AROUND TEST
808 001315 331000 000001 SKIPL TAC ;TURN AROUND?
809 001316 640040 012000 TRC TAC,12000 ;YES, SWITCH DIR., TURN AROUND DELAY
810 001317 721201 320200 CONO DTC,320200(TAC);TELL CONTROL
811 001320 720200 001216' CONO DC,DCIN ;TELL DATA CONTROL ALSO
812 001321 200040 001661' SRCHXT: MOVE TAC,TEMP ;RESET TAAC
813 001322 254520 001303' JEN @SRCH ;AND EXIT THE INTERRUPT
814
815
816 IFN CPBIT, <
817 ;HERE WHEN DUMP-MODE POINTER RUNS OUT
818 001323 000000 000000 DMPADV: 0
819 001324 202040 001661' MOVEM TAC,TEMP
820 001325 354040 001673' AOSA TAC,USPNTR ;ADVANCE LOC OF POINTER
821 001326 542040 001673' DMPAV1: HRRM TAC,USPNTR
822 001327 336060 000001 SKIPN TAC,@TAC ;END OF LIST?
823 001330 254000 001336' JRST DMPAV3 ;YES, STOP TAPE
824 001331 270040 001675' ADD TAC,ADRPRG ;ADD RELOCATION
825 001332 327040 001326' JUMPG TAC,DMPAV1
826 001333 202040 001660' MOVEM TAC,PNTR ;NEW POINTER
827 001334 200040 001661' DMPAV2: MOVE TAC,TEMP ;RESTORE TAC
828 001335 254520 001323' JEN @DMPADV
829
830
831 001336 264000 001241' DMPAV3: JSR SHUTDN ;SHUT DOWN TAPE
832 001337 254000 001334' JRST DMPAV2 ;RESTORE TAC AND EXIT THE INTERRUPT
833
834 >
    
```

```

835
836 001340 721370 012000 ;HERE WHEN CORRECT BLOCK NUMBER IS FOUND
837 001341 254000 001351' FOUND: CONSZ DTC,10000 ;GOING FORWARD?
838 001342 202040 001227' JRST IORVRS ;NO
839 001343 202040 001260' FND1: MOVE TAC,IOWD ;YES, SET BLKI/BLKO INTO PI LOC
840 001344 202040 001670' MOVE TAC,DCLOC
841 001345 602040 000400' TRNE TAC,COMAND ;PICK UP READ OR WRITE COMMAND
842 001346 722200 000070' CONU TAC,400 ;WRITE?
843 001347 721271 360070' CONO DC,DCOUT ;YES, CONDITION DATA CONTROL
844 001350 254000 001321' JRST DTC,360000(TAC) ;START DATA FLOW
845 SRCHYT ;AND LEAVE
846 001351 201040 010070' IORVRS: MOVEI TAC,10000 ;SET IO FOR REVERSE
847 001352 436040 001670' ORM TAC,COMAND
848 001353 201040 000177' MOVEI TAC,177
849 001354 272040 001660' ADDM TAC,PNTR ;READ FROM TOP OF BUFFER DOWN
850 001355 200040 002114' MOVE TAC,[JSR RVERS] ;HAVE TO DO SOME STUFF AT INTERRUPT
851 001356 254000 001343' JRST FND1
    
```

```

852                                     ;INTERRUPT HERE FOR FLAG CHANNEL
853 001357 721740 000037 DTCINT: CONSO DTS,37 ;INTERRUPT FOR DECTAPE?
854 001360 254070 001360' JRST . ;NO. GO AWAY
855 001361 721340 040000 CONSO DTC,40000 ;YES. JOB DONE ENABLED?
856 001362 254000 001522' JRST RLKNUM ;NO. READING BLOCK NUMBERS
857 001363 721370 004000 CONSZ DTC,4000 ;YES. TIME FLAG ENABLED?
858 001364 254000 001510' JRST TIMINT ;YES. CHECK IF THIS IS A TIME INTERRUPT
859 001365 720200 000000 DTCIN1: CONO DC,0 ;NO. TURN OFF DATA CONTROL
860 001366 264070 000000 JSR DTCSAV ;SAVE ACS
861 001367 200320 001656' MOVE DEVDAT,USEWRD ;RESTORE DEVDAT
862 001370 200006 000002 MOVE IOS,DEVIOS(DEVDAT) ;AND IOS
863 001371 200340 001657' MOVE PROG,USEPRG
864 001372 721700 000001 CONSZ DTS,1 ;JOB DONE LIT?
865 001373 721700 000116 CONSZ DTS,116 ;AND NO ERRORS?
866 001374 254000 001616' JRST ERRS ;NO. ERROR
867 001375 332030 001671' SKIPE ERRFLG ;DATA MISSED?
868 001376 254000 001616' JRST ERRS ;YES. TOO BAD
869
870                                     ;DATA WAS READ IN OR WRITTEN OUT FINE
871 001377 603030 004000 DTCIN2: TLNE IOS,DMPMOD ;DUMP MODE?
872 001400 254000 001567' JRST DMPTHR ;YES. GO ELSEWHERE
873 001401 607000 002000 TLNN IOS,SINGL
874 001402 254000 001405' JRST .+3
875 001403 135040 002105' LDB TAC,(POINT 10,RUF,27) ;GET 1ST BLOCK NO. IF READ
876
877 001404 202046 000015 MOVEM TAC,IBLK(DEVDAT) ;CAME FROM LOOKUP
878 001405 623000 000001 TLZE IOS,IOW ;STORE IN DDR
879 001406 260140 000000 PUSHJ POP,SETIOD ;NO. IN IO WAIT?
880 001407 623000 012000 TLZE IOS,SINGL+RWDIR ;YES. TAKE OUT OF WAIT
881 001410 254000 001445' JRST THRUTP ;DIRECTORY OPERATION?
882 001411 603000 000020 TLNE IOS,IO ;YES. LEAVE
883 001412 254000 001467' JRST RUTHRU ;WRITING?

```



```

916                                     ;HERE WHEN TAPE IS DONE
917 001445 721340 020000 THRUTP: CONSO DTC,20000 ;ION INTERRUPT LEVEL?
918 001446 254000 001457' JRST THRUTD ;NO. TAPE IS NOT MOVING
919 001447 200040 001667' MOVE TAC,UNIT ;SET TO STOP TAPE
920 001450 721300 010000 CONSZ DTC,10000 ;GOING REVERSE?
921 001451 660040 010000 TRO TAC,10000 ;YES
922 001452 721201 245000 CONO DTC,245000(TAC) ;STOP TAPE, WITH TIME FLAG INTERRUPT ON
923                                     ;ENABLE JOB DONE AS A FLAG FOR DTCINT
924 001453 371000 000000 THRUTA: SOSL DCREQ ;GIVE UP DATA CONTROL (DECTAPE CONTROL
925 001454 476000 000000 SETOM DCAVAL ;WILL BE GIVEN UP AT NEXT INTERRUPT)
926 001455 621000 044000 TLZ IOS,DMPMOD+NOBUF ;RESET DUMP-MODE BIT
927 001456 254000 000000 JRST CLRACT ;RESET IOACT AND RETURN
928
929 001457 260140 001453' THRUTD: PUSHJ PDP,THRUTA ;GIVE UP DATA CONTROL
930 001460 371000 001436' SOSL DTREQ ;GIVE UP DECTAPE CONTROL
931 001461 476000 000000 SETOM DTAVAL
932 001462 720200 000000 CONO DC,0 ;SHUT OFF DATA CONTROL
933 001463 700600 001246' CONO PI,DCOFF ;AND TURN OFF ITS PI CHANNEL
934 001464 721200 000000 CONO DTC,0 ;SHUT DOWN DATA CONTROL
935 001465 513000 001357' HLLZS DTCINT
936 001466 263140 000000 POPJ PDP, ;AND LEAVE
    
```

```

937                                     EXTERN  CLOCK
938
939                                     ;HERE ON END OF OUTPUT BLOCK
940 001467 260140 001063' OUTHRU: PUSHJ  PDP,ADV8FE  ;GET NEXT BUFFER
941 001470 254000 001445'         JKST   THRUTP   ;NOT FULL
942 001471 322200 001445'         JUMPE  BLK,THRUTP ;DONE IF BLK=0
943 001472 550206 000016'         HRRZ   BLK,0BLK(DEVDAT);NEXT BLOCK TO WRITE
944 001473 303200 001101'         CAILF  BLK,TOPBLK ;LEGAL?
945 001474 254000 001445'         JRST   THRUTP   ;NO. CATCH ERROR ON UOO LEVEL
946 001475 375000 001664'         SOSGE  QUANTM   ;YES, HAD CHAN LONG ENOUGH?
947 001476 337000 001460'         SKIPG  DTREQ    ;AND SOMEONE ELSE WANT IT?
948 001477 254000 001012'         JRST   FILBUF   ;NO, GO WRITE NEXT BLOCK
949 001500 254000 001445'         JRST   THRUTP   ;YES, GIVE UP TAPE
950
951
952                                     ;TURN TAPE AROUND AFTER END-ZONE INTERRUPT
953 001501 202040 001662' TURN:  MOVEM  TAC,TEMPA ;SAVE TAC
954 001502 200040 001667'         MOVE   TAC,UNIT  ;UNIT AND CHANNEL
955 001503 721340 010000'         CONSO  DTC,10000 ;IN REVERSE?
956
956 001504 660040 010000'         TRD    TAC,10000 ;NO, NO WILL BE
957 001505 721201 322200'         CONO  DTC,322200(TAC);READ BLOCK NOS IN OPPOSITE DIRECTION
958 001506 200040 001662'         MOVE  TAC,TEMPA ;RESTORE TAC
959 001507 254520 000000'         JEN    @DTCCHL  ;AND EXIT THE INTERRUPT
960
961 001510 721740 000020' TIMINT: CONSO  DTS,20   ;TIME FLAG INTERRUPT ON?
962 001511 254000 001527'         JRST  SPRIUS   ;NO. THIS IS A SPURIOUS INTERRUPT
963 001512 250040 001532'         EXCH  TAC,TIMREQ ;ANOTHER TAPE CAN NOT BE
964 001513 700600 001221'         CONO  PI,PIOFF  ;SELECTED FOR 25 MSEC SO
965 001514 136040 000000'         IDPB  TAC,CLOCK  ;PUT IN A CLOCK REQUEST
966 001515 700600 001224'         CONO  PI,PION ;TO WAIT FOR 3 TICKS
967 001516 250040 001532'         EXCH  TAC,TIMREQ ;BEFORE ALLOWING NEXT USER ON
968 001517 721200 000000'         CONO  DTC,2    ;SHUT DOWN DTC
969 001520 513000 001357'         HLLZS DTCINT
970 001521 254520 001507'         JEN    @DTCCHL  ;AND LEAVE
971
972
973                                     ;HERE WITH AN INTERRUPT WHILE SEARCHING FOR BLOCK NUMBERS
974 001522 721700 000002' BLKNUM: CONSZ  DTS,2   ;END ZONE INTERRUPT?
975 001523 254000 001501'         JRST  TURN    ;YES, TURN AROUND
976 001524 721740 000014'         CONSO DTS,14   ;NO, ILLEGAL OP OR PARITY ERROR?
977 001525 254000 001360'         JRST  DTCINT+1 ;NO, NOT REALLY A DECTAPE INTERRUPT
978 001526 254000 001365'         JRST  DTCINT1  ;YES, GO HANDLE THE ERROR
979
980                                     ;HERE ON A SPURIOUS DECTAPE INTERRUPT
981 001527 721240 001662' SPRIUS: CONI  DTC,TEMPA ;READ DTC STATUS BITS
982 001530 721220 001662'         CONO  DTC,@TEMPA ;GIVE A CONO TO DTC TO CLEAR INTERRUPT
983 001531 254520 001521'         JEN    @DTCCHL  ;EXIT THE INTERRUPT
984
985 001532 001533' 000003' TIMREQ: XWD    .+1,3  ;E CLOCK TICKS
986 001533 371000 001476'         SOSL  DTREQ    ;ALLOW OTHER JOBS
    
```

DTCSRN - NEW FORMAT DECTAPE SERVICE FOR 551 (PDP-6) MACRO,V36 19:06 4-JUN-69 PAGE 41
DTA551 A,WACHS/TW/RCC 01 JUN 69 V012

987 001534 476070 001461' SETOM DTAVAL ;TO GET DT CONTROL NOW
988 001535 263140 000000 POPJ PDP,

```

989                                     ;COME HERE ON END OF DUMP OR SAVE MODE BLOCK
990 001536                               SVDMTH: IFN   CPBIT, <
991 001536 603000 040000                TLNE   IOS,NOBUF
992 001537 254000 001561'              JRST   USDMTH           ;HANDLE DIFFERENTLY IF DIRECT TO USER
993                                     >
994 001540 205100 777601                MOVSI  TAC1,-177
995                                     IFE    CPBIT, <
996                                     TRNE  IOS,UDSD
997                                     SUB   TAC1,ONEONE       ;SET UP TAC1 WITH COUNT
998                                     >
999 001541 607000 000020                TLNN   IOS,IO
1000 001542 254000 001556'              JRST  SVDMIN           ;INPUT FILE
1001 001543 550206 000016              HRRZ  BLK,OBLK(DEV DAT) ;OUTPUT FILE, NEXT BLOCK
1002 001544 322200 001550'              JUMPE BLK,DMPHHA       ;LAST BLOCK
1003                                     IFE    CPBIT, <
1004                                     TRNE  IOS,UDSD           ;IF NON-STD MODE
1005                                     AOSA  OBLK(DEV DAT)    ;WRITE CONSECUTIVE BLOCKS
1006                                     >
1007 001545 307200 001101                CAIG  BLK,TOPLK       ;NOT LAST, LEGAL BLOCK NUMBER?
1008 001546 263140 000000                POPJ  PDP,             ;YES, RETURN
1009
1010 001547 660000 040000                TRO   IOS,IORCTL      ;BLOCK TOO LARGE
1011 001550 262140 000001                DMPHHA: POP          PDP,TAC
1012 001551 402006 000021                DMPH2: SETZM        SVDWRD(DEV DAT) ;ZERO DUMP-MODE STUFF
1013 001552 402006 000020                DMPH3: SETZM        DMPLST(DEV DAT)
1014 001553 623000 000001                DMPH3: TLZE        IOS,IOW           ;IN IO WAIT?
1015 001554 260140 001406'              PUSHJ PDP,SETIOD      ;YES, RESTART JOB
1016 001555 254000 001445'              JRST  THRUTP
1017
1018                                     ;HERE ON END OF SAVE MODE INPUT BLOCK
1019 001556 554200 001677'              SVDMIN: HLRZ        BLK,BUF           ;NEXT BLOCK NUMBER
1020                                     IFE    CPBIT, <
1021                                     TRNE  IOS,UDSD           ;NON-STANDARD?
1022                                     AOSA  BLK,IBLK(DEV DAT) ;YES, READ CONSECUTIVE BLOCKS
1023                                     >
1024 001557 542206 000015                HRRM  BLK,IBLK(DEV DAT) ;SAVE IN DDB
1025 001560 254000 001167'              JRST  CPOPJ1
1026
1027
1028                                     IFN CPBIT, <
1029                                     ;HERE WHEN THROUGH DUMP-MODE DIRECTLY TO USER
1030 001561 201106 000015                USDMTH: MOVEI      TAC1,IBLK(DEV DAT)
1031 001562 603000 000020                TLNE   IOS,IO
1032 001563 201106 000016                MOVEI  TAC1,OBLK(DEV DAT)           ;SET TAC1 TO RIGHT BLOCK NUMBER
1033 001564 200040 001672'              MOVE  TAC,BLKCNT      ;UPDATE BLOCK COUNTER
1034 001565 272042 000000                ADDM  TAC,(TAC1)
1035 001566 254000 001550'              JRST  DMPHHA           ;THROUGH
1036                                     >
    
```

```

1037
1038      001567 260140 001536' ;HERE WHEN THROUGH DUMP MODE BLOCK
1039      001570 254000 001173' DMPTHR: PUSHJ PDP,SVDWTH ;END OF BLOCK HOUSEKEEPING
1040      ;JRST DMPFLC ;FILL BUFFER FOR NEXT OUTPUT
1041
1042      ;HERE WHEN THROUGH READING A DUMP-MODE BLOCK
1043      001571 332046 000021 DMIFIL: SKIPE TAC,SVDWRD(DEVDAT) ;PARTIAL COMMAND?
1044      001572 254000 001575' JRST DMIFLB ;YES. CONTINUE
1045      001573 260140 001157' DMIFLA: PUSHJ PDP,NXTCOM ;NO. GET NEXT COMMAND
1046      001574 254000 001551' JRST DMPH2 ;END OF LIST - THROUGH
1047      001575 200402 001700' DMIFLB: MOVE TEM,BUF+1(TAC1) ;NEXT DATA WORD
1048      001576 202401 000000 MOVEM TEM,(TAC) ;GIVE TO USER
1049      001577 252100 001602' AOBJP TAC1, +3 ;IF BUFFER IS FULL
1050      001600 253040 001575' AOBJN TAC,DMIFLB ;GET NEXT WORD
1051      001601 254000 001573' JRST DMIFLA ;GET NEXT COMMAND
1052
1053      001602 253040 001605' AOBJN TAC, +3 ;BUFFER ID FULL. IS COUNT EXACTLY 177?
1054      001603 260140 001157' PUSHJ PDP,NXTCOM ;THAT COM. IS DONE. GET NEXT
1055      001604 254000 001551' JRST DMPH2 ;END OF LIST - THROUGH
1056      001605 202046 000021 MOVEM TAC,SVDWRD(DEVDAT) ;SAVE PARTIAL COMMAND FOR NEXT TIME
1057      001606 322200 001612' JUMPE BLK,DMPEOF ;IF EOF - LIGHT BIT
1058      001607 307200 001101 RONXT: CAIG BLK,TOPBLK ;BLOCK LEGAL?
1059      001610 254000 001200' JRST READBC ;GO READ BLOCK NUMBER
1060      001611 664000 040000 TROA IOS,IOBKTL ;LIGHT ERROR BIT
1061
1062
1063      ;EOF BEFORE ALL DATA IS IN - DUMP MODE
1064      001612 660000 020000 DMPEOF: TRO IOS,IODEND ;LIGHT EOF BIT
1065      001613 254000 001551' JRST DMPH2 ;GIVE UP TAPE
    
```

```

1065 001614 260140 001551' SVADER: PUSHJ PDP,DMPTH2 ;GIVE UP CONTROL
1066 001615 254000 000000 JRST ADRERR ;TYPE ERROR MESSAGE
1067
1068
1069 ;COME HERE ON ERROR
1070 001616 350040 001665' ERRS: AOS TAC,ERRCNT ;BUMP COUNT
1071 IFN CPBIT, <
1072 001617 200100 001674' MOVE TAC1,SVPNTR
1073 001620 202100 001673' MOVEM TAC1,USPNTR
1074 001621 200102 000000 MOVE TAC1,(TAC1)
1075 001622 270100 001675' ADD TAC1,ADRPRG
1076 001623 202100 001660' MOVEM TAC1,PNTR ;RESET STUFF FOR POSSIBLE DIRECT IO
1077 >
1078 001624 721740 000004 CONSO DTS,4 ;IF ILLEGAL OP - DONT RETRY
1079 001625 303040 000000 CAILE TAC,DTTRY ;ENOUGH REREADS?
1080 001626 334000 000000 SKIPA ;YES. PERMANENT ERROR
1081 001627 254000 001252' JRST FNDBL2 ;NO. TRY AGAIN
1082
1083 ;PERMANENT ERROR
1084 001630 721700 000010 PERMER: CONSZ DTS,10
1085 001631 660000 100000 TRO IOS,IODTER ;PARITY
1086 001632 721740 000100 CONSO DTS,100
1087 001633 332000 001671' SKIPE ERRFLG
1088 001634 660000 200000 TRO IOS,IODERR ;MISSED DATA
1089 001635 721700 000004 CONSZ DTS,4
1090 001636 660000 400000 NOBLK0: TRO IOS,I0IMPM ;ILLEGAL OP
1091 001637 603000 004000 TLNE IOS,DMPMOD ;IF IN DUMP-MODE
1092 001640 254000 001567' JRST DMPTHR ;CONTINUE WITH REST OF LIST
1093 001641 607000 012020 TLNN IOS,I0+RWDIR+SINGL ;READING DATA?
1094 001642 254000 001377' JRST DTCIN2 ;YES. GIVE BUFFER TO USER
1095 001643 623000 000001 TLZE IOS,IOW
1096 001644 260140 001554' PUSHJ PDP,SETIOD ;OUT OF IO WAIT
1097 001645 627000 012000 TLZN IOS,RWDIR+SINGL ;DIRECTORY OPERATION?
1098 001646 254000 001445' JRST THRUTP ;NO. RETURN TO USER
1099 001647 260140 001445' PUSHJ PDP,THRUTP ;YES. STOP TAPE
1100 001650 205040 400000 MOVSJ TAC,400000 ;SET DIR NOT IN CORE BIT
1101 001651 412046 000004 ANDCAM TAC,DEVMOD(DEVDAT)
1102 001652 135200 000000 BDDIR: LDR BLK,PJOBN ;TELL ERRCON THE NUMBER OF THE JOB
1103 001653 254000 000000 JRST BDDIR ;GO PRINT ERROR MESSAGE
    
```

```

1124 001654 777600 001677' RFPNTR: ICWD      200,RUF+1
1125 001655 000001 000001 ONEONE: XWD      1,1
1126 001656 000000 000000 USEWRD: ?
1127 001657 000000 000000 USEPRG: ?
1128 001660 000000 000000 PNTR: ?
1129 001661 000000 000000 TEMP: ?
1130 001662 000000 000000 TEMPA: ?
1131 001663 000000 000000 BLOCK: ?
1132 001664 000000 000000 QUANTM: ?
1133 001665 000000 000000 ERRCNT: ?
1134 001666 000000 000000 FNDTMP: ?
1135 001667 000000 000000 UNIT: ?
1136 001670 000000 000000 COMAND: ?
1137 001671 000000 000000 ERRFLG: ?
1138                                     IFN      CPBIT, <
1139 001672 000000 000000 BLKCNT: ?
1140 001673 000000 000000 USPNTR: ?
1141 001674 000000 000000 SVPNTR: ?
1142 001675 000000 000000 ADRPRG: ?
1143 001676 000000 000000 DIRCTN: ?
1144                                     >
1145 001677                                     RUF:   BLOCK  200
1146 002077                                     DATAEND: END
1147 002077 777777 777601
1148 002100 777752 000123
1149 002101 050000 000000
1150 002102 000026 000026
1151 002103 000040 020000
1152 002104 000001 001677'
1153 002105 101200 001677'
1154 002106 720100 000700
1155 002107 264000 001323'
1156 002110 720000 000300
1157 002111 264000 001303'
1158 002112 264000 001236'
1159 002113 400000 010000
1160 002114 264000 001226'
1161 002115 001677' 000001
  
```

NO ERRORS DETECTED

PROGRAM BREAK IS 002116

AC1	000015	INT	AC2	000016	INT	AC3	000017	INT
ADRERR	001615	EXT	ADRPRG	001675	INT	ADVBFE	001467	FXT
ADVBFF	001430	EXT	RADDR	001653	FXT	RDIR	001652	FXT
RFPNTR	001654	FXT	BLK	000074	FXT	BLKCHK	000752	FXT
BLKCNT	001672	FXT	BLKNUM	001522	FXT	BLKSRA	000420	FXT
BLKSRB	000423	FXT	BLKSPC	000415	FXT	BLOCK	001663	FXT
RUF	001677	FXT	CALVXT	000624	FXT	CHNGDR	000400	FXT
CLOCK	001514	EXT	CLRACT	001456	EXT	CLSDMP	000721	FXT
COMMAND	001670	EXT	COMCHK	001073	EXT	CUR2HM	001052	FXT
CPBIT	777777	FXT	CPPOPJ	000547	EXT	CPPOPJ1	001560	FXT
DAT	000005	INT	DCAVAL	001454	EXT	DCIN	001320	FXT
DCLOG	001343	EXT	DCLOC1	001263	EXT	DCOFF	001463	FXT
DCON	001217	EXT	DCOUT	001346	EXT	DCREJ	001453	FXT
DECPTR	000455	FXT	DEVCAT	000006	INT	DEVEXT	000012	INT
DEVFIL	000011	INT	DEVIAD	000007	INT	DEVIOS	000002	INT
DEVMOD	000004	INT	DEVUAD	000010	INT	DIRBLK	000144	FXT
DIRCHK	000401	FXT	DIRCTN	001676	FXT	DISPAD	000017	FXT
DLFTF	000464	FXT	DLOC	000014	FXT	DMIFIL	001571	FXT
DMIFLA	001573	FXT	DMIFLB	001575	FXT	DMPADV	001323	FXT
DMPAV1	001326	FXT	DMPAV2	001334	FXT	DMPAV3	001336	FXT
DMPCLS	020000	FXT	DMPDF	001612	FXT	DMPFIL	001130	FXT
DMPFLA	001133	FXT	DMPFLB	001131	FXT	DMPFLC	001173	FXT
DMPFI	000725	FXT	DMPPLT	000020	FXT	DMPMOD	004000	FXT
DMPD	000773	FXT	DMPPTH	001140	FXT	DMPQVA	001154	FXT
DMPQVR	001150	FXT	DMPSET	001070	FXT	DMPST2	001101	FXT
DMPTH2	001551	FXT	DMPTH3	001553	FXT	DMPTHA	001550	FXT
DMPTHR	001567	FXT	DMPTH4	001104	FXT	DRPTR	000543	FXT
DSER1	000360	FXT	DMPST1	001104	FXT	DTAEND	000277	FXT
DTATHR	001236	FXT	DSERCH	000356	FXT	DTAEND	000210	FXT
DTCHL	001531	EXT	DTAVAL	001534	EXT	DTCDBB	000000	INT
DTCHS	000222	INT	DTCCHN	001255	EXT	DTCDSP	000224	FXT
DTCINI	001365	FXT	DTCCIR	000022	FXT	DTCINI	000243	INT
DTCINT	001357	INT	DTCIN2	001377	FXT	DTCSRN	000200	INT
DTCHK	001012	FXT	DTCSAV	001366	EXT	DTS	000214	FXT
DTTRY	001625	EXT	DTREC	001533	EXT	ENTR	000472	FXT
ENTR2	000476	FXT	DVDIR1	000000	INT	ENTRC	000526	FXT
ENTR3	000570	FXT	ENTRA	000506	FXT	ERRCNT	001665	FXT
ERRFLG	001671	FXT	EOF	000756	FXT	FAKADV	001062	FXT
FILBIF	001012	FXT	FRPS	001616	FXT	FNDBL2	001252	FXT
FNDRL3	001271	FXT	FND1	001343	FXT	FNDBLK	001250	FXT
FNDTMP	001666	FXT	FNDBL4	001277	FXT	FREXIT	000440	FXT
FSTBLK	000013	FXT	FOUND	001340	FXT	FTDISK	777777	FXT
FLOGI	777777	FXT	FICCL	777777	FXT	FTSWAP	777777	FXT
GETDGD	000712	EXT	FTRC10	777777	FXT	GETF	000602	FXT
HUNGST	000011	INT	GETJT	000712	FXT	IO	000020	INT
IOBKTL	000000	INT	IBLK	000015	FXT	IODERR	000000	INT
IOCTFK	100000	INT	IODEND	000000	INT	IOGO	001211	FXT
IOIMPM	000000	INT	IOEND	000000	INT	IOS	000000	INT
IOW	000001	INT	IORVRS	001351	FXT	JBADR	000000	FXT
JBTSST	000000	EXT	IOWD	001227	FXT	JORSAV	000000	FXT
LOOK	000247	FXT	JORDOT	000762	EXT	LOOKC	000307	FXT
LOOKD	000331	FXT	LOOKA	000256	FXT	LSTFRE	000442	FXT
MARKDR	000536	FXT	LSTBLK	001050	FXT	NEWENT	000553	FXT
			NAMSTR	000123	FXT			

NMFOUN	000372'	NMLOOK	000363'	NOBLK0	001636'
NOPLKS	000640'	NORUF	040000	NORUF0	737777
NOLINK	000200	NXTCM1	001162'	NXTCM2	001170'
NXTCOM	001157'	NXTFRE	000433'	OBLK	000016
ONEONE	001655'	OUCOMP	001031'	OUFULL	001022'
OUT	000676' EXT	OUTBL2	001043'	OUTBLK	001033'
OUTHRU	001467'	PDP	000003 INT	PERMER	001630'
PIOFF	001513' EXT	PIOMOD	000666' EXT	PION	001515' FXT
PJOBM	001652' EXT	PNTR	001660'	PROG	000007 INT
PUNIT	001253' EXT	QUANT	000005	QUANTM	001664'
RDNXT	001607'	RQWRT	001202'	READBC	001200'
READRF	001176'	RELA9	000000 EXT	RENAM	000315' EXT
RENAM1	000332'	RENAM2	000335'	RENER1	000350'
RENER2	000352'	PVER5	001226'	RVERSE	001000
RVTHRU	001234'	RWDIR	010000	SD	000015 INT
SETACT	000720' EXT	SETI	000574'	SETI00	001644' FXT
SETO	000576'	SETPTR	000426'	SETWD4	000560'
SHUTDN	001241'	SINGL	002000	SPACE	000006
SPRIUS	001527'	SRCH	001303'	SRCHXT	001321'
STOIOS	001007' EXT	STOTAC	000610' EXT	STOWD4	000271'
SVADER	001614'	SVDMIN	001556'	SVMTH	001536'
SVDWRD	000021	SVPNTR	001674'	TAC	000001 INT
TAC1	000002 INT	TEM	000010 INT	TEMP	001661'
TEMPA	001662'	THRUIN	001440'	THRUTA	001453'
THRUTD	001457'	THRUTP	001445'	THSDAT	000502' FXT
TIMINT	001510'	TIMREQ	001532'	TOPBLK	001101
TOUSFR	001106'	TPOPJ	000364' EXT	TPOPJ1	000000 FXT
TURN	001501'	UADCK1	000357' EXT	UCLS	000663'
UDSD	000100	UIN	000730'	UNIT	001667'
UOUT	000777'	UOUT2	001010'	UREL	000677'
USDMTM	001561'	USFPRG	001657'	USEWRD	001656'
USLSTA	000636'	USPNTR	001673'	USR00T	000771' FXT
USRFR1	000616'	USRFRE	000611'	USRLST	000633'
UTPCLR	000642'	UUD	000014 INT	VDTASR	000012 INT
WAIT1	000711' EXT	WRTBLK	001045'	ZERCOR	000762'

A	6#	6													
AC1	6#	6	119	141	189										
AC2	6#	6	476	529	670										
AC3	6#	273	307												
ADAFER	21	1066													
ADAPRG	631	824	1175	1122#											
ADVBF	21	593	940												
ADVBF	21	510	921												
ADEFER	6#	6													
AL	6#	6													
ASSCON	6#	6													
ASSPRG	6#	6													
R	6#	6													
BADDR	21	1103													
BDDIR	138	1172#													
RFPNTR	714	1174#													
PLK	26#	134	136	144	146	147	214	226	232	233	237	239	240	248	
	255	247	262	309	312	314	315	317	325	330	331	332	333	335	
	376	378	384	385	388	391	392	393	395	401	402	403	409	453	
	484	488	491	524	539	540	542	557	558	559	564	568	569	571	
	572	580	751	895	897	898	923	924	942	943	944	1001	1002	1007	
	1019	1224	1256	1057	1172										
BLKCHK	489	574#	543												
BLKCNT	623	1033	1119#												
BLKNUM	856	974#													
BLKSPA	228#	233	247												
BLKSPB	126	232#	279												
BLKSRC	122	137	224#	275	358										
BLOCK	761	799	1111#												
BUF	471	550	551	552	557	564	566	645	657	658	659	660	663	675	
	875	890	892	895	1219	1046	1174	1125#							
BUFFNT	6#	6													
BUFFRD	6#	6													
CALNXT	394#	405													
CHNGDR	51#	165	326	416	449	589									
CLKR	6#	6													
CLOCK	937	965													
CLRACT	19	927													
CLSDMP	435	469#	469												
CLSI	6#	6													
CLSOUT	6#	6													
CMWB	6#	6													
COMMAND	722	840	847	1116#											
COMCHK	22	672													
COR2HM	541	574#													
CORCNT	6#	6													
CPRIT	60#	475	528	676	618	641	654	672	717	771	780	800	816	990	
	995	1273	1220	1028	1271	1118									
CPCPJ	21	276	268	270	272	276	373	334							
CPAPJ1	21	179	151	225	230	284	612	635	688	1025					
	6#	6													
DAT	6#	6	170	175	178	224	225	228	239	259	266	267	268	269	
	270	271	272	277	321	322	323	334	336	363	366	368	608	621	

FTKCT	6#													
FTLOGI	10#													
FTMONP	6#													
FTRPV	6#													
FTRA10	6#													
FTRC10	10#													
FTRCHK	6#													
FTREAS	6#													
FTSLEE	6#													
FTSWAP	10#													
FTTALK	6#													
FTTIME	6#													
FTTRAC	6#													
FTTRPS	6#													
FTTTY5	6#													
GETDCD	20	459												
GETDT	216	452	459#	470	545	600	704							
GETF	94	372#												
HSAMSK	6#	6												
HSAPDS	6#	6												
HSASIZ	6#	6												
HUNGCT	6#	6												
HUNGST	6#	6	68											
I	6#	6												
IB	6#	6												
IBLK	40#	141	144	301	368	476	484	877	897	898	1024	1030		
IBUFR	6#	6												
ICLOSB	6#	6												
ILM	6#	6												
ILUERR	6#	6												
INRFR	6#	6												
INITR	6#	6												
INPB	6#	6												
IO	6#	6	132	215	451	483	535	634	882	999	1031	1093		
IOACT	6#	6												
IOREG	6#	6												
IOBKTL	6#	6	507	1010	1059									
IOROT	6#	6												
IOCON	6#	6												
IOEND	6#	6	362	1063										
IOERR	6#	6	899	1088										
IODTER	6#	6	899	1085										
IOEND	6#	6	362	510										
IOFST	6#	6												
IOGO	717#													
IOIMPM	6#	6	899	1090										
IONRCK	6#	6												
IOPAR	6#	6												
IORVRS	837	846#												
IOS	6#	6	108	132	133	165	211	215	217	283	310	320	326	362
	389	398	412	416	430	431	440	447	449	451	469	483	486	490
	507	510	511	535	536	544	547	554	555	560	563	589	592	595
	599	607	632	634	703	713	718	772	778	781	782	783	784	786

UWPOFF	6#	6				
VDTASR	14#	14				
WAITI	21	140	219	367	372	455
WRTBLK	454	576#				
WTMASK	6#	6				
ZERCOR	479	517#				

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