ATARI®

810™

DOS UTILITIES SOURCE LISTING (DOS II)

ATARI®
A Warner Communications Company

C017894

ATARI® 400/800™

DOS UTILITIES
SOURCE LISTING
(DOS II)

NOTICE

TO ALL PERSONS RECEIVING THIS DOCUMENT

AUGUST 1981

```
ERR LINE  ADDR  B1 B2 B3 B4      DISK UTILITY PROGRAMS (DUP)  VER 2.9  11/18/80                PAGE    1

    1                                .TITLE  'DISK UTILITY PROGRAMS (DUP)  VER 2.9  11/18/80'
    2                                LIST X
    3                           ;
    4                           ;
    5                           ;CHANGED FOR SYSTEM RESET -- DUPFLG
    6                           ;ADDED INTERRUPT ROUTINES FROM SIO -- KB
    7                           ;ADDED SAVE/RESTORE OF DOSINI VECTOR -- KB
    8                           ;
    9                           ;*****************************************************************************
   10                           ;       THIS IS FINAL VERSION OF DUP  ---- 2.0S ----
   11                           ;*****************************************************************************
   12                           ;
   13                           ;       FILENAME = DOS2.DUP20S ON TANDEM
   14                           ;
```

```
                        .PAGE
                ;  ****  EQUATES  ****
                ;
                ;
                ;
15              CIO     =       $E456
16              DKHND   =       $E453
17              SETVBV  =       $E45C
18              SYSVBV  =       $E45F
19              XITVBV  =       $E462
20      E456    CIOINV  =       $E46E
21      E453    MEMTOP  =       $2E5
22      E45C    BRKKEY  =       $11
23      E45F    DOSVEC  =       $A
24      E462    DOSINI  =       $C              ;DOS INIT VECTOR
25      E46E    WARMST  =       8
26      02E5    LMARGN  =       $52
27      0011    RMARGN  =       $53
28      000A    CARTST  =       $BFFA
29      000C    INTRVEC =       $20A            ;INTERRUPT VECTOR LOC FOR SIO PATCH
30      0008    MEMLO   =       $2E7
31      0052    SHFLOK  =       $2BE
32      0053    INITAD  =       $2E2
33      BFFA    RUNAD   =       $2E0
34      020A    ICHIDZ  =       $20
35      02E7    ICDNOZ  =       $21
36      02BE    ICBALZ  =       $24
37      02E2    ICBAHZ  =       $25
38      02E0    ICIDNO  =       $2E
39      0020    MAXDEV  =       $21
40      0021    HATABS  =       $31A
41      0024    USRDOS  =       $1700
42      0025    FMS     =       $700
43      002E    FMINIT  =       FMS+$E0
44      0021    DOS     =       FMS+$E40
45      031A    WRMSTR  =       $E474           ;WARM START VECTOR
46      1700    BSIOR   =       $772            ;ENTRY POINT TO FMS DISK HANDLER USED BY
47      0700    CDTMV3  =       $21C            ;ADDRESS OF SYSTEM TIMER # 3
48      07E0    CDTMF3  =       $22A            ;ADDRESS OF SYS TIMER # 3 TIME OUT FLAG
49      1540    ;
50      E474    CR      =       $9B
51      0772    CUP     =       $1C
52      021C    CDN     =       $1D
53      022A    CLF     =       $1E
54              CRT     =       $1F
55      009B    DLL     =       $9C
56      001C    CLSCR   =       $7D
57      001D    EOF     =       $88             ;ENDFILE RETURN CODE FROM CIO
58      001E    ;
59      001F    ;
60      009C    OPEN    =       $03
61      007D    CLOSE   =       $0C
62      0088    PUTCHR  =       $0B
63              GETCHR  =       $07
64
65      0003
66      000C
67      000B
68      0007
```

```
ERR LINE  ADDR  B1 B2 B3 B4

     69  0005
     70  0009
     71  0020
     72  0021
     73  00FE
     74  0023
     75  0024
     76  0053
     77
     78  0010
     79
     80  02EA
     81
     82
     83  0300
     84  0301
     85  0302
     86  0303
     87  0304
     88  0305
     89  030A
     90  030B
     91
     92  0340
     93  0340
     94  0341
     95  0342
     96  0343
     97  0344
     98  0345
     99  0348
    100  0349
    101  034A
    102  034B
    103
    104  0000
    105  0008
    106  000C
    107
    108
    109
    110
    111
```

```
GETREC  =       $05
PUTREC  =       $09
RENAME  =       $20
DELETE  =       $21
FORMAT  =       $FE
LOCK    =       $23
UNLOCK  =       $24
STAREQ  =       $53             ;STATUS COMMAND TO DISK CONTROLLER
;
IOCB1   =       $10
;
DVSTAT  =       $2EA            ;ADDRESS OF STATUS INFO STORED BY OS
;
;
DCB     =       $300
DUNIT   =       DCB+1
DCOMND  =       DCB+2
DSTATS  =       DCB+3
DBUFLO  =       DCB+4
DBUFHI  =       DCB+5
DSLO    =       DCB+$A
DSHI    =       DCB+$B
;
IOCB    =       $340
ICHID   =       IOCB+0
ICDNO   =       IOCB+1
ICCOM   =       IOCB+2
ICSTA   =       IOCB+3
ICBAL   =       IOCB+4
ICBAH   =       IOCB+5
ICBLL   =       IOCB+8
ICBLH   =       IOCB+9
ICAX1   =       IOCB+10
ICAX2   =       IOCB+11
;
SYSED   =       $0
OWRIT   =       $08
ORDWRT  =       $0C
;
HILO    .MACRO  P1
P1&H    =       P1&/256
P1&L    =       (-256)*&P1&H+&P1
        .ENDM
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    112
    113
    114
    115
    116
    117  0018
    118  001A
    119  001A

                .PAGE
;   ****  ZERO  PAGE  VARIABLES  ****
;
;
                *=      $18
JMPTBL:         .RES    2
RAMLO:          .RES    2
BUFADR          =       RAMLO           ;SAVE AREA FOR BUFFER ADDRESS USED BY US
```

```
                        .PAGE
                ;  ****  INIT CODE FOR DUP  ****
                ;
                ;
                ;       INITIALIZATION CODE FOR DUP - CALLS FMS INIT CODE.
                ;       CALLED ON WARM START AND COLD START.
                ;
 120            *=      DOS
 121 1540 A9 00 LDA     #0
 122 1542 8D 9E 15 STA  OPT
 123 1545 A9 9F LDA     #.LOW.MNDUPL
 124 1547 85 0A STA     DOSVEC
 125 1549 A9 17 LDA     #.LOW.MNDUPH
 126 154B 85 0B STA     DOSVEC+1
 127 154D A9 23 LDA     #.LOW.ISRSIR    ;SET UP INTERRUPT VECTORS FOR SIO PATCH
 128 154F 8D 0A 02 STA  INTRVEC         ;INSTEAD OF USING THE SERIAL INPUT READY
 129 1552 A9 1A LDA     #.HIGH.ISRSIR   ;SERVICE ROUTINE AND THE SERIAL OUTPUT
 130 1554 8D 0B 02 STA  INTRVEC+1       ;INTERRUPT SERVICE ROUTINE IN THE OS ROM
 131 1557 A9 E6 LDA     #.LOW.ISRODN    ;USE THE VERSIONS IN RAM FOLLOWING THE
 132 1559 8D 0C 02 STA  INTRVEC+2       ;RESIDENT PORTION OF DUP
 133 155C A9 19 LDA     #.HIGH.ISRODN
 134 155E 8D 0D 02 STA  INTRVEC+3
 135 1561 20 E0 07 JSR  FMINIT
 136 1564 A5 08 LDA     WARMST          ;ON COLDSTART, LOAD AUTORUN SYS
 137 1566 D0 15 BNE     CKMDOS          ;WARMSTART CHECK IF DUP WAS RUNNING
 138 1568 A9 0C LDA     #.LOW.AFL
 139 156A 8D 54 03 STA  ICBAL+$10
 140 156D A9 17 LDA     #.LOW.AFH
 141 156F 8D 55 03 STA  ICBAH+$10
 142 1572 20 93 15 JSR  INITX           ;CLEAR DUPFLG SHOW DUP NOT IN MEMORY.
 143 1575 A9 C0 LDA     #$C0
 144 1577 20 A6 15 JSR  STLOAD          ;LOAD, INIT AND RUN THE AUTORUN FILE
 145 157A 4C AA 19 JMP  CLOSX           ;MAKE SURE IOCB #1 IS CLOSED & RETURN
 146            ;
 147 157D AD 9D 15 CKMDOS LDA  DUPFLG   ;SEE IF DUP WAS IN MEMORY
 148 1580 F0 11    BEQ  INITX           ;=ZERO THEN WASN'T
 149            ;
 150 1582 AD 9E 17 LDA  MEMFLG          ;SEE IF USER AREA WRITTEN TO MEM.SAV
 151 1585 F0 12    BEQ  CLDSET          ;=ZERO THEN WASN'T
 152 1587 20 3F 19 JSR  LDMEM1          ;ELSE GET USER MEMORY BACK IN
 153            ;
 154 158A 20 2E 19 JSR  RELDIN          ;RELOAD SAVED DOSINI VECTOR
 155 158D 20 93 15 JSR  INITX           ;CLEAR DUP IN MEMORY FLAG
 156 1590 20 74 E4 JSR  WRMSTR          ;REDO WARMSTART
 157            ;
 158 1593 A9 00 INITX LDA  #0           ;SAY DUP NOT IN MEMORY
 159 1595 8D 9D 15 STA  DUPFLG          ;CLEAR FLAG
 160 1598 60       RTS
 161            ;
 162 1599 85 08 CLDSET STA  WARMST      ;NO VALID USER MEMORY
 163 159B F0 F6    BEQ  INITX           ;SET TO COLD START
```

```
                        .PAGE
                ;  ****  LOADER ROUTINE  ****
                ;
                ;
                ;       LOADS FROM THE FILE (MUST BE LOAD FORMAT)
                ;       INTO MEMORY.  RETURNS:
                ;         X=0 LOAD OK
                ;         X=1 OPEN ERRORS Y=CIO CODE
                ;         X=2 READ ERRORS Y=CIO CODE
                ;         X=3 BAD LOAD FILE
                ;       ON ENTRY, IOCB 1 POINTS TO FILENAME.
                ;
159D 00         DUPFLG  .BYTE   0               ;FLAG -IF DUP IN MEMORY NOT ZERO
159E 00         OPT     .BYTE   0               ;HOLDS VALUE OF OPTION GIVEN BY USER
159F 00         LOADFG  .BYTE   0               ;FLAG = $80 IF MEMORY FILE DOESN'T HAVE
15A0            HDBUF:  .RES    4
15A4                    HILO    HDBUF
0015            +HDBUFH   =       HDBUF/256
00A0            +HDBUFL   =       (-256)*HDBUFH+HDBUF
15A4
15A4 A9 80      SFLOAD  LDA     #$80
15A6 8D 9F 15   STLOAD  STA     LOADFG
15A9 A9 47      LOAD    LDA     #.LOW.RTS
15AB 8D E0 02           STA     RUNAD
15AE A9 16              LDA     #.HIGH.RTS
15B0 8D E1 02           STA     RUNAD+1         ;MAKE RUN AT EOF DEFAULT TO RTS
15B3 A2 10              LDX     #$10
15B5 A9 03              LDA     #OPEN
15B7 9D 42 03           STA     ICCOM,X
15BA A9 04              LDA     #4              ;OPEN TYPE=INPUT
15BC 9D 4A 03           STA     ICAX1,X
15BF 20 56 E4           JSR     CIO             ;TRY TO OPEN FILE
15C2 10 04              BPL     RDLF            ;CONT IF OK
15C4 A9 01              LDA     #1              ;OPEN ERRORS
15C6 D0 7E              BNE     CLFX            ;CLOSE AND EXIT
15C8 A2 10      RDLF    LDX     #$10
15CA A9 F4              LDA     #.LOW.DBUFL
15CC 9D 44 03           STA     ICBAL,X
15CF A9 1D              LDA     #.LOW.DBUFH
15D1 9D 45 03           STA     ICBAH,X
15D4 A9 02              LDA     #2
15D6 9D 48 03           STA     ICBLL,X
15D9 A9 00              LDA     #0
15DB 9D 49 03           STA     ICBLH,X
15DE 8D 0B 17           STA     MEMLDD          ;CLEAR MEM.SAV LOADED FLAG
15E1 A9 07              LDA     #GETCHR
15E3 9D 42 03           STA     ICCOM,X
15E6 20 56 E4           JSR     CIO
15E9 30 64              BMI     ERST            ;IF ERRS
15EB A9 FF              LDA     #$FF
15ED CD F4 1D           CMP     DBUF            ;CHECK FOR VALID LOAD FILE
15F0 D0 56              BNE     LNLF
15F2 CD F5 1D           CMP     DBUF+1
15F5 D0 51              BNE     LNLF            ;BRANCH IF NOT A LOAD FILE
```

```
ERR LINE  ADDR  B1 B2 B3 B4

     225  15F7  A2 10
     226  15F9  A9 A0
     227  15FB  9D 44 03
     228  15FE  A9 15
     229  1600  9D 45 03
     230  1603  A9 04
     231  1605  9D 48 03
     232  1608  A9 00
     233  160A  9D 49 03
     234  160D  20 56 E4
     235  1610  10 46
     236  1612  C0 88
     237  1614  D0 39
     238
     239
     240
     241  1616  20 AA 19
     242  1619  2C 9E 15
     243  161C  30 03
     244  161E  20 08 17
     245  1621  A9 00
     246  1623  2C 9F 15
     247  1626  8D 9F 15
     248  1629  30 1B
     249  162B  20 73 18
     250  162E  30 05
     251  1630  68
     252  1631  68
     253  1632  4C B8 17
     254
     255
     256
     257
     258  1635  AD 9D 15
     259  1638  D0 0A
     260  163A  A9 1B
     261  163C  A2 17
     262  163E  20 BE 19
     263  1641  4C 01 18
     264
     265
     266
     267  1644  A9 00
     268  1646  AA
     269  1647  60
     270
     271
     272
     273  1648  20 AA 19
     274  164B  A9 03
     275  164D  D0 F7
     276  164F  98
     277  1650  48
     278  1651  20 AA 19
```

```
RDDRC   LDX     #$10
        LDA     #.LOW.HDBUFL
        STA     ICBAL,X
        LDA     #.LOW.HDBUFH
        STA     ICBAH,X
        LDA     #4
RDDRC1  STA     ICBLL,X
        LDA     #0
        STA     ICBLH,X
        JSR     CIO             ;NO ERROR CHECK SO CAN CATCH EOF
        BPL     STOK            ;IF NO ERROR
        CPY     #$88            ;SEE IF EOF
        BNE     ERST            ;IF SOME ERROR STATUS
;
;EOF SO DONE, EXIT
;
        JSR     CLOSX           ;CLOSE IOCB'S 1 AND 2
        BIT     OPT
        BMI     DRUN            ;BRANCH IF NO RUN OPTION
        JSR     JMPRUN          ;JUMP THROUGH RUN VECTOR
DRUN    LDA     #0              ;OK STATUS
        BIT     LOADFG          ;WAS MEMORY SWAPPED?
        STA     LOADFG
        BMI     CLFX            ;BRANCH IF MEMORY WASN'T SWAPPED
        JSR     MEMSVQ          ;DOES MEMORY SAVE FILE EXIST?
        BMI     DRUN1           ;BRANCH IF NOT
        PLA
        PLA
        JMP     GOOD            ;WRITE MEMORY AND RELOAD DUP
;
;       SEE IF DUP WRITTEN OVER.  IF IS RELOAD & TELL USER NEED MEM.SAV T
;       LOAD THIS FILE.
;
DRUN1   LDA     DUPFLG          ;SEE IF DUP CLOBBERED
        BNE     DRUN2           ;NO, THEN RETURN
        LDA     #.LOW.NMSFL     ;ELSE TELL USER NEED MEM.SAV
        LDX     #.LOW.NMSFH
        JSR     PRNTMSG         ;PRINT MSG
        JMP     RRDUP           ;RELOAD & RUN DUP
;
;       RETURN TO CALLING ROUTINE
;
DRUN2   LDA     #0              ;NO DUP ERR MSG ON EOF
CLFX    TAX
RTS     RTS
;
;       ERROR RETURNS
;
LNLF    JSR     CLOSX
        LDA     #3              ;BAD LOAD FILE
        BNE     CLFX
ERST    TYA
        PHA
        JSR     CLOSX
```

```
 279  1654  68              PLA
 280  1655  A8              TAY
 281  1656  D0 EE           BNE     CLFX
 282                  ;
 283                  ;       CONTINUE WITH LOAD - CHECK LOAD ADDRESS FOR HEADER
 284                  ;       HEADER IF HAVE CONCATENATED LOAD FILES
 285                  ;
 286  1658  A2 10      STOK    LDX     #$10
 287  165A  AD A0 15          LDA     HDBUF           ;MOVE PARAMS TO IOCB
 288  165D  9D 44 03          STA     ICBAL,X
 289  1660  48                PHA
 290  1661  AD A1 15          LDA     HDBUF+1
 291  1664  9D 45 03          STA     ICBAH,X
 292  1667  A8                TAY
 293  1668  68                PLA
 294  1669  C8                INY                     ;WAS ADDRESS FF?
 295  166A  D0 1F             BNE     ADOK            ;BRANCH IF NOT
 296  166C  A8                TAY
 297  166D  C8                INY                     ;OTHER BYTE FF?
 298  166E  D0 1B             BNE     ADOK            ;BRANCH IF NOT
 299                  ;
 300                  ;       HAVE A HEADER & START ADDRESS - GET END ADDRESS FOR TEXT & DO AG
 301                  ;
 302  1670  AD A2 15          LDA     HDBUF+2
 303  1673  8D A0 15.         STA     HDBUF
 304  1676  AD A3 15          LDA     HDBUF+3
 305  1679  8D A1 15          STA     HDBUF+1         ;MOVE LOAD ADDRESS
 306  167C  A9 A2             LDA     #.LOW.HDBUF+2
 307  167E  9D 44 03          STA     ICBAL,X
 308  1681  A9 15             LDA     #.HIGH.(HDBUF+2)
 309  1683  9D 45 03          STA     ICBAH,X         ;SO LOAD ADDRESS DOESN'T GET WIPED OUT B
 310  1686  A9 02             LDA     #2
 311  1688  4C 05 16          JMP     RDDRC1
 312                  ;
 313                  ;       GET LENGTH OF TEXT.  THEN DETERMINE IF IN DUP
 314                  ;
 315  168B  AD A2 15  ADOK    LDA     HDBUF+2
 316  168E  38                SEC
 317  168F  ED A0 15          SBC     HDBUF
 318  1692  9D 48 03          STA     ICBLL,X
 319  1695  AD A3 15          LDA     HDBUF+3
 320  1698  ED A1 15          SBC     HDBUF+1
 321  169B  9D 49 03          STA     ICBLH,X
 322  169E  AD A1 15          LDA     HDBUF+1
 323  16A1  20 FA 16          JSR     AWDQ            ;IS BEGINNING ADDRESS WITHIN DUP?
 324  16A4  B0 15             BCS     AWD             ;BRANCH IF SO
 325  16A6  AD A3 15          LDA     HDBUF+3
 326  16A9  20 FA 16          JSR     AWDQ            ;IS ENDING ADDRESS WITHIN DUP?
 327  16AC  B0 0D             BCS     AWD             ;BRANCH IF SO
 328                  ;
 329                  ;       SINCE TEXT IN DUP, LOAD MEM SAV IF NECCESARY
 330                  ;
 331  16AE  AD 0B 17  ANWD    LDA     MEMLDD
 332  16B1  30 08             BMI     AWD             ;BRANCH IF MEM.SAV ALREADY LOADED
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    333  16B3  A9 80
    334  16B5  0D 9F 15
    335  16B8  8D 9F 15
    336  16BB  FE 48 03
    337  16BE  D0 03
    338  16C0  FE 49 03
    339  16C3  2C 9F 15
    340  16C6  30 13
    341  16C8  AD 0B 17
    342  16CB  30 0E
    343  16CD  CE 0B 17
    344  16D0  20 39 19
    345  16D3  A9 00
    346  16D5  8D 9D 15
    347  16D8  20 2E 19
    348
    349
    350
    351  16DB  A2 10
    352  16DD  A9 47
    353  16DF  8D E2 02
    354  16E2  A9 16
    355  16E4  8D E3 02
    356  16E7  20 56 E4
    357  16EA  10 03
    358  16EC  4C 4F 16
    359  16EF  2C 9E 15
    360  16F2  30 03
    361  16F4  20 05 17
    362  16F7  4C F7 15
    363
    364
    365
    366
    367
    368
    369  16FA  C9 1D
    370  16FC  90 06
    371  16FE  C9 34
    372  1700  2A
    373  1701  49 01
    374  1703  4A
    375  1704  60
    376
    377
    378  1705  6C E2 02
    379  1708  6C E0 02
    380
    381
    382  170B  00
    383  170C  44 31 3A 41
    384  1710  55 54 4F 52
    385  1714  55 4E 2E 53
    386  1718  59 53 9B
```

```
        LDA     #$80
        ORA     LOADFG
        STA     LOADFG          ;SET MEM.SAV DOESN'T HAVE TO BE LOADED F
AWD     INC     ICBLL,X
        BNE     *+5
        INC     ICBLH,X
        BIT     LOADFG          ;DOES MEMORY HAVE TO BE LOADED
        BMI     DLM             ;BRANCH IF NOT
        LDA     MEMLDD          ;WAS MEM.SAV ALREADY LOADED?
        BMI     DLM             ;BRANCH IF SO
        DEC     MEMLDD
        JSR     LDMEM           ;LOAD MEM.SAVE FILE (IF IT EXISTS)
        LDA     #0              ;SHOW USER AREA NOT DUP IN MEMORY
        STA     DUPFLG
        JSR     RELDIN          ;RESTORE DOS IN VECTOR FROM SAVED LOC
;
;       SET NO INIT ADDR DEFAULT THEN READ IN TEXT & ATTEMPT INIT
;
DLM     LDX     #$10
        LDA     #.LOW.RTS
        STA     INITAD
        LDA     #.HIGH.RTS
        STA     INITAD+1        ;INIT DEFAULTS TO AN RTS
        JSR     CIO             ;READ DATA DIRECTLY TO MEMORY
        BPL     DLM1
        JMP     ERST            ;IF ERRORS
DLM1    BIT     OPT
        BMI     DINIT           ;BRANCH IF NO GO OPTION
        JSR     JMPINT          ;DO INIT
DINIT   JMP     RDDRC           ;GET NEXT SECTION OF LOAD FILE
;
;       SUBROUTINE TO DETERMINE IF ADDRESS IS WITHIN DUP ADDRESS SPACE.
;       ENTRY - HI BYTE OF ADDRESS IN REG. A
;       RETURNS - CARRY SET : WITHIN DUP
;                 CARRY CLR : NOT WITHIN DUP
;
AWDQ    CMP     #.LOW.NDOSH
        BCC     AWDQR           ;BRANCH IF HI BYTE LT DUP START
        CMP     #.LOW.NMDUPH+1
        ROL     A
        EOR     #1
        LSR     A               ;COMPLEMENT CARRY
AWDQR   RTS
;
;
JMPINT  JMP     (INITAD)
JMPRUN  JMP     (RUNAD)
;
;
MEMLDD  .BYTE   0
AF      .BYTE   'D1:AUTORUN.SYS',CR
```

```
      387  171B                        HILO    AF
      388  0017                +AFH    =       AF/256
      389  000C                +AFL    =       (-256)*AFH+AF
      390  171B  4E 45 45 44    NMSF    .BYTE   'NEED MEM.SAV TO LOAD THIS FILE.',CR
      391  171F  20 4D 45 4D
      392  1723  2E 53 41 56
      393  1727  20 54 4F 20
      394  172B  4C 4F 41 44
      395  172F  20 54 48 49
      396  1733  53 20 46 49
      397  1737  4C 45 2E 9B
      398  173B                        HILO    NMSF
      399  0017                +NMSFH  =       NMSF/256
      400  001B                +NMSFL  =       (-256)*NMSFH+NMSF
```

```
ERR LINE  ADDR  B1 B2 B3 B4

 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414  173B  44 31 3A 4D
 415  173F  45 4D 2E 53
 416  1743  41 56 9B
 417  1746
 418  0017
 419  003B
 420  1746  20 AA 19
 421  1749  A9 08
 422  174B  9D 4A 03
 423  174E  20 79 17
 424  1751  30 38
 425
 426
 427
 428
 429  1753  A9 0B
 430  1755  9D 42 03
 431  1758  A9 7C
 432  175A  9D 44 03
 433  175D  A9 1D
 434  175F  9D 45 03
 435  1762  A9 8A
 436  1764  9D 48 03
 437  1767  A9 15
 438  1769  9D 49 03
 439  176C  20 56 E4
 440  176F  30 1A
 441  1771  20 AA 19
 442  1774  30 15
 443  1776  A0 00
 444  1778  60
 445
 446  1779  A9 03
 447  177B  9D 42 03
 448  177E  A9 3B
 449  1780  9D 44 03
 450  1783  A9 17
 451  1785  9D 45 03
 452  1788  4C 56 E4
 453
 454  178B  8C 9A 17
```

```
        .PAGE
;   ****  CREATE  MEM.SAV  FILE  ****
;
;
;ROUTINE WRITTEN BY M.E., APRIL 21,1980
;THIS ROUTINE CREATES A FILE ON DISK OF DATA FROM MEMORY
;CREATE FILE CALLED 'D1:MEM.SAV', SET Y=1
;
;ABLE TO CREATE FILE THEN SET REG.Y=ERROR RETURNED FROM CIO
;THE RAM TO BE OCCUPIED BY DUP IS STORED BY THIS ROUTINE INTO
;'MEMORY.SAV'
;
;
NAME    .BYTE   'D1:MEM.SAV',CR


        HILO    NAME
+NAMEH    =      NAME/256
+NAMEL    =      (-256)*NAMEH+NAME
MWRITE  JSR     CLOSX           ;CLOSE IOCB AND OPEN IT TO WRITE
        LDA     #OWRIT          ;
        STA     ICAX1,X         ;
        JSR     OREST           ;OPEN FOR WRITE
        BMI     ERRWR           ;IF ERROR THEN JMP AND RET
;
;
;WRITE MEMORY BLOCK
;
        LDA     #PUTCHR
        STA     ICCOM,X
        LDA     #.LOW.NDOSL     ;STORE START OF BLOCK FOR CIO
        STA     ICBAL,X
        LDA     #.LOW.NDOSH     ;START ADDR (HIGH)
        STA     ICBAH,X
        LDA     #.LOW.MLENL+1   ;LENGTH OF BLOCK
        STA     ICBLL,X
        LDA     #.LOW.MLENH     ;LENGTH(HIGH)
        STA     ICBLH,X
        JSR     CIO             ;WRITE DATA BLOCK
        BMI     ERRWR           ;IF WRITE ERROR THEN JMP
        JSR     CLOSX
        BMI     ERRWR
        LDY     #0
RET     RTS
;
OREST   LDA     #.LOW.OPEN
        STA     ICCOM,X
        LDA     #.LOW.NAMEL     ;ROUTINE TO COMPLETE OPEN OF 'D1:MEMORY
        STA     ICBAL,X         ;CALLING SUB SUPPLIES 'READ' OR 'WRITE'
        LDA     #.LOW.NAMEH     ;IN ICAX1
        STA     ICBAH,X
        JMP     CIO
;
ERRWR   STY     TEMP+1          ;TEMP STORE FOR Y FLAG
```

```
455  178E  20 AA 19          JSR     CLOSX           ;CLOSE #$20
456  1791  A9 21             LDA     #.LOW.DELETE    ;DELETE PART OF MENSAV
457  1793  9D 42 03          STA     ICCOM,X
458  1796  20 79 17          JSR     OREST
459  1799  A0 00       TEMP   LDY     #0              ;RESTORE FLAG
460  179B  60               RTS                     ;RETURN TO MAIN CALLER
```

```
461                             .PAGE
462                     ;  **** ENTRY POINT ON 'DOS' CALL ****
463                     ;
464                     ;
465  179C  00 00        INISAV  .DBYTE  0               ;DOSINI VECTOR SAVE LOC
466  179E  00            MEMFLG  .BYTE   0
467  179F  A2 00         MNDUP   LDX     #0
468  17A1  8E 9E 17              STX     MEMFLG
469  17A4  8E 9F 15              STX     LOADFG
470  17A7  CA                    DEX
471  17A8  86 08                 STX     WARMST
472  17AA  20 76 19              JSR     INITIO
473                     ;
474  17AD  20 73 18              JSR     MEMSVQ          ;FIND OUT IF FILE D1:MEM.SAV EXISTS
475  17B0  10 06                 BPL     GOOD            ;BRANCH IF MEM.SAV FILE EXITS
476  17B2  A9 00                 LDA     #0
477  17B4  85 08                 STA     WARMST          ;CLEAR WARM START FLAG
478  17B6  F0 3F                 BEQ     FINAL
479                     ;
480                     ;
481  17B8  20 46 17      GOOD    JSR     MWRITE          ;WRITE USER AREA TO MEM.SAV
482  17BB  30 05                 BMI     ERROR
483  17BD  CE 9E 17              DEC     MEMFLG          ;SHOW MEMORY WRITTEN
484  17C0  30 35                 BMI     FINAL
485                     ;
486  17C2  A9 3A         ERROR   LDA     #.LOW.ERRMES    ;PRINT ERROR OCCURED MSG
487  17C4  A2 18                 LDX     #.HIGH.ERRMES
488  17C6  20 BE 19              JSR     PRNTMSG         ;GOTO MSG PRINTER
489                     ;
490  17C9  A9 5B                 LDA     #.LOW.ERR       ;PRINT QUERY TO RUN DOS
491  17CB  A2 18                 LDX     #.HIGH.ERR
492  17CD  20 BE 19              JSR     PRNTMSG         ;GOTO MSG PRINTER
493                     ;
494                     ;       WAIT FOR Y TO RUN DOS
495                     ;
496  17D0  A9 05                 LDA     #GETREC
497  17D2  8D 42 03              STA     ICCOM
498  17D5  A9 00                 LDA     #.LOW.STAKL
499  17D7  8D 44 03              STA     ICBAL
500  17DA  A9 01                 LDA     #.LOW.STAKH
501  17DC  8D 45 03              STA     ICBAH
502  17DF  A9 02                 LDA     #2
503  17E1  8D 48 03              STA     ICBLL
504  17E4  A9 00                 LDA     #0
505  17E6  8D 49 03              STA     ICBLH
506  17E9  20 56 E4              JSR     CIO
507  17EC  AD 00 01              LDA     STAK            ;SEE IF Y TYPED
508  17EF  C9 59                 CMP     #'Y
509  17F1  D0 38                 BNE     RTCART          ;BRANCH IF NOT
510  17F3  A9 00                 LDA     #0
511  17F5  85 08                 STA     WARMST
512                     ;
513  17F7  A2 20         FINAL   LDX     #$20
514  17F9  A9 0C                 LDA     #CLOSE
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    515  17FB  9D 42 03
    516  17FE  20 56 E4
    517
    518  1801  A5 0C
    519  1803  8D 9C 17
    520  1806  A5 0D
    521  1808  8D 9D 17
    522
    523  180B  A9 40
    524  180D  85 0C
    525  180F  A9 15
    526  1811  85 0D
    527
    528  1813  A9 2F
    529  1815  A2 10
    530  1817  9D 44 03
    531  181A  A9 18
    532  181C  9D 45 03
    533  181F  A0 00
    534  1821  8C 9E 15
    535  1824  88
    536  1825  8C 9D 15
    537  1828  20 A4 15
    538  182B  60
    539  182C  45 3A 9B
    540  182F
    541  0018
    542  002C
    543  182F
    544  0017
    545  009F
    546  182F  44 31 3A 44
    547  1833  55 50 2E 53
    548  1837  59 53 9B
    549
    550  183A  45 52 52 4F
    551  183E  52 2D 53 41
    552  1842  56 49 4E 47
    553  1846  20 55 53 45
    554  184A  52 20 4D 45
    555  184E  4D 4F 52 59
    556  1852  20 4F 4E 20
    557  1856  44 49 53 4B
    558  185A  9B
    559  185B  54 59 50 45
    560  185F  20 59 20 54
    561  1863  4F 20 53 54
    562  1867  49 4C 4C 20
    563  186B  52 55 4E 20
    564  186F  44 4F 53 9B
```

```
        STA     ICCOM,X         ;SET UP CLOSE COMMAND
        JSR     CIO             ;PERFORM CLOSE COMMAND
;
RRDUP   LDA     DOSINI          ;SAVE DOS INIT VECTOR
        STA     INISAV
        LDA     DOSINI+1
        STA     INISAV+1
;
        LDA     #.LOW.DOS       ;SET UP DUP INIT ADDR AS
        STA     DOSINI          ;DOS INIT VECTOR
        LDA     #.HIGH.DOS
        STA     DOSINI+1
;
RRDUP1  LDA     #.LOW.DUPSYS
        LDX     #$10
        STA     ICBAL,X
        LDA     #.HIGH.DUPSYS
        STA     ICBAH,X
        LDY     #0
        STY     OPT             ;ASSURE NO /N OPTION IN EFFECT
        DEY                     ;SHOW THAT DUP IS IN MEMORY
        STY     DUPFLG
        JSR     SFLOAD          ;LOAD DUP.SYS AND RUN IT
RTCART  RTS
EC      .BYTE   'E:',CR
        HILO    EC
+ECH    =       EC/256
+ECL    =       (-256)*ECH+EC
        HILO    MNDUP
+MNDUPH =       MNDUP/256
+MNDUPL =       (-256)*MNDUPH+MNDUP
DUPSYS  .BYTE   'D1:DUP.SYS',CR

;
ERRMES  .BYTE   'ERROR-SAVING USER MEMORY ON DISK',CR


ERR     .BYTE   'TYPE Y TO STILL RUN DOS',CR
```

```
 565
 566                        ;  ****  SUBROUTINES  FOR  RESIDENT  DUP  ****
 567                        ;
 568                        ;
 569                        ;       ROUTINE TESTS IF MEM.SAV IS PRESENT ON THE DISK.
 570                        ;       RETURNS - MINUS IF NOT THERE
 571                        ;                 PLUS  IF MEM.SAV IS THERE
 572                        ;
 573  1873  20 B4 19        MEMSVQ  JSR     CLOS20          ;CLOSE IOCB # 2
 574  1876  A9 03                   LDA     #OPEN
 575  1878  9D 42 03                STA     ICCOM,X
 576  187B  A9 3B                   LDA     #.LOW.NAMEL
 577  187D  9D 44 03                STA     ICBAL,X
 578  1880  A9 17                   LDA     #.LOW.NAMEH
 579  1882  9D 45 03                STA     ICBAH,X
 580  1885  A9 0C                   LDA     #ORDWRT
 581  1887  9D 4A 03                STA     ICAX1,X         ;TRY TO OPEN D1:MEM.SAV FOR READ/WRITE
 582  188A  20 56 E4                JSR     CIO
 583  188D  08                      PHP                     ;SAVE STATUS
 584  188E  20 B4 19                JSR     CLOS20          ;CLOSE MEM.SAV
 585  1891  28                      PLP                     ;RESTORE STATUS
 586  1892  60                      RTS
 587  1893
 588                        ;
 589                        ;
 590                        ;       SAVE FILE SUBROUTINE - WRITE FILE BODY, INIT, & RUN VECTORS
 591                        ;
 592  1893  A9 00           WDR1    LDA     #0              ;THIS IMMEDIATE VALUE MODIFIED
 593  1895  F0 03                   BEQ     WDR2            ;BR IF MEM FILE DOESNT HAVE TO BE LOADED
 594  1897  20 39 19                JSR     LDMEM
 595  189A  A2 10           WDR2    LDX     #$10
 596  189C  20 56 E4                JSR     CIO             ;DO SAVE - WRITE BODY TO DISK
 597  189F  A9 00           INITQ   LDA     #0              ;THIS IMMED VALUE CHANGED DURING SAVE
 598  18A1  F0 1A                   BEQ     RUNQ            ;SET TO FF WHEN AN INIT VECTOR IS PRESNT
 599  18A3  EE A0 18                INC     INITQ+1
 600  18A6  AD E2 02                LDA     INITAD
 601  18A9  8D E4 19                STA     VECTR           ;IF INIT VECTOR FOR FILE SAVE IT
 602  18AC  AD E3 02                LDA     INITAD+1
 603  18AF  8D E5 19                STA     VECTR+1
 604  18B2  A9 E2                   LDA     #.LOW.INITAD
 605  18B4  AA                      TAX
 606  18B5  8D E0 19                STA     LDST
 607  18B8  A9 02                   LDA     #.HIGH.INITAD
 608  18BA  20 EF 18                JSR     WRVEC
 609  18BD  A9 00           RUNQ    LDA     #0              ;THIS IMMEDIATE VALUE MODIFIED
 610  18BF  F0 1A                   BEQ     NORNAD          ;SET TO FF WHEN A RUN VECTOR IS PRESENT
 611  18C1  EE BE 18                INC     RUNQ+1
 612  18C4  AD E0 02                LDA     RUNAD
 613  18C7  8D E4 19                STA     VECTR           ;IF RUN VECTOR FOR FILE SAVE IT
 614  18CA  AD E1 02                LDA     RUNAD+1
 615  18CD  8D E5 19                STA     VECTR+1
 616  18D0  A9 E0                   LDA     #.LOW.RUNAD
 617  18D2  AA                      TAX
 618  18D3  8D E0 19                STA     LDST
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    619  18D6  A9 02
    620  18D8  20 EF 18
    621  18DB  20 AA 19
    622  18DE  AD 9E 17
    623  18E1  2D 94 18
    624  18E4  F0 06
    625  18E6  EE 94 18
    626  18E9  4C 13 18
    627  18EC  4C 75 20
    628
    629
    630
    631  18EF  8D E1 19
    632  18F2  E8
    633  18F3  8E E2 19
    634  18F6  8D E3 19
    635  18F9  A2 10
    636  18FB  A9 E0
    637  18FD  9D 44 03
    638  1900  A9 19
    639  1902  9D 45 03
    640  1905  A9 06
    641  1907  9D 48 03
    642  190A  A9 00
    643  190C  9D 49 03
    644  190F  4C 56 E4
    645
    646
    647
    648
    649  1912  20 39 19
    650  1915  A9 00
    651  1917  8D 9D 15
    652  191A  20 2E 19
    653  191D  6C FA BF
    654
    655
    656
    657
    658  1920  20 39 19
    659  1923  A9 00
    660  1925  8D 9D 15
    661  1928  20 2E 19
    662  192B  6C 1A 00
    663
    664
    665
    666  192E  AD 9C 17
    667  1931  85 0C
    668  1933  AD 9D 17
    669  1936  85 0D
    670  1938  60
    671
    672
```

```
          LDA     #.HIGH.RUNAD
          JSR     WRVEC
NORNAD    JSR     CLOSX           ;CLOSE IOCBS 1 &2
          LDA     MEMFLG
          AND     WDR1+1
          BEQ     DRRDUP
          INC     WDR1+1          ;RESET MEM.NEEDS TO BE LOADED FLAG
          JMP     RRDUP1          ;RELOAD & RUN DUP
DRRDUP    JMP     DOSOS           ;RUN THE SWAPPED IN DUP
;
;
;
WRVEC     STA     LDST+1
          INX
          STX     LDND
          STA     LDND+1
          LDX     #$10
          LDA     #.LOW.LDST
          STA     ICBAL,X
          LDA     #.HIGH.LDST
          STA     ICBAH,X
          LDA     #6
          STA     ICBLL,X
          LDA     #0
          STA     ICBLH,X
          JMP     CIO             ;WRITE INIT OR RUN ADDRESS
;
;
;         JUMP TO CARTRIDGE
;
CLMJMP    JSR     LDMEM
          LDA     #0              ;SHOW DUP NO LONGER IN MEMORY
          STA     DUPFLG
          JSR     RELDIN          ;RESTORE DOS INIT VECTOR SAVED
          JMP     (CARTST)        ;JUMP TO CARTRIDGE
;
;
;         LOAD MEM.SAV (IF IT EXISTS) BEFORE RUN AT ADDRESS
;
LMTR      JSR     LDMEM           ;LOAD MEM.SAVE IF IT EXISTS
          LDA     #0              ;SHOW THAT DUP NO LONGER IN MEMORY
          STA     DUPFLG
          JSR     RELDIN          ;RESTORE DOS INIT VECTOR SAVED
          JMP     (RAMLO)         ;RUN AT ADDRESS
;
;         RESTORE DOSINI VECTOR FROM SAVED LOCATION
;
RELDIN    LDA     INISAV
          STA     DOSINI
          LDA     INISAV+1
          STA     DOSINI+1
          RTS
;
;
```

```
ERR LINE  ADDR  B1 B2 B3 B4

     673
     674
     675
     676  1939  AD 9E 17
     677  193C  D0 01
     678  193E  60
     679  193F  20 73 18
     680  1942  10 06
     681  1944  A9 00
     682  1946  85 08
     683  1948  F0 24
     684
     685  194A  A9 03
     686  194C  9D 42 03
     687  194F  20 56 E4
     688  1952  A9 07
     689  1954  9D 42 03
     690  1957  A9 8A
     691  1959  9D 48 03
     692  195C  A9 15
     693  195E  9D 49 03
     694  1961  A9 7C
     695  1963  9D 44 03
     696  1966  A9 1D
     697  1968  9D 45 03
     698  196B  20 56 E4
     699  196E  A9 0C
     700  1970  9D 42 03
     701  1973  4C 56 E4
     702
     703
     704
     705  1976  20 6E E4
     706
     707  1979  A2 00
     708  197B  A9 03
     709  197D  9D 42 03
     710  1980  A9 2C
     711  1982  9D 44 03
     712  1985  A9 18
     713  1987  9D 45 03
     714  198A  A9 0C
     715  198C  9D 4A 03
     716  198F  20 56 E4
     717
     718  1992  A2 00
     719  1994  8E 1C 02
     720  1997  8E 1D 02
     721  199A  A0 01
     722  199C  A9 03
     723  199E  8D 2A 02
     724  19A1  20 5C E4
     725  19A4  AD 2A 02
     726  19A7  D0 FB
```

```
;       SUBROUTINE - LDMEM
;       LOAD MEM.SAV IF IT EXISTS
;
LDMEM   LDA     MEMFLG
        BNE     LDMEM1          ;BRANCH IF MEMORY WAS SAVED
        RTS
LDMEM1  JSR     MEMSVQ
        BPL     LDMEM2          ;BRANCH IF MEM.SAV FILE DOES EXIST
        LDA     #0              ;TELL CART PGM AREA CLOBBERED
        STA     WARMST
        BEQ     CLOS2           ;GO CLOSE AND GOTO CART
;
LDMEM2  LDA     #OPEN
        STA     ICCOM,X
        JSR     CIO             ;REOPEN MEM.SAV
        LDA     #GETCHR
        STA     ICCOM,X
        LDA     #.LOW.MLENL+1
        STA     ICBLL,X
        LDA     #.LOW.MLENH
        STA     ICBLH,X
        LDA     #.LOW.NDOSL
        STA     ICBAL,X
        LDA     #.LOW.NDOSH
        STA     ICBAH,X
        JSR     CIO
CLOS2   LDA     #CLOSE
        STA     ICCOM,X
        JMP     CIO             ;CLOSE MEM.SAV
;
;       CLOSE ALL IOCBS & RE-OPEN ZERO AS SCREEN EDITOR
;
INITIO  JSR     CIOINV          ;THIS ROUTINE CLOSES ALL IOCB'S
                                ;THEN REOPENS THE SCREEN EDITOR
        LDX     #0
        LDA     #OPEN
        STA     ICCOM,X
        LDA     #.LOW.ECL
        STA     ICBAL,X
        LDA     #.LOW.ECH
        STA     ICBAH,X
        LDA     #ORDWRT
        STA     ICAX1,X
        JSR     CIO
;
        LDX     #0              ;DELAY UNTIL DMA (SCREEN) IS RESTORED
        STX     CDTMV3          ;CLEAR TIMER NUMBER 3
        STX     CDTMV3+1
        LDY     #1              ;WAIT FOR ONE VBLANK
        LDA     #3              ;USE TIMER # 3
        STA     CDTMF3          ;SET TIMER DONE FLAG TO NOT DONE
        JSR     SETVBV          ;SYSTEM CALL TO SET TIMER
WAITIM  LDA     CDTMF3          ;WAIT UNTIL TIMER IS DONE
        BNE     WAITIM
```

```
                                ;
                                        RTS
                                ;
                                ;  CLOSX - CLOSE IOCBS 10,20
                                ;
19A9  60                        CLOSX   LDA     #CLOSE
                                        LDX     #$10
19AA  A9 0C                             STA     ICCOM,X
19AC  A2 10                             JSR     CIO
19AE  9D 42 03
19B1  20 56 E4                  ;
                                ;       ENTRY TO CLOSE IOCB # 2 ONLY
                                ;
                                CLOS20  LDX     #$20
19B4  A2 20                             LDA     #CLOSE
19B6  A9 0C                             STA     ICCOM,X
19B8  9D 42 03                          JMP     CIO
19BB  4C 56 E4
                                ;
                                ;       SUBROUTINE - PRNTMSG
                                ;       PUTS A CHARACTER STRING TERMINATED BY A CARRIAGE RETURN CHAR TO
                                ;       SCREEN EDITOR.
                                ;
                                ;       ENTRY - REG A : LOW BYTE MSG ADDRESS
                                ;               REG X : HI BYTE MSG ADDRESS
                                ;
                                ;       PUT PARAMS IN IOCB - USE IOCB 0 FOR SCREEN EDITOR
                                ;
19BE  8D 44 03                  PRNTMSG STA     ICBAL           ;SET MSG ADDR IN IOCB BUFF ADDR
19C1  8E 45 03                          STX     ICBAH
                                ;
                                ;       SET UP REST OF IOCB
                                ;
19C4  A9 80                             LDA     #$80            ;SET IN BUFFER LENGTH
19C6  8D 48 03                          STA     ICBLL           ;ASSUME 128 BYTES MAX
19C9  A2 00                             LDX     #0              ;USE REG X TO SET IN IOCB INDEX FOR CIO
19CB  8E 49 03                          STX     ICBLH
19CE  A9 09                             LDA     #PUTREC         ;PUT MSG
19D0  8D 42 03                          STA     ICCOM
                                ;
                                ;       TEST IF DUP IS RESIDENT - IF IS THEN USE INDIRECT CIO ROUTINE
                                ;       TO TEST FOR BREAK KEY ABORT
                                ;
19D3  AD 9D 15                          LDA     DUPFLG          ;=ZERO IF NON-RESIDENT DUP NOT IN MEM
19D6  D0 03                             BNE     INMEM           ;IN MEMORY THEN USE INDIRECT CIO CALL
                                ;
19D8  4C 56 E4                          JMP     CIO             ;ELSE GO DIRECT TO CIO & RETURN
                                ;
19DB  4C AA 31                  INMEM   JMP     CIO1            ;USE CIO CALL WITH TEST FOR BREAK KEY AB
                                ;
                                ;
19DE  FF FF                     SAVH    .BYTE   $FF,$FF
19E0                                    HILO    SAVH
0019                            +SAVHH    =      SAVH/256
00DE                            +SAVHL    =      (-256)*SAVHH+SAVH
19E0                            LDST:   .RES    2
```

```
      781  19E2                        HILO    LDST
      782  0019              +LDSTH      =       LDST/256
      783  00E0              +LDSTL      =       (-256)*LDSTH+LDST
      784  19E2               LDND:    .RES    2
      785  19E4               VECTR:   .RES    2
```

```
786                     .PAGE
787                ;  ****  SIO  INTERRUPT  SERVICE  ROUTINES  ****
788                ;
789                ;
790                ;       EQUATES FOR INTERRUPT ROUTINES MOVED FROM SIO
791                ;
792                ;       ZERO PAGE
793                ;
794  0032          BUFRLO  =       $32             ;POINTER TO BYTE TO SEND OR RECEIVE
795  0033          BUFRHI  =       $33
796  0034          BFENLO  =       $34             ;POINTER TO BYTE AFTER END OF BUFFER
797  0035          BFENHI  =       $35
798  0031          CHKSUM  =       $31             ;LOC TO STORE DATA FRAME CHECKSUM
799  003B          CHKSNT  =       $3B             ;CHECKSUM SENT FLAG- =FF SENT
800  003C          NOCKSM  =       $3C             ;FLAG NO CHECK SUM TO BE RECEIVED-NOT ZE
801  0030          STATUS  =       $30             ;HOLD FOR STATUS TO BE PUT IN DCB
802  0038          BUFRFL  =       $38             ;FLAG-IF FF RECEIVE BUFFER IS FULL
803  0039          RECVDN  =       $39             ;FLAG RECEIVE NOT DONE. USED BY WAIT LOO
804  0010          POKMSK  =       $10             ;POKEY INTERRUPT MASK SHADOW FOR IRQEN
805                ;
806                ;       HARDWARE REGISTERS USED IN SIO INTERRUPT ROUTINES
807                ;
808  D20A          SKRES   =       $D20A           ;SERIAL PORT STATUS RESET ON POKEY
809  D20D          SEROUT  =       $D20D           ;SERIAL OUTPUT REGISTER
810  D20D          SERIN   =       SEROUT          ;SERIAL PORT INPUT REG ON POKEY
811  D20E          IRQEN   =       $D20E           ;IRQ INTERRUPT ENABLE ON POKEY
812  D20F          SKSTAT  =       $D20F           ;SERIAL PORT STATUS REG ON POKEY
813                ;
814                ;       ERROR CODES RETURNED BY SIO
815                ;
816  008C          FRMERR  =       $8C             ;FRAMING ERROR ON INPUT
817  008E          OVRRUN  =       $8E             ;DATA FRAME OVER RUN-BIT D5 IN SKSTAT
818  008F          CHKERR  =       $8F             ;DATA FRAME CHECKSUM ERROR
```

```
819                             .PAGE
820                     ;  **** INTERRUPT SERVICE ROUTINE TO OUTPUT DATA NEEDED ****
821                     ;
822                     ;
823                     ;
824                     ;       IT UPDATES THE BYTE TO PUT ON SERIAL I/O BUS POINTER
825                     ;       UNTIL END OF BUFFER.  AFTER EACH UPDATE OF THE PTR ADDS THE
826                     ;       VALUE OF THE BYTE TO THE CHECKSUM.  OUTPUTS THE CHECKSUM WHEN
827                     ;       PTR EQUALS THE END OF BUFFER PTR (POINTS TO BYTE AFTER BUFFER).
828                     ;       RETURNS TO THIS ROUTINE AFTER CHECKSUM PASSED AND RESETS POKEY
829                     ;       INTERRUPT REG TO HAVE THE TRANSMIT DONE ROUTINE CALLED TO END
830                     ;       WAIT LOOP (SEE SIO LISTING).
831                     ;
832                     ;       K.B. 6/10/80
833                     ;
834  19E6  98          ISRODN  TYA                     ;SAVE Y REG ON STACK
835  19E7  48                  PHA
836                     ;
837  19E8  E6 32               INC     BUFRLO
838  19EA  D0 02               BNE     NOWRPO          ;INCREMENT PTR TO NEXT BYTE
839  19EC  E6 33               INC     BUFRHI          ;TO SEND
840                     ;
841                     ;       PATCH TO ROUTINE - CHANGED CHECK
842                     ;
843  19EE  A5 32       NOWRPO  LDA     BUFRLO          ;CHECK IF PTR IS WITHIN BUFFER
844  19F0  C5 34               CMP     BFENLO          ;DO A DOUBLE PRECISION SUBTRACT
845  19F2  A5 33               LDA     BUFRHI
846  19F4  E5 35               SBC     BFENHI
847  19F6  90 1A               BCC     NOTEND          ;BRANCH IF (BUFR) < (BFEN)-MORE TO SEND
848                     ;
849  19F8  A5 3B               LDA     CHKSNT          ;TEST IF CHECKSUM ALREADY SENT
850  19FA  D0 09               BNE     RELONE          ;BRANCH IF ALREADY SENT
851                     ;
852                     ;       SEND CHECKSUM AND SET FLAG
853                     ;
854  19FC  A5 31               LDA     CHKSUM
855  19FE  8D 0D D2            STA     SEROUT          ;PUT CHECKSUM IN SERIAL OUT REG
856  1A01  C6 3B               DEC     CHKSNT          ;SET FLAG TO FF HEX
857  1A03  D0 09               BNE     CHKDON          ;RETURN
858                     ;
859                     ;       AFTER CHECKSUM SENT AND CAUSE NEXT INTERRUPT THEN CHANGE POKEY
860                     ;       MASK TO ENABLE TRANSMIT DONE INTERRUPT AND TERMINATE WAIT LOOP.
861                     ;
862  1A05  A5 10       RELONE  LDA     POKMSK          ;GET POKEY MASK
863  1A07  09 08               ORA     #$08            ;OR IN ENABLE
864  1A09  85 10               STA     POKMSK
865  1A0B  8D 0E D2            STA     IRQEN           ;ENABLE THE INTERRUPTS
866                     ;
867                     ;       RESTORE REGS AND RETURN
868                     ;
869  1A0E  68          CHKDON  PLA
870  1A0F  A8                  TAY                     ;RESTORE Y REG
871  1A10  68                  PLA                     ;RESTORE A REG SAVED IN OS IRQ INTERRUPT
872  1A11  40                  RTI
```

```
 873                            ;
 874                            ;       MORE TO SEND.  SEND NEXT BYTE POINTED AT BY BUFR.
 875                            ;
 876  1A12  A0 00               NOTEND  LDY     #0
 877  1A14  B1 32                       LDA     (BUFRLO),Y      ;GET NEXT BYTE
 878  1A16  8D 0D D2                    STA     SEROUT          ;PUT IN SERIAL OUT REG
 879                            ;
 880  1A19  18                          CLC
 881  1A1A  65 31                       ADC     CHKSUM          ;ADD BYTE TO CHECKSUM
 882  1A1C  69 00                       ADC     #0
 883  1A1E  85 31                       STA     CHKSUM
 884                            ;
 885  1A20  4C 0E 1A                    JMP     CHKDON          ;GO RETURN AND WAIT FOR NEXT BYTE
 886                            ;
 887                            ;        ******** END OF OUT SERVICE ROUTINE ********
```

```
                                .PAGE
                        ;  **** SERIAL INPUT READY INTERRUPT SERVICE ROUTINE ****
                        ;
                        ;
                        ;       AFTER SERIAL RECEIVE IS ENABLED ROUTINE IS USED TO COLLECT
                        ;       BYTES FROM THE SERIAL INPUT REG AND PUT THEM IN BUFFER.
                        ;       WILL STOP WHEN BUFFER IS FULL.  IF A CHECKSUM IS EXPECTED
                        ;       ROUTINE WILL MARK BUFFER FULL AND CONTINUE.  WHEN CHECKSUM
                        ;       RECEIVED IT WILL CHECK IF = TO CHECKSUM IT WAS MAKING.
                        ;       WILL STORE ERRORS FOUND IN STATUS LOCATION.
                        ;
                        ;       THE IRQ INTERRUPT HANDLER IN THE OS PUSHES THE USER'S A REGISTER
                        ;       ONTO THE STACK BEFORE CALLING THIS ROUTINE.
                        ;
                        ;       K.B.    6/11/80
                        ;
1A23  98        ISRSIR  TYA                     ;SAVE Y REG ON STACK
1A24  48                PHA
                        ;
                        ;       GET STATUS FROM POKEY THEN RESET IT.
                        ;
1A25  AD 0F D2          LDA     SKSTAT
1A28  8D 0A D2          STA     SKRES           ;IGNORES VALUE- JUST STROBED
                        ;
                        ;       CHECK FOR ERRORS
                        ;
1A2B  30 04             BMI     NTFRAM          ;BIT 8 SET IF NO FRAMING ERROR
1A2D  A0 8C             LDY     #FRMERR
1A2F  84 30             STY     STATUS          ;SET FRAME ERROR STATUS
                        ;
1A31  29 20     NTFRAM  AND     #$20            ;IF BIT 5 CLEAR THEN FRAME OVER RUN
1A33  D0 04             BNE     NTOVRN          ;BRANCH IF NO OVER RUN
1A35  A0 8E             LDY     #OVRRUN
1A37  84 30             STY     STATUS          ;ELSE SET OVERRUN ERROR STATUS
                        ;
                        ;       CHECK IF BUFFER FULL AND THIS IS A CHECKSUM.  IF IT IS, THEN
                        ;       CHECK IF DATA SENT WAS VALID.
                        ;
1A39  A5 38     NTOVRN  LDA     BUFRFL          ;TEST FOR BUFFER FULL (NOT ZERO)
1A3B  F0 13             BEQ     NOTYET          ;IF ZERO THEN NOT YET, THIS IS DATA.
1A3D  AD 0D D2          LDA     SERIN           ;ELSE THIS IS CHECKSUM
1A40  C5 31             CMP     CHKSUM          ;ARE THEY EQUAL?
1A42  F0 04             BEQ     SRETRN          ;YES, THEN RETURN
1A44  A0 8F             LDY     #CHKERR         ;ELSE SET CHECK SUM ERROR STATUS
1A46  84 30             STY     STATUS
                        ;
                        ;       SET RECEIVE DONE TO END WAIT LOOP
                        ;
1A48  A9 FF     SRETRN  LDA     #$FF            ;DONE VALUE
1A4A  85 39             STA     RECVDN
                        ;
                        ;       RESTORE REGS AND RETURN
                        ;
```

```
942  1A4C  68          SUSUAL  PLA
943  1A4D  A8                  TAY                     ;RESTORE Y REG
944  1A4E  68                  PLA                     ;RESTORE A REG
945  1A4F  40                  RTI
946                    ;
947                    ;        IF BYTE IS DATA, THEN GET HERE.  PUT BYTE IN BUFFER AND CHECK IF
948                    ;        AT END OF BUFFER.
949                    ;
950  1A50  AD 0D D2    NOTYET  LDA     SERIN           ;GET DATA BYTE
951  1A53  A0 00               LDY     #0
952  1A55  91 32               STA     (BUFRLO),Y      ;STORE IT IN THE BUFFER
953                    ;
954  1A57  18                  CLC
955  1A58  65 31               ADC     CHKSUM          ;ADD DATA BYTE TO CHECKSUM
956  1A5A  69 00               ADC     #0
957  1A5C  85 31               STA     CHKSUM
958                    ;
959  1A5E  E6 32               INC     BUFRLO          ;INCREMENT POINTER TO LOCATION
960  1A60  D0 02               BNE     NTWRP1          ;FOR NEXT BYTE INPUT
961  1A62  E6 33               INC     BUFRHI
962                    ;
963                    ;        THE PATCH  CHANGED THE TEST FOR END OF BUFFER
964                    ;
965  1A64  A5 32       NTWRP1  LDA     BUFRLO          ;DO DOUBLE PRECISION SUBTRACT
966  1A66  C5 34               CMP     BFENLO
967  1A68  A5 33               LDA     BUFRHI
968  1A6A  E5 35               SBC     BFENHI          ;CARRY CLEAR IF BORROW
969  1A6C  90 DE               BCC     SUSUAL          ;BRANCH IF (BUFR) < (BFEN)-WITHIN BUFFER
970                    ;
971                    ;        DONE WITH DATA.  SEE IF CHECKSUM TO BE SENT
972                    ;
973  1A6E  A5 3C               LDA     NOCKSM          ;IF = ZERO THEN A CHECKSUM
974  1A70  F0 06               BEQ     GOON            ;WILL FOLLOW THE DATA
975                    ;
976  1A72  A9 00               LDA     #0              ;ELSE NO CHECKSUM TO FOLLOW
977  1A74  85 3C               STA     NOCKSM          ;CLEAR NO CHECKSUM FLAG
978  1A76  F0 D0               BEQ     SRETRN          ;RETURN AFTER SET RECEIVE DONE FLAG
979                    ;
980                    ;        SET BUFFER FULL AND THEN GO GET CHECKSUM
981                    ;
982  1A78  C6 38       GOON    DEC     BUFRFL          ;SET BUFFER FULL FLAG TO FF
983  1A7A  D0 D0               BNE     SUSUAL          ;GO RETURN
984                    ;
985                    ;        ******** END OF RECEIVE SERIAL INPUT INTERRUPT ROUTINE ********
986  1A7C              MDEND   =       *
987  1A7C                      HILO    MDEND
988  001A              +MDENDH   =       MDEND/256
989  007C              +MDENDL   =       (-256)*MDENDH+MDEND
990                             *=      $70C
991  070C  7C 1A               .BYTE   MDENDL,MDENDH   ;SET END ADDR IN FMS PAST RES DUP SO
992                    ;                               ;BUFFERS DON'T CLOBBER IT.
993  0100              STAK    =       $100
994  070E                      HILO    STAK
995  0001              +STAKH    =       STAK/256
```

```
      996  0000                    +STAKL      =       (-256)*STAKH+STAK
```

```
 997                            .PAGE
 998                    ;  ********  BEGINNING  OF  NON-RESIDENT  PORTION  OF  DUP  ********
 999                    ;
1000                    ;
1001  1D7C               NDOS    =       MDEND+$300      ;END OF THE SYSTEM BUFFERS AND MINIDUP
1002  070E                       HILO    NDOS
1003  001D               +NDOSH    =      NDOS/256
1004  007C               +NDOSL    =      (-256)*NDOSH+NDOS
1005                             *=      NDOS
1006  1D7C
1007  1D7C               PAR:    .RES    40              ;PARAMETER AREA
1008  001D               PARH    =       PAR/256
1009  007C               PARL    =       (-256)*PARH+PAR
1010  1DA4               LINE:   .RES    80              ;TYPE IN LINE BUFFER
1011  001D               LBUFH   =       LINE/256
1012  00A4               LBUFL   =       (-256)*LBUFH+LINE
1013  1DF4               DBUF:   .RES    $100            ;DATA BUFFER FOR COPY
1014  1E74               DB1     =       DBUF+$80
1015  1DF1               DB3     =       DBUF-3
1016  1EF4                       HILO    DBUF
1017  001D               +DBUFH    =      DBUF/256
1018  00F4               +DBUFL    =      (-256)*DBUFH+DBUF
1019  1EF4                       HILO    DB1
1020  001E               +DB1H     =      DB1/256
1021  0074               +DB1L     =      (-256)*DB1H+DB1
1022  1EF4                       HILO    DB3
1023  001D               +DB3H     =      DB3/256
1024  00F1               +DB3L     =      (-256)*DB3H+DB3
1025  0000               DBLL    =       0
1026  0001               DBLH    =       1               ;DATA BUFFER LENGTH=$100
1027  00FA               EDBLL   =       $FA             ;DATA BUFFER LENGTH USED IN USEPGM
1028  0000               EDBLH   =       0               ;MUST BE A MULTIPLE OF 125, SECTOR DATA
1029  1EF4               MENUSZ: .RES    1
1030  1EF5               PER:    .RES    1
1031  1EF6               UNNO:   .RES    1
1032  1EF7               RCNT:   .RES    1
1033  1EF8               SSTAT:  .RES    1
1034  1EF9               SWDP:   .RES    5
1035  1EFE               CSRC:   .RES    1
1036  1EFF               CDES:   .RES    1
1037  1F00               SAVX:   .RES    1
1038  1F01               PTR:    .RES    1
1039  1F02               IPTR:   .RES    1
1040  1F03               CTR:    .RES    1
1041  1F04               T1:     .RES    2
1042  1F04               BUFLEN  =       T1              ;SAVE AREA FOR BUFR LEN, USED IN USEPGM
1043  1F06               STVEC:  .RES    2               ;A TEMP OF SOME KIND
1044  1F06               MLT125  =       STVEC           ;TEMP STORE FOR MULTIPLE OF 125, USEPGM
1045  1F08               SECSIZ: .RES    2               ;TO STORE SECT SIZE IN BYTES FOR DUP DSK
1046  1F0A               EOFFLG: .RES    1               ;ENDFILE FLAG FOR SOURCE IN DUPFIL
1047  1F0B               FTRF:   .RES    1               ;FIRST TIME READ FLAG USED IN DUPFIL
1048  1F0B               TWODRV  =       FTRF            ;FLAG TO SHOW IF 1 OR 2 DRIVES. USED IN
1049  1F0C               DTH     =*
1050  1F0C                       HILO    DTH
```

```
     1051  001F                    +DTHH     =       DTH/256
     1052  000C                    +DTHL     =       (-256)*DTHH+DTH
     1053  1F0C  45 3A 9B           EDN      .BYTE    'E:',CR
     1054  001F                     EDH      =        EDN/256
     1055  000C                     EDL      =        (-256)*EDH+EDN
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    1056
    1057
    1058
    1059
    1060  1F0F  7D
    1061  1F10  44 49 53 4B
    1062  1F14  20 4F 50 45
    1063  1F18  52 41 54 49
    1064  1F1C  4E 47 20 53
    1065  1F20  59 53 54 45
    1066  1F24  4D 20 49 49
    1067  1F28  20 56 45 52
    1068  1F2C  53 49 4F 4E
    1069  1F30  20 32 2E 30
    1070  1F34  53 9B
    1071  1F36  43 4F 50 59
    1072  1F3A  52 49 47 48
    1073  1F3E  54 20 31 39
    1074  1F42  38 30 20 41
    1075  1F46  54 41 52 49
    1076  1F4A  9B 9B
    1077  1F4C  41 2E 20 44
    1078  1F50  49 53 4B 20
    1079  1F54  44 49 52 45
    1080  1F58  43 54 4F 52
    1081  1F5C  59 20 49 2E
    1082  1F60  20 46 4F 52
    1083  1F64  4D 41 54 20
    1084  1F68  44 49 53 4B
    1085  1F6C  9B
    1086  1F6D  42 2E 20 52
    1087  1F71  55 4E 20 43
    1088  1F75  41 52 54 52
    1089  1F79  49 44 47 45
    1090  1F7D  20 20 4A 2E
    1091  1F81  20 44 55 50
    1092  1F85  4C 49 43 41
    1093  1F89  54 45 20 44
    1094  1F8D  49 53 4B 9B
```

```
        .PAGE
;  ****  DOS  MENU  ****
;
;
DMENU   .BYTE   CLSCR
        .BYTE   'DISK OPERATING SYSTEM II VERSION 2.0S',CR




        .BYTE   'COPYRIGHT 1980 ATARI',CR,CR


        .BYTE   'A. DISK DIRECTORY I. FORMAT DISK',CR


        .BYTE   'B. RUN CARTRIDGE  J. DUPLICATE DISK',CR
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    1095
    1096  1F91  43 2E 20 43        .BYTE   'C. COPY FILE       K. BINARY SAVE',CR
    1097  1F95  4F 50 59 20
    1098  1F99  46 49 4C 45
    1099  1F9D  20 20 20 20
    1100  1FA1  20 20 4B 2E
    1101  1FA5  20 42 49 4E
    1102  1FA9  41 52 59 20
    1103  1FAD  53 41 56 45
    1104  1FB1  9B
    1105  1FB2  44 2E 20 44        .BYTE   'D. DELETE FILE(S) L. BINARY LOAD',CR
    1106  1FB6  45 4C 45 54
    1107  1FBA  45 20 46 49
    1108  1FBE  4C 45 28 53
    1109  1FC2  29 20 4C 2E
    1110  1FC6  20 42 49 4E
    1111  1FCA  41 52 59 20
    1112  1FCE  4C 4F 41 44
    1113  1FD2  9B
    1114  1FD3  45 2E 20 52        .BYTE   'E. RENAME FILE     M. RUN AT ADDRESS',CR
    1115  1FD7  45 4E 41 4D
    1116  1FDB  45 20 46 49
    1117  1FDF  4C 45 20 20
    1118  1FE3  20 20 4D 2E
    1119  1FE7  20 52 55 4E
    1120  1FEB  20 41 54 20
    1121  1FEF  41 44 44 52
    1122  1FF3  45 53 53 9B
    1123  1FF7  46 2E 20 4C        .BYTE   'F. LOCK FILE       N. CREATE MEM.SAV',CR
    1124  1FFB  4F 43 4B 20
    1125  1FFF  46 49 4C 45
    1126  2003  20 20 20 20
    1127  2007  20 20 4E 2E
    1128  200B  20 43 52 45
    1129  200F  41 54 45 20
    1130  2013  4D 45 4D 2E
    1131  2017  53 41 56 9B
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    1132
    1133  201B  47 2E 20 55        .BYTE   'G. UNLOCK FILE    O. DUPLICATE FILE',CR
    1134  201F  4E 4C 4F 43
    1135  2023  4B 20 46 49
    1136  2027  4C 45 20 20
    1137  202B  20 20 4F 2E
    1138  202F  20 44 55 50
    1139  2033  4C 49 43 41        .BYTE   'H. WRITE DOS FILES',CR
    1140  2037  54 45 20 46
    1141  203B  49 4C 45 9B
    1142  203F  48 2E 20 57
    1143  2043  52 49 54 45
    1144  2047  20 44 4F 53
    1145  204B  20 46 49 4C
    1146  204F  45 53 9B
    1147  2052  1D 1D 1D 1D        .BYTE   CDN,CDN,CDN,CDN,CDN
    1148  2056  1D
    1149  2057                    DMEND   =*
    1150  0148                    DULEN   =       DMEND-DMENU
    1151  2057                            HILO    DULEN
    1152  0001                    +DULENH     =     DULEN/256
    1153  0048                    +DULENL     =     (-256)*DULENH+DULEN
    1154  2057                            HILO    DMENU
    1155  001F                    +DMENUH     =     DMENU/256
    1156  000F                    +DMENUL     =     (-256)*DMENUH+DMENU
    1157                            ;
    1158  2057  39 21 EE 26        DUJPT   .WORD   DIRLST,STCAR,CPYFIL,DELFIL,RENFIL,LKFIL,ULFIL
    1159  205B  78 23 C9 21
    1160  205F  37 26 70 29
    1161  2063  98 29
    1162  2065  D9 27 80 26        .WORD   WBOOT,FMTDSK,DUPDSK,SAVFIL,LDFIL,BRUN,MEMSAV
    1163  2069  58 2A 2E 2F
    1164  206D  1A 29 4C 27
    1165  2071  9A 27
    1166  2073  1E 2D               .WORD   DUPFIL
    1167  2075                            HILO    DUJPT
    1168  0020                    +DUJPTH     =     DUJPT/256
    1169  0057                    +DUJPTL     =     (-256)*DUJPTH+DUJPT
    1170  000F                     DUNUM  =       15              ;NUMBER OF FUNCTIONS
```

```
1171                            .PAGE
1172                    ;  **** DISK OPERATING SYS MONITOR ****
1173                    ;
1174                    ;
1175  2075  A2 FF      DOSOS   LDX     #$FF
1176  2077                    HILO    DOSOS
1177  0020              +DOSOSH   =       DOSOS/256
1178  0075              +DOSOSL   =       (-256)*DOSOSH+DOSOS
1179  2077  D8                CLD                     ;MAKE SURE DECIMAL MODE OFF
1180  2078  86 11            STX     BRKKEY
1181  207A  E8                INX
1182  207B  8E 9F 15          STX     LOADFG
1183  207E  A9 02             LDA     #2
1184  2080  85 52             STA     LMARGN
1185  2082  A9 27             LDA     #39
1186  2084  85 53             STA     RMARGN          ;SET MARGINS
1187  2086  A5 10             LDA     POKMSK          ;ENABLE BREAK INTERRRUPTS
1188  2088  09 80             ORA     #$80
1189  208A  85 10             STA     POKMSK
1190  208C  8D 0E D2          STA     IRQEN
1191  208F  20 76 19          JSR     INITIO          ;CLOSE FILES
1192                    ;
1193                    ;       DISK UTILITY MONITOR
1194                    ;
1195  2092              DSKUTL
1196  2092  A9 0F       DU1     LDA     #DUNUM
1197  2094  8D F4 1E            STA     MENUSZ          ;SET MENU SIZE
1198  2097  A9 57              LDA     #.LOW.DUJPTL
1199  2099  85 18              STA     JMPTBL
1200  209B  A9 20              LDA     #.LOW.DUJPTH
1201  209D  85 19              STA     JMPTBL+1        ;SET UP JUMP TABLE ADDRESS
1202                    ;  FALL THRU TO MENU SELECT
1203                    ;
1204                    ;
1205                    ;
1206                    ;  MENU SELECT MONITOR -- VECTORS TO ROUTINE SELECTED FROM MENU.
1207                    ;
1208  209F  A9 0F       SHMEN   LDA     #.LOW.DMENUL    ;GET MENU ADDRESS
1209  20A1  8D 44 03            STA     ICBAL
1210  20A4  A9 1F              LDA     #.LOW.DMENUH
1211  20A6  8D 45 03            STA     ICBAH
1212  20A9  A9 48              LDA     #.LOW.DULENL    ;GET MENU LENGTH
1213  20AB  8D 48 03            STA     ICBLL
1214  20AE  A9 01              LDA     #.LOW.DULENH
1215  20B0  8D 49 03            STA     ICBLH
1216  20B3  20 A3 31            JSR     DSPMSG          ;SHOW MENU
1217                    ;
1218                    ;       SELECT ITEM FROM MENU
```

```
ERR LINE  ADDR  B1 B2 B3 B4

1219
1220
1221
1222  20B6  A2 FF
1223  20B8  9A
1224  20B9  E8
1225  20BA  8E 41 23
1226  20BD  A9 1A
1227  20BF  A2 21
1228  20C1  20 BE 19
1229  20C4  A9 40
1230  20C6  8D BE 02
1231  20C9  20 7E 30
1232
1233  20CC  C9 9B
1234  20CE  F0 CF
1235
1236  20D0  38
1237  20D1  E9 41
1238  20D3  30 2E
1239  20D5  CD F4 1E
1240  20D8  10 29
1241  20DA  0A
1242  20DB  A8
1243  20DC  B1 18
1244  20DE  C8
1245  20DF  85 1A
1246  20E1  B1 18
1247  20E3  85 1B
1248  20E5  A0 01
1249  20E7  B1 1A
1250  20E9  AA
1251  20EA  88
1252  20EB  B1 1A
1253  20ED  20 B5 31
1254  20F0  20 BB 31
1255  20F3  A5 1A
1256  20F5  18
1257  20F6  69 02
1258  20F8  85 1A
1259  20FA  A5 1B
1260  20FC  69 00
1261  20FE  85 1B
1262  2100  6C 1A 00
1263  2103  A9 0D
1264  2105  A2 21
1265  2107  20 B5 31
1266  210A  4C B6 20
```

```
          .PAGE
;   ****  FUNCTIONS COME HERE WHEN THEY ARE DONE  ****
;
MENUSL    LDX     #$FF            ;RESET STACK AT THIS POINT
          TXS
          INX
          STX     WCFLAG          ;CLEAR WILD-CARD FLAG
          LDA     #.LOW.SITL      ;SELECT ITEM MESSAGE
          LDX     #.LOW.SITH
          JSR     PRNTMSG
          LDA     #$40            ;MAKE SURE UPPER CASE
          STA     SHFLOK
          JSR     CHRGET          ;GO GET KEYBOARD CHAR.
;
          CMP     #CR             ;IF CR RE-DISPLAY MENU
          BEQ     SHMEN
;
          SEC
          SBC     #'A             ;CONVRT ASCII CHAR. TO BINARY # & SUB 1
          BMI     RANGE           ;IF ASCII CHAR NOT A #, GO READ AGAIN
          CMP     MENUSZ          ;IS THE # ENTERED > MENU SIZE?
          BPL     RANGE           ;IF YES, GO READ AGAIN
          ASL     A
          TAY                     ;SET INDEX TO (MENU # - 1) * 2
          LDA     (JMPTBL),Y
          INY
          STA     RAMLO           ;GET STRING POINTER
          LDA     (JMPTBL),Y
          STA     RAMLO+1
          LDY     #1              ;LOAD STRING POINTER INTO REGISTERS
          LDA     (RAMLO),Y       ;FOR DSPLIN
          TAX
          DEY
          LDA     (RAMLO),Y
          JSR     DSPLIN          ;PRINT MODULES INITIAL STRING
          JSR     SCROL           ;SCROLL INPUT WINDOW
          LDA     RAMLO           ;INC BY 2 TO POINT PAST STRING POINTER
          CLC
          ADC     #2
          STA     RAMLO
          LDA     RAMLO+1
          ADC     #0              ;CARRY
          STA     RAMLO+1         ;PUT HI BYTE
          JMP     (RAMLO)         ;JUMP TO ROUTINE SELECTED BY MENU.
RANGE     LDA     #.LOW.NSIL
          LDX     #.LOW.NSIH      ;NO SUCH ITEM MESSAGE
          JSR     DSPLIN
          JMP     MENUSL
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    1267
    1268  210D  4E 4F 20 53        .PAGE
    1269  2111  55 43 48 20  NSI   .BYTE   'NO SUCH ITEM',CR
    1270  2115  49 54 45 4D
    1271  2119  9B
    1272
    1273
    1274                          ;
    1275  211A  53 45 4C 45       ;PROMPT FOR MENU SELECTION OR REDISPLAY MENU - RETURN IS IN INVERSE
    1276  211E  43 54 20 49       ;
    1277  2122  54 45 4D 20  SIT   .BYTE   'SELECT ITEM OR ',$D2,$C5,$D4,$D5,$D2,$CE
    1278  2126  4F 52 20 D2
    1279  212A  C5 D4 D5 D2
    1280  212E  CE
    1281  212F  20 46 4F 52        .BYTE   ' FOR MENU',CR
    1282  2133  20 4D 45 4E
    1283  2137  55 9B
    1284  2139                    HILO    NSI
    1285  0021              +NSIH   =       NSI/256
    1286  000D              +NSIL   =       (-256)*NSIH+NSI
    1287  2139                    HILO    SIT
    1288  0021              +SITH   =       SIT/256
    1289  001A              +SITL   =       (-256)*SITH+SIT
    1290  20B6               MNSL   =       MENUSL
    1291  2139                    HILO    MNSL
    1292  0020              +MNSLH  =       MNSL/256
    1293  00B6              +MNSLL  =       (-256)*MNSLH+MNSL
```

```
1294                                    .PAGE
1295                                ;  **** DIRECTORY LISTING ROUTINE ****
1296                                ;
1297                                ;
1298  2139  A7 21                   DIRLST  .WORD   DLMG
1299  213B  20 CF 30                        JSR     GETIC1
1300  213E  20 C4 2E                        JSR     USEBUF          ;INIT BUFADR & BUFLEN
1301  2141  AE 01 1F                        LDX     PTR
1302  2144  A9 9B                           LDA     #CR
1303  2146  9D 7B 1D                        STA     PAR-1,X         ;ASSURE GOOD TERM
1304  2149  BD 7A 1D                        LDA     PAR-2,X         ;LAST CHAR OF SEARCH SPEC
1305  214C  C9 3A                           CMP     #':             ;IF COLON, ADD *.*
1306  214E  D0 18                           BNE     GLF
1307  2150  A9 2A                           LDA     #'*
1308  2152  9D 7B 1D                        STA     PAR-1,X
1309  2155  9D 7D 1D                        STA     PAR+1,X
1310  2158  A9 2E                           LDA     #'.
1311  215A  9D 7C 1D                        STA     PAR,X
1312  215D  A9 9B                           LDA     #CR
1313  215F  9D 7E 1D                        STA     PAR+2,X
1314  2162  E8                              INX
1315  2163  E8                              INX
1316  2164  E8                              INX
1317  2165  8E 01 1F                        STX     PTR
1318  2168  8E 00 1F              GLF       STX     SAVX
1319  216B  A2 20                           LDX     #$20
1320  216D  20 DD 31                        JSR     PIOCB
1321  2170  20 E8 30                        JSR     GETFIL
1322  2173  20 C4 30                        JSR     PERX
1323  2176  A9 06                           LDA     #6              ;READ DIR INFO
1324  2178  A2 10                           LDX     #$10
1325  217A  9D 4A 03                        STA     ICAX1,X
1326  217D  A9 03                           LDA     #OPEN           ;OPEN
1327  217F  9D 42 03                        STA     ICCOM,X
1328  2182  8E FE 1E                        STX     CSRC            ;COPY SOURCE=DIRECTORY INFO
1329  2185  E0 10                           CPX     #$10
1330  2187  D0 01                           BNE     *+3
1331  2189  20 EE 31                        JSR     CIOCL
1332  218C  AD 01 1F                        LDA     PTR
1333  218F  38                              SEC
1334  2190  ED 00 1F                        SBC     SAVX
1335  2193  C9 03                           CMP     #3              ;IF ONLY 3 CHARS, IS 'D:'CR, USE DEFAULT
1336  2195  F0 03                           BEQ     DLST1
1337  2197  4C 5E 25              DLSTO     JMP     PDES            ;GO INTO COPY
1338  219A  AE 00 1F              DLST1     LDX     SAVX
1339  219D  BD 7C 1D                        LDA     PAR,X
1340  21A0  C9 44                           CMP     #'D
1341  21A2  D0 F3                           BNE     DLSTO
1342  21A4  4C 6C 25                        JMP     PDES1           ;GO INTO COPY WITH DES='E:'
```

```
    1343                                .PAGE
    1344  21A7  44 49 52 45    DLMG    .BYTE   'DIRECTORY--SEARCH SPEC,LIST FILE?',CR
    1345  21AB  43 54 4F 52
    1346  21AF  59 2D 2D 53
    1347  21B3  45 41 52 43
    1348  21B7  48 20 53 50
    1349  21BB  45 43 2C 4C
    1350  21BF  49 53 54 20
    1351  21C3  46 49 4C 45
    1352  21C7  3F 9B
```

```
1353
1354                                ;  ****  DELETE  FILE  ROUTINE  ****
1355                                ;
1356                                ;
1357  21C9  0D 23                   DELFIL  .WORD   DEMG
1358  21CB  20 CF 30                        JSR     GETIC1
1359  21CE  20 C4 30                        JSR     PERX            ;EXIT IF PARAM ERRORS
1360                                ;
1361  21D1  20 6E 26                        JSR     CHKVER          ;BE SURE THAT IT IS VER. 2 DISKETTE
1362                                ;
1363                                ;       CONTINUE WITH DELETE - ALLOW ONLY FOR DISK DEVICE ID
1364                                ;
1365  21D4  AD 7C 1D                        LDA     PAR             ;GET DEVICE
1366  21D7  C9 44                           CMP     #'D             ;ONLY ALLOW DELETE FOR D:
1367  21D9  F0 1A                           BEQ     DF1
1368  21DB  A9 E5                           LDA     #.LOW.NDFL
1369  21DD  A2 21                           LDX     #.LOW.NDFH
1370  21DF  20 B5 31                        JSR     DSPLIN
1371  21E2  4C B6 20                        JMP     MENUSL
1372  21E5  4E 4F 54 20     NDF             .BYTE   'NOT A DISK FILE',CR
1373  21E9  41 20 44 49
1374  21ED  53 4B 20 46
1375  21F1  49 4C 45 9B
1376  21F5                                  HILO    NDF
1377  0021                  +NDFH           =       NDF/256
1378  00E5                  +NDFL           =       (-256)*NDFH+NDF
1379  21F5  A2 10           DF1             LDX     #$10
1380  21F7  AD 9E 15                        LDA     OPT
1381  21FA  C9 4E                           CMP     #'N             ;IF OPTION=N, NO QUERY
1382  21FC  D0 0B                           BNE     DWQ             ;NO, DELETE WITH QUERY
1383  21FE  A9 21                           LDA     #DELETE
1384  2200  9D 42 03                        STA     ICCOM,X
1385  2203  20 EE 31                        JSR     CIOCL
1386  2206  4C B6 20                        JMP     MENUSL
1387  2209  A9 F7           DWQ             LDA     #.LOW.TYQL
1388  220B  A2 22                           LDX     #.LOW.TYQH
1389  220D  20 B5 31                        JSR     DSPLIN          ;SAY TYPE Y TO DELETE...
1390  2210  A9 00                           LDA     #0
1391  2212  8D 02 1F                        STA     IPTR            ;HOW MANY FILES TO SKIP, NONE AT FIRST
1392  2215  A2 20                           LDX     #$20            ;SET UP DELETE IOCB
1393  2217  A9 21                           LDA     #DELETE
1394  2219  9D 42 03                        STA     ICCOM,X
1395  221C  A9 F1                           LDA     #.LOW.DB3L
1396  221E  9D 44 03                        STA     ICBAL,X
1397  2221  A9 1D                           LDA     #.LOW.DB3H
1398  2223  9D 45 03                        STA     ICBAH,X
1399  2226  A9 44                           LDA     #'D
1400  2228  8D F1 1D                        STA     DBUF-3
1401  222B  A9 3A                           LDA     #':
1402  222D  8D F3 1D                        STA     DBUF-1
1403  2230  AD 7D 1D                        LDA     PAR+1           ;DEVICE NUMBER OR : FROM OF INPUT
1404  2233  C9 3A                           CMP     #':
1405  2235  D0 02                           BNE     *+4
1406  2237  A9 31                           LDA     #'1
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    1407  2239  8D F2 1D          STA     DBUF-2          ;KLUDGE KLUDGE KLUDGE
    1408  223C  A2 10       IDRD  LDX     #$10
    1409  223E  A9 03              LDA     #OPEN
    1410  2240  9D 42 03          STA     ICCOM,X
    1411  2243  A9 06              LDA     #6
    1412  2245  9D 4A 03          STA     ICAX1,X         ;DIR READ OPEN
    1413  2248  A9 7C              LDA     #PARL
    1414  224A  9D 44 03          STA     ICBAL,X
    1415  224D  A9 1D              LDA     #PARH
    1416  224F  9D 45 03          STA     ICBAH,X
    1417  2252  20 EE 31          JSR     CIOCL
    1418  2255  A9 F4              LDA     #.LOW.DBUFL
    1419  2257  9D 44 03          STA     ICBAL,X
    1420  225A  A9 1D              LDA     #.LOW.DBUFH
    1421  225C  9D 45 03          STA     ICBAH,X
    1422  225F  A9 05              LDA     #GETREC
    1423  2261  9D 42 03          STA     ICCOM,X
    1424  2264  A9 00              LDA     #0
    1425  2266  8D 01 1F          STA     PTR             ;HOW MANY FILES WE HAVE SKIPPED
    1426                      ;
    1427                      ; READ FILENAME FROM DIR, QUERY AND DELETE
    1428                      ;
    1429  2269  A2 10       RDFN  LDX     #$10
    1430  226B  A9 00              LDA     #0
    1431  226D  9D 48 03          STA     ICBLL,X
    1432  2270  A9 01              LDA     #1
    1433  2272  9D 49 03          STA     ICBLH,X
    1434  2275  20 EE 31          JSR     CIOCL           ;READ A LINE FROM DIRECTORY
    1435  2278  AD F5 1D          LDA     DBUF+1          ;IF FILE LINE, THIS IS BLANK
    1436  227B  C9 20              CMP     #' 
    1437  227D  D0 68              BNE     DELX            ;THIS IS FREE BLOCKS LINE
    1438  227F  EE 01 1F          INC     PTR             ;COUNT THIS FILE
    1439  2282  AD 01 1F          LDA     PTR             ;HAVE WE SKIPPED ENUF YET
    1440  2285  CD 02 1F          CMP     IPTR
    1441  2288  30 DF              BMI     RDFN            ;BR IF NO
    1442  228A  A2 00              LDX     #0              ;PUT PTR
    1443  228C  A0 02              LDY     #2              ;GET PTR
    1444                      ;
    1445                      ; MESSAGE DELETE FILE NAMES
    1446                      ;
    1447  228E  B9 F4 1D    MDN1  LDA     DBUF,Y          ;END OF FILENAME
    1448  2291  C9 20              CMP     #' 
    1449  2293  F0 09              BEQ     MDN2
    1450  2295  9D F4 1D          STA     DBUF,X
    1451  2298  E8                INX
    1452  2299  C8                INY
    1453  229A  E0 08              CPX     #8
    1454  229C  30 F0              BMI     MDN1
    1455                      ;
    1456                      ; FILENAME IS MOVED, PUT .EXT
    1457                      ;
    1458  229E  A9 2E       MDN2  LDA     #'.
    1459  22A0  9D F4 1D          STA     DBUF,X
    1460  22A3  E8                INX
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    1461  22A4  A0 0A
    1462  22A6  B9 F4 1D
    1463  22A9  9D F4 1D
    1464  22AC  C8
    1465  22AD  E8
    1466  22AE  C0 0D
    1467  22B0  30 F4
    1468  22B2  8E 00 1F
    1469  22B5  A9 3F
    1470  22B7  9D F4 1D
    1471  22BA  E8
    1472  22BB  A9 9B
    1473  22BD  9D F4 1D
    1474  22C0  A9 F1
    1475  22C2  A2 1D
    1476  22C4  20 B5 31
    1477  22C7  20 7E 30
    1478  22CA  C9 59
    1479  22CC  D0 9B
    1480  22CE  AD 01 1F
    1481  22D1  8D 02 1F
    1482  22D4  AE 00 1F
    1483  22D7  A9 9B
    1484  22D9  9D F4 1D
    1485  22DC  A2 20
    1486  22DE  20 EE 31
    1487  22E1  20 ED 22
    1488  22E4  4C 3C 22
    1489  22E7  20 ED 22
    1490  22EA  4C B6 20
    1491  22ED  A2 10
    1492  22EF  A9 0C
    1493  22F1  9D 42 03
    1494  22F4  4C EE 31
    1495  22F7  54 59 50 45
    1496  22FB  20 22 59 22
    1497  22FF  20 54 4F 20
    1498  2303  44 45 4C 45
    1499  2307  54 45 2E 2E
    1500  230B  2E 9B
    1501  230D
    1502  0022
    1503  00F7
    1504  230D  44 45 4C 45
    1505  2311  54 45 20 46
    1506  2315  49 4C 45 20
    1507  2319  53 50 45 43
    1508  231D  9B
    1509
```

```
        LDY     #10             ;WHERE EXT IS
MDN3    LDA     DBUF,Y
        STA     DBUF,X
        INY
        INX
        CPY     #13
        BMI     MDN3
        STX     SAVX            ;PUT CR HERE LATER
        LDA     #'?             ;FOR QUERY
        STA     DBUF,X
        INX
        LDA     #CR
        STA     DBUF,X
        LDA     #.LOW.DB3L
        LDX     #.LOW.DB3H
        JSR     DSPLIN          ;GO ASK ABOUT THIS FILE
        JSR     CHRGET
        CMP     #'Y
        BNE     RDFN            ;GO DO NEXT FILENAME
        LDA     PTR             ;NUMBER FILES WE HAVE GONE THRU SO FAR
        STA     IPTR            ;IS NEW NUMBER TO SKIP
        LDX     SAVX
        LDA     #CR
        STA     DBUF,X
        LDX     #$20            ;DELETE IOCB
        JSR     CIOCL
        JSR     CLOS1
        JMP     IDRD            ;CLOSE AND REOPEN DIR READ FILE
DELX    JSR     CLOS1           ;CLOSE DIR READ FILE
        JMP     MENUSL
CLOS1   LDX     #$10
        LDA     #CLOSE
        STA     ICCOM,X
        JMP     CIOCL           ;DO CLOSE AND RETURN
TYQ     .BYTE   'TYPE ',$22,'Y',$22,' TO DELETE...',CR



        HILO    TYQ
+TYQH   =       TYQ/256
+TYQL   =       (-256)*TYQH+TYQ
DEMQ    .BYTE   'DELETE FILE SPEC',CR


;LIST
```

```
1510                                    .PAGE
1511                                ;  **** COPY FILE ROUTINE ****
1512                                ;
1513                                ;
1514  231E  43 4F 50 59            CPMG    .BYTE   'COPY--FROM, TO?',CR
1515  2322  2D 2D 46 52
1516  2326  4F 4D 2C 20
1517  232A  54 4F 3F 9B
1518  232E  4F 50 54 49            OE      .BYTE   'OPTION NOT ALLOWED',CR
1519  2332  4F 4E 20 4E
1520  2336  4F 54 20 41
1521  233A  4C 4C 4F 57
1522  233E  45 44 9B
1523  2341                                HILO    OE
1524  0023                          +OEH  =       OE/256
1525  002E                          +OEL  =       (-256)*OEH+OE
1526                                ;
1527                                ;
1528                                ;
1529                                ;
1530                                ;
1531  2341                          WCFLAG: .RES  1
1532  2342                          WCSKP1: .RES  1
1533  2343                          WCSKP2: .RES  1
1534  0014                          WCBUFL  =     20
1535  2344                          WCBUF:  .RES  WCBUFL
1536  2358  20 20 43 4F            WCOPYM  .BYTE   '  COPYING---'
1537  235C  50 59 49 4E
1538  2360  47 2D 2D 2D
1539  2364  44 4E 3A                WCBUF2  .BYTE   'DN:'
1540  2367                                  .RES    WCBUFL-3
1541  2378  1E 23                   CPYFIL  .WORD   CPMG            ;COPY FILE PROMPT
1542  237A  20 CF 30                        JSR     GETIC1          ;GET SOURCE DEVICE, ETC.
1543  237D  AD 01 1F                        LDA     PTR
1544  2380  8D 00 1F                        STA     SAVX
1545  2383  AD 7C 1D                        LDA     PAR             ;GET 1ST CHAR. OF DEVICE
1546  2386  C9 44                           CMP     #'D             ;TEST IF IT IS THE DISK
1547  2388  D0 07                           BNE     JMPNWC          ;BR IF NOT THE DISK (THEN USE OLD CODE)
1548  238A  A2 00                           LDX     #0              ;LOOK AT SOURCE FILE SPEC.
1549  238C  20 D7 2E                        JSR     LOOKWC          ;LOOK FOR WILDCARDS IN FILE SPEC.
1550  238F  F0 03                           BEQ     CPYFL1          ;BRANCH IF WILDCARDS USED IN DISK SPEC.
1551  2391  4C E1 24                JMPNWC  JMP     NOTWC           ;USE OLD CODE
1552  2394  A9 80                    CPYFL1  LDA     #$80
1553                                ;
1554                                ;
1555  2396  8D 41 23                WCINIT  STA     WCFLAG          ;'WLDCARD' MODE  (COPY-FILE OR DUP-FIL
1556  2399  A9 00                           LDA     #0
1557  239B  8D 42 23                        STA     WCSKP1
1558                                ;
1559  239E  A9 00                    WCOPYL  LDA     #0
1560  23A0  8D 43 23                        STA     WCSKP2
1561  23A3  A2 10                           LDX     #$10            ;OPEN DIRECTORY
1562  23A5  A9 06                           LDA     #6
1563  23A7  9D 4A 03                        STA     ICAX1,X
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    1564  23AA  A9 03
    1565  23AC  9D 42 03
    1566  23AF  A9 7C
    1567  23B1  9D 44 03
    1568  23B4  A9 1D
    1569  23B6  9D 45 03
    1570  23B9  20 EE 31
    1571
    1572
    1573  23BC  A9 05
    1574  23BE  9D 42 03
    1575  23C1  A9 14
    1576  23C3  9D 48 03
    1577  23C6  A9 00
    1578  23C8  9D 49 03
    1579  23CB  A9 44
    1580  23CD  9D 44 03
    1581  23D0  A9 23
    1582  23D2  9D 45 03
    1583  23D5  20 EE 31
    1584
    1585  23D8  AD 44 23
    1586  23DB  C9 30
    1587  23DD  90 0F
    1588  23DF  C9 3A
    1589  23E1  B0 0B
    1590
    1591  23E3  A9 0C
    1592  23E5  9D 42 03
    1593  23E8  20 EE 31
    1594  23EB  4C B6 20
    1595
    1596
    1597  23EE  AD 42 23
    1598  23F1  CD 43 23
    1599  23F4  F0 05
    1600
    1601  23F6  EE 43 23
    1602  23F9  D0 C1
    1603
    1604  23FB  EE 42 23
    1605
    1606  23FE  A9 0C
    1607  2400  9D 42 03
    1608  2403  20 EE 31
    1609
    1610
    1611  2406  A0 02
    1612  2408  B9 4E 23
    1613  240B  D9 15 24
    1614  240E  D0 08
    1615  2410  88
    1616  2411  10 F5
    1617  2413  30 89
```

```
          LDA   #OPEN
          STA   ICCOM,X
          LDA   #.LOW.PAR
          STA   ICBAL,X
          LDA   #.HIGH.PAR
          STA   ICBAH,X
          JSR   CIOCL
;
;
WCOPYR    LDA   #GETREC          ;READ DIRECTORY
          STA   ICCOM,X
          LDA   #WCBUFL
          STA   ICBLL,X
          LDA   #0
          STA   ICBLH,X
          LDA   #.LOW.WCBUF
          STA   ICBAL,X
          LDA   #.HIGH.WCBUF
          STA   ICBAH,X
          JSR   CIOCL
;
          LDA   WCBUF            ;IF 1ST CHAR. OF DIR READ IS A #-IT IS T
          CMP   #'0
          BCC   WCGOT
          CMP   #':
          BCS   WCGOT
;
          LDA   #CLOSE           ;ALL DONE -- NORM EXIT OF WILDCARD COPY
          STA   ICCOM,X
          JSR   CIOCL
          JMP   MENUSL
;
;
WCGOT     LDA   WCSKP1           ;IF ALREADY COPIED OR SKIPPED THIS FILE
          CMP   WCSKP2
          BEQ   SKIP1
;
          INC   WCSKP2
          BNE   WCOPYR
;
SKIP1     INC   WCSKP1
;
          LDA   #CLOSE           ;CLOSE DIRECTORY READ FILE
          STA   ICCOM,X
          JSR   CIOCL
;
;
          LDY   #2               ;DON'T COPY  SYS  FILES
SYSLOP    LDA   WCBUF+10,Y
          CMP   DOTSYS,Y
          BNE   NOSYS
          DEY
          BPL   SYSLOP
          BMI   WCOPYL
```

```
1618
1619  2415  53 59 53          ;
                        DOTSYS  .BYTE   'SYS'
1620                    ;
1621  2418  A0 31        NOSYS   LDY     #'1             ;CALC SOURCE DRIVE NUMBER
1622  241A  AD 7D 1D            LDA     PAR+1
1623  241D  C9 3A               CMP     #':
1624  241F  F0 01               BEQ     WCGOT1
1625  2421  A8                  TAY
1626  2422  8C 65 23    WCGOT1  STY     WCBUF2+1
1627                    ;
1628                    ;
1629  2425  A2 02               LDX     #2              ;COMPRESS SPACES, ADD ':', ADD 'CR'
1630  2427  A0 03               LDY     #3
1631                    ;
1632  2429  BD 44 23    COMPR1  LDA     WCBUF,X
1633  242C  C9 20               CMP     #'
1634  242E  F0 04               BEQ     COMPR2
1635  2430  99 64 23            STA     WCBUF2,Y
1636  2433  C8                  INY
1637                    ;
1638  2434  E8          COMPR2  INX
1639  2435  E0 0A               CPX     #10
1640  2437  D0 F0               BNE     COMPR1
1641                    ;
1642  2439  BD 44 23            LDA     WCBUF,X
1643  243C  C9 20               CMP     #'
1644  243E  F0 16               BEQ     COMPR5
1645  2440  A9 2E               LDA     #'.
1646  2442  99 64 23            STA     WCBUF2,Y
1647  2445  C8                  INY
1648  2446  BD 44 23    COMPR3  LDA     WCBUF,X
1649  2449  C9 20               CMP     #'
1650  244B  F0 04               BEQ     COMPR4
1651  244D  99 64 23            STA     WCBUF2,Y
1652  2450  C8                  INY
1653  2451  E8          COMPR4  INX
1654  2452  E0 0D               CPX     #13
1655  2454  D0 F0               BNE     COMPR3
1656                    ;
1657  2456  A9 9B        COMPR5  LDA     #CR
1658  2458  99 64 23            STA     WCBUF2,Y
1659                    ;
1660                    ;
1661  245B  A9 58               LDA     #.LOW.WCOPYM    ;PRINT 'COPYING---DEV:FILENAME.EXT' MSG
1662  245D  A2 23               LDX     #.HIGH.WCOPYM
1663  245F  20 B5 31            JSR     DSPLIN
1664                    ;
1665  2462  2C 41 23            BIT     WCFLAG
1666  2465  50 0F               BVC     WCOPY           ;BR TO MIDDLE OF DUP FILE ROUTINE IF DU
1667                    ;
1668  2467  A2 10               LDX     #$10            ;SET UP BUFR ADDR TO PNT TO WLDCARD FIL
1669  2469  A9 64               LDA     #.LOW.WCBUF2
1670  246B  9D 44 03            STA     ICBAL,X
1671  246E  A9 23               LDA     #.HIGH.WCBUF2
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    1672  2470  9D 45 03          STA     ICBAH,X
    1673  2473  4C 4F 2D          JMP     WCDUPS
    1674                        ;
    1675  2476  20 41 2E  WCOPY   JSR     USEPGM          ;SET BUFFER SIZES
    1676  2479  A2 10            LDX     #$10            ;OPEN COPY SOURCE FILE
    1677  247B  A9 03            LDA     #OPEN
    1678  247D  9D 42 03          STA     ICCOM,X
    1679  2480  A9 04            LDA     #4
    1680  2482  9D 4A 03          STA     ICAX1,X
    1681  2485  A9 64            LDA     #.LOW.WCBUF2
    1682  2487  9D 44 03          STA     ICBAL,X
    1683  248A  A9 23            LDA     #.HIGH.WCBUF2
    1684  248C  9D 45 03          STA     ICBAH,X
    1685  248F  8E FE 1E          STX     CSRC
    1686  2492  20 EE 31          JSR     CIOCL
    1687                        ;
    1688  2495  A2 20            LDX     #$20
    1689  2497  20 DD 31          JSR     PIOCB           ;GET COPY DESTINATION FILE
    1690  249A  AD 01 1F          LDA     PTR             ;SAVE PTR,IPTR- MIGHT REPET GETTING 2ND
    1691  249D                  MES
    1692  249D  48                PHA
    1693  249E  AD 02 1F          LDA     IPTR
    1694  24A1  48                PHA
    1695  24A2  20 E8 30          JSR     GETFIL          ;GET 2ND FILE NAME TO PAR
    1696  24A5  68                PLA                     ;RECOVER  IPTR,PTR
    1697  24A6  8D 02 1F          STA     IPTR
    1698  24A9  68                PLA
    1699  24AA  8D 01 1F          STA     PTR
    1700  24AD  AE 00 1F          LDX     SAVX
    1701  24B0  BD 7C 1D          LDA     PAR,X
    1702  24B3  C9 44            CMP     #'D
    1703  24B5  F0 03            BEQ     WCOPY0
    1704  24B7  4C 5E 25          JMP     PDES            ;JMP TO OLD CPY-FILE CODE IF NOT DSK DES
    1705                        ;
    1706  24BA  A0 31  WCOPY0    LDY     #'1             ;CALCULATE DESTINATION DRIVE #
    1707  24BC  BD 7D 1D          LDA     PAR+1,X
    1708  24BF  C9 3A            CMP     #':
    1709  24C1  F0 01            BEQ     WCOPY1
    1710                        ;
    1711  24C3  A8                TAY
    1712  24C4  CC 65 23  WCOPY1  CPY     WCBUF2+1
    1713  24C7  D0 06            BNE     WCOPY2
    1714  24C9  20 AA 19          JSR     CLOSX           ;CANT COPY TO SAME DRVE NMBR - ERR & EXI
    1715                        ;
    1716  24CC  4C 74 25          JMP     ODMS
    1717                        ;
    1718                        ;
    1719  24CF  A2 20  WCOPY2    LDX     #$20
    1720  24D1  8C 65 23          STY     WCBUF2+1        ;CHANGE FILESPEC TO DESTINATION
    1721  24D4  A9 64            LDA     #.LOW.WCBUF2
    1722  24D6  9D 44 03          STA     ICBAL,X
    1723  24D9  A9 23            LDA     #.HIGH.WCBUF2
    1724  24DB  9D 45 03          STA     ICBAH,X
    1725  24DE  4C 94 25          JMP     OPDES1          ;CONTINUE INTO OLD COPY-FILE CODE
```

```
1726                            ;
1727                            ;
1728  24E1                      NOTWC   =       *
1729  24E1  A2 20                       LDX     #$20            ;IOCB 3
1730  24E3  20 DD 31                    JSR     PIOCB
1731  24E6  20 E8 30                    JSR     GETFIL          ;GET SECOND FILENAME
1732                            ;
1733                            ;       MAKE SURE DESTINATION IS NOT DOS.SYS
1734                            ;
1735  24E9  AE 00 1F                    LDX     SAVX            ;ENTRY-INDEX TO DEST FILE SPEC
1736  24EC  20 ED 2E                    JSR     TSTDOS          ;WON'T RETURN IF IS DOS.SYS
1737                            ;
1738  24EF  AE 00 1F                    LDX     SAVX
1739  24F2  20 D7 2E                    JSR     LOOKWC
1740  24F5  D0 30                       BNE     NWCIND          ;BRANCH IF NO WILDCARDS IN DESTINATION
1741  24F7  A9 01                       LDA     #.LOW.NWAL
1742  24F9  A2 25                       LDX     #.LOW.NWAH
1743  24FB  20 B5 31                    JSR     DSPLIN
1744  24FE  4C B6 20                    JMP     MENUSL
1745  2501  57 49 4C 44        NWA     .BYTE   'WILD CARDS NOT ALLOWED IN DESTINATION',CR
1746  2505  20 43 41 52
1747  2509  44 53 20 4E
1748  250D  4F 54 20 41
1749  2511  4C 4C 4F 57
1750  2515  45 44 20 49
1751  2519  4E 20 44 45
1752  251D  53 54 49 4E
1753  2521  41 54 49 4F
1754  2525  4E 9B
1755  2527                            HILO    NWA
1756  0025                      +NWAH   =       NWA/256
1757  0001                      +NWAL   =       (-256)*NWAH+NWA
1758  2527                      NWCIND  =       *
1759  2527  20 C4 30                    JSR     PERX            ;IF PARAM ERRS, EXIT
1760  252A  20 41 2E                    JSR     USEPGM          ;ASK USR IF CAN USE PGM AREA OR DATA BFR
1761  252D                      PSRC    =       *
1762  252D  AD 7C 1D                    LDA     PAR             ;GET 1ST LETR OF PARAM
1763  2530  C9 4B                       CMP     #'K
1764  2532  F0 40                       BEQ     ODMS            ;K:GETS 'OPTION DOESNT MAKE SENSE' FOR N
1765  2534  C9 43                       CMP     #'C
1766  2536  F0 3C                       BEQ     ODMS            ;C: GETS 'OPTION DOESNOT MAKE SENSE' FOR
1767  2538  C9 45                       CMP     #'E             ;E: AS SOURCE IS SPECIAL
1768  253A  D0 08                       BNE     OPSRC           ;IF NO THEN OPEN SOURCE FILE
1769  253C  A2 00                       LDX     #0
1770  253E  8E FE 1E                    STX     CSRC
1771  2541  4C 5E 25                    JMP     PDES
1772  2544  C9 53               OPSRC   CMP     #'S
1773  2546  F0 2C                       BEQ     ODMS            ;S: AS SOURCE GETS O.D.M.S. FOR NOW
1774                            ;
1775                            ;       OPEN SOURCE FILE
1776                            ;
1777  2548  A2 10                       LDX     #$10
1778  254A  A9 03                       LDA     #OPEN
1779  254C  9D 42 03                    STA     ICCOM,X
```

```
ERR LINE  ADDR  B1 B2 B3 B4

1780  254F  A9 04
1781  2551  9D 4A 03
1782  2554  8E FE 1E
1783  2557  E0 10
1784  2559  D0 1F
1785  255B  20 EE 31
1786
1787
1788
1789  255E  AE 00 1F
1790  2561  BD 7C 1D
1791
1792  2564  C9 4B
1793  2566  F0 0C
1794
1795  2568  C9 45
1796  256A  D0 15
1797  256C  A9 00
1798  256E  8D FF 1E
1799  2571  4C AB 25
1800  2574  A9 2E
1801  2576  A2 23
1802  2578  20 B5 31
1803  257B  20 AA 19
1804  257E  4C B6 20
1805
1806  2581  C9 43
1807  2583  F0 EF
1808  2585  AE 9E 15
1809
1810  2588  E0 41
1811  258A  D0 08
1812  258C  C9 44
1813  258E  D0 E4
1814  2590  A9 09
1815  2592  D0 02
1816  2594  A9 08
1817  2596  A2 20
1818  2598  9D 4A 03
1819  259B  A9 03
1820  259D  9D 42 03
1821  25A0  8E FF 1E
1822  25A3  20 EE 31
1823  25A6  A9 00
1824  25A8  9D 4B 03
1825
1826
1827
1828  25AB  A9 07
1829  25AD  AE FE 1E
1830  25B0  AC FF 1E
1831  25B3  9D 42 03
1832  25B6  A9 0B
1833  25B8  99 42 03
```

```
        LDA     #4              ;OPEN IN
        STA     ICAX1,X
        STX     CSRC
        CPX     #$10
        BNE     *+33
        JSR     CIOCL           ;OPEN SOURCE FILE HERE
;
;       READY FOR OPEN OF DESTINATION
;
PDES    LDX     SAVX
        LDA     PAR,X
;
        CMP     #'K             ;IS DEST KEYBOARD?
        BEQ     ODMS            ;YES, THEN CAN'T DO IT
;
        CMP     #'E             ;CHECK FOR SPECIAL CASE
        BNE     OPDES           ;IF NOT
PDES1   LDA     #0              ;SPECIAL CASE - DONT OPEN, USE EXISTING
        STA     CDES
        JMP     DOCPY
ODMS    LDA     #OEL
        LDX     #OEH            ;SAY OPTION NOT ALLOWED
        JSR     DSPLIN
        JSR     CLOSX           ;CLOSE IOCB 1 & 2
        JMP     MENUSL
;
OPDES   CMP     #'C             ;C: GETS 'OPTION DOESNOT MAKE SENSE' FOR
        BEQ     ODMS            ;GET 2ND FILE OPTION
        LDX     OPT
;
        CPX     #'A             ;APPEND TO DISK FILE
        BNE     OPDES1
        CMP     #'D
        BNE     ODMS
        LDA     #9
        BNE     OPDES3
OPDES1  LDA     #8
OPDES3  LDX     #$20
        STA     ICAX1,X         ;OPEN TYPE OUT
        LDA     #OPEN
        STA     ICCOM,X OPEN
        STX     CDES
        JSR     CIOCL
        LDA     #0
        STA     ICAX2,X
;
;COPY FROM CSRC TO CDES
;
DOCPY   LDA     #GETCHR
GC1     LDX     CSRC
        LDY     CDES
        STA     ICCOM,X
        LDA     #PUTCHR
        STA     ICCOM,Y
```

```
ERR LINE  ADDR  B1 B2 B3 B4

1834  25BB  A5 1A
1835  25BD  9D 44 03
1836  25C0  99 44 03
1837  25C3  A5 1B
1838  25C5  9D 45 03
1839  25C8  99 45 03
1840  25CB  AE FE 1E
1841  25CE  AD 04 1F
1842  25D1  9D 48 03
1843  25D4  AD 05 1F
1844  25D7  9D 49 03
1845  25DA  20 56 E4
1846  25DD  8C F8 1E
1847  25E0  AE FF 1E
1848  25E3  AC FE 1E
1849  25E6  B9 48 03
1850  25E9  9D 48 03
1851  25EC  B9 49 03
1852  25EF  9D 49 03
1853  25F2  19 48 03
1854  25F5  F0 03
1855  25F7  20 EE 31
1856  25FA  AD F8 1E
1857  25FD  10 CC
1858  25FF  C9 88
1859  2601  F0 03
1860  2603  4C F6 31
1861  2606  AE FE 1E
1862  2609  F0 08
1863
1864
1865
1866  260B  A9 0C
1867  260D  9D 42 03
1868  2610  20 56 E4
1869  2613  AE FF 1E
1870  2616  F0 08
1871  2618  A9 0C
1872  261A  9D 42 03
1873  261D  20 56 E4
1874  2620  AE FF 1E
1875  2623  D0 07
1876  2625  A9 E9
1877  2627  A2 26
1878  2629  20 BE 19
1879  262C
1880
1881  262C  2C 41 23
1882  262F  10 03
1883  2631  4C 9E 23
1884  2634  4C B6 20
```

```
                LDA     BUFADR          ;ADDRESS OF BUFFER - EITHER
                STA     ICBAL,X         ;PGM AREA (MEMLO) OR DATA BUFFER (DBUF)
                STA     ICBAL,Y
                LDA     BUFADR+1        ;BUFADR IN LSB,MSB ORDER
                STA     ICBAH,X
                STA     ICBAH,Y
CLOOP           LDX     CSRC
                LDA     BUFLEN          ;LENGTH OF BUFFER ADDRESSED
                STA     ICBLL,X         ;BY BUFADR
                LDA     BUFLEN+1        ;BOTH BUFADR & BUFLEN ARE ASSIGNED
                STA     ICBLH,X         ;IN SUBROUTINE USEPGM
                JSR     CIO             ;READ FROM INPUT
                STY     SSTAT
                LDX     CDES
                LDY     CSRC
                LDA     ICBLL,Y
                STA     ICBLL,X
                LDA     ICBLH,Y
                STA     ICBLH,X
                ORA     ICBLL,Y         ;IF SOURCE FILE LENGTH = 0
                BEQ     CKRS            ;DON'T DO WRITE
                JSR     CIOCL           ;WRITE, ABORT IF ERROR
CKRS            LDA     SSTAT           ;GET READ OPERATION STATUS BACK
                BPL     CLOOP           ;IF OK, GO READ SOME MORE
                CMP     #$88            ;EOF STATUS
                BEQ     *+5
                JMP     CIOER           ;IF NOT, ABORT
CLOC            LDX     CSRC
                BEQ     DU4             ;IF E:, DONT CLOSE
;
;CLOSE SOURCE FILE
;
                LDA     #CLOSE
                STA     ICCOM,X
                JSR     CIO
DU4             LDX     CDES
                BEQ     DU3             ;IF DES=E:
                LDA     #CLOSE
                STA     ICCOM,X
                JSR     CIO
DU3             LDX     CDES
                BNE     DU6
                LDA     #.LOW.DDSK+1
                LDX     #.HIGH.(DDSK+1)
                JSR     PRNTMSG         ;PRNT A CR BEFOR SELECT OR WLDCARD PRMPT
DU6             =       *
;
                BIT     WCFLAG
                BPL     DU5
                JMP     WCOPYL          ;BRANCH BACK TO WILD CARD LOOP
DU5             JMP     MENUSL
```

```
ERR LINE  ADDR  B1 B2 B3 B4

1885
1886
1887
1888
1889
1890
1891
1892
1893
1894  2637  52 26
1895  2639  20 CF 30
1896  263C  20 DA 30
1897  263F  20 C4 30
1898
1899  2642  20 6E 26
1900
1901
1902
1903  2645  A9 20
1904  2647  A2 10
1905  2649  9D 42 03
1906  264C  20 EE 31
1907  264F  4C B6 20
1908  2652  52 45 4E 41
1909  2656  4D 45 20 2D
1910  265A  20 47 49 56
1911  265E  45 20 4F 4C
1912  2662  44 20 4E 41
1913  2666  4D 45 2C 20
1914  266A  4E 45 57 9B
1915
1916
1917
1918
1919
1920  266E  A0 01
1921  2670  AD 7D 1D
1922  2673  C9 3A
1923  2675  F0 03
1924  2677  29 0F
1925  2679  A8
1926  267A  8C F6 1E
1927
1928  267D  4C F3 28
1929
```

```
        .PAGE
;  ****  RENAME  FILE  ROUTINE  ****
;
;
;RENAME SETS UP IOCB #1 WITH THE OLD FILE NAME AND THE BUFFER ADDRESS
;POINTS TO THE NEW FILE NAME.  THE NEW FILE SPECIFICATION CANNOT HAVE
;A DEVICE ID.  THE DEVICE ID IS THE SAME AS SPECIFIED FOR THE OLD FILE
;D2:ABC.S2,QQQ.R3      THIS RENAMES ABC.S2 ON DRIVE #2 TO QQQ.R3
;
RENFIL  .WORD   RNMG
        JSR     GETIC1          ;GET OLD FILE SPEC & PUT ADDR IN IOCB
        JSR     GETNAME         ;GET NEW FILE NAME
        JSR     PERX            ;EXIT IF PARAMETER ERRORS
;
        JSR     CHKVER          ;MAKE SURE VER 2 DISKETTE
;
;       CONTINUE WITH RENAME
;
        LDA     #RENAME
        LDX     #$10
        STA     ICCOM,X
        JSR     CIOCL
        JMP     MENUSL
RNMG    .BYTE   'RENAME - GIVE OLD NAME, NEW',CR



;
;********************  SUBROUTINE  ********************
;
;       MAKE SURE THIS IS A VERSION 2 FORMAT DISK
;
CHKVER  LDY     #1              ;ASSUME DRIVE 1- GET DRIVE #
        LDA     PAR+1           ;TEST CHAR 2 OF FILE SPEC FOR SEMICOLON
        CMP     #':             ;IF IS, USING DEFAULT DRIVE (1)
        BEQ     DRV1            ;IT IS, SO SAVE DRIVE #
        AND     #$0F            ;ELSE CHAR 2 IS ASCII REP OF DRIVE #
        TAY                     ;CONVERT TO BINARY & SAVE IT
DRV1    STY     UNNO            ;SAVE DRIVE #
;
        JMP     TSTVER2         ;TST FOR VERS. 2 DISK- WONT RTURN IF NOT
;
```

```
1930                            .PAGE
1931                    ;  **** FORMAT DISK ROUTINE ****
1932                    ;
1933                    ;
1934  2680  B9 26      FMTDSK  .WORD   WHD
1935  2682  20 3C 30          JSR     GETLIN
1936  2685  20 BE 32          JSR     GETDN
1937  2688  18                CLC
1938  2689  69 30            ADC     #'0
1939  268B  8D E8 26          STA     DDSK
1940  268E  8D EB 26          STA     CDSK
1941  2691  20 C4 30          JSR     PERX
1942  2694  A9 D0            LDA     #.LOW.VFML      ;QUERY TO VERIFY DRIVE NUMBER
1943  2696  A2 26            LDX     #.LOW.VFMH
1944  2698  20 B5 31          JSR     DSPLIN
1945  269B  20 7E 30          JSR     CHRGET
1946  269E  C9 59            CMP     #'Y             ;SEE IF OK
1947  26A0  D0 14            BNE     FMX
1948  26A2  A9 EA            LDA     #.LOW.FDPL
1949  26A4  A2 10            LDX     #$10
1950  26A6  9D 44 03          STA     ICBAL,X
1951  26A9  A9 26            LDA     #.LOW.FDPH
1952  26AB  9D 45 03          STA     ICBAH,X
1953  26AE  A9 FE            LDA     #FORMAT
1954  26B0  9D 42 03          STA     ICCOM,X
1955  26B3  20 EE 31          JSR     CIOCL           ;CALL CIO TO DO FORMAT
1956  26B6  4C B6 20  FMX     JMP     MENUSL          ;EXIT.
1957  26B9  57 48 49 43  WHD  .BYTE   'WHICH DRIVE TO FORMAT?',CR
1958  26BD  48 20 44 52
1959  26C1  49 56 45 20
1960  26C5  54 4F 20 46
1961  26C9  4F 52 4D 41
1962  26CD  54 3F 9B
1963  26D0  54 59 50 45  VFM  .BYTE   'TYPE ',$22,'Y',$22,' TO FORMAT DISK '
1964  26D4  20 22 59 22
1965  26D8  20 54 4F 20
1966  26DC  46 4F 52 4D
1967  26E0  41 54 20 44
1968  26E4  49 53 4B 20
1969  26E8            DDSK:   .RES    1
1970  26E9  9B                .BYTE   CR
1971  26EA  44        FDP     .BYTE   'D'
1972  26EB            CDSK:   .RES    1
1973  26EC  3A 9B             .BYTE   ':',CR
1974  26EE                    HILO    WHD
1975  0026            +WHDH     =     WHD/256
1976  00B9            +WHDL     =     (-256)*WHDH+WHD
1977  26EE                    HILO    VFM
1978  0026            +VFMH     =     VFM/256
1979  00D0            +VFML     =     (-256)*VFMH+VFM
1980  26EE                    HILO    FDP
1981  0026            +FDPH     =     FDP/256
1982  00EA            +FDPL     =     (-256)*FDPH+FDP
```

```
1983                                .PAGE
1984                            ;   **** START CARTRIDGE ROUTINE ****
1985                            ;
1986                            ;
1987  E45F                      SYVBL   =       SYSVBV
1988  26EE                              HILO    SYVBL
1989  00E4                    +SYVBLH     =       SYVBL/256
1990  005F                    +SYVBLL     =       (-256)*SYVBLH+SYVBL
1991  E462                      XTVBL   =       XITVBV
1992  26EE                              HILO    XTVBL
1993  00E4                    +XTVBLH     =       XTVBL/256
1994  0062                    +XTVBLL     =       (-256)*XTVBLH+XTVBL
1995  26EE  4B 27               STCAR   .WORD   SCMG            ;NO MSG, PRINT A <CR>
1996  BFFD                      ROMTST  =       $BFFD
1997  26F0  AC FD BF                    LDY     ROMTST          ;TEST IF RAM OR OTHER
1998  26F3  A9 AA                       LDA     #$AA            ;PATTERN #1
1999  26F5  8D FD BF                    STA     ROMTST
2000  26F8  CD FD BF                    CMP     ROMTST
2001  26FB  D0 17                       BNE     NOTRAM          ;BRANCH IF NOT RAM
2002  26FD  A9 55                       LDA     #$55            ;PATTERN #2
2003  26FF  8D FD BF                    STA     ROMTST
2004  2702  CD FD BF                    CMP     ROMTST
2005  2705  D0 0D                       BNE     NOTRAM          ;BRANCH IF NOT RAM
2006                            ;
2007  2707  8C FD BF                    STY     ROMTST          ;THERE IS VALID RAM - SAY NO CART
2008  270A  A9 3F               NOCART  LDA     #.LOW.NCAL
2009  270C  A2 27                       LDX     #.LOW.NCAH      ;SAY NO CART
2010  270E  20 B5 31                    JSR     DSPLIN
2011  2711  4C B6 20                    JMP     MENUSL
2012                            ;
2013                            ;       CHECK IF ROM OR EMPTY ADDRESS SPACE
2014                            ;
2015  2714  AD FC BF             NOTRAM  LDA     $BFFC           ;KNOWN ROM ZERO BYTE
2016  2717  D0 F1                       BNE     NOCART          ;BRANCH IF EMPTY ADDRESS SPACE
2017                            ;
2018  2719  AA                          TAX                     ;SINCE EMPTY ADDR SPACE GIVES A RANDOM
2019  271A  AD FD BF             CKCART  LDA     ROMTST          ;VALUE, TEST THE SAME LOC MANY TIMES.
2020  271D  F0 EB                       BEQ     NOCART          ;BRANCH IF NO CARTRIDGE
2021  271F  CD FD BF                    CMP     ROMTST
2022  2722  D0 E6                       BNE     NOCART          ;BRANCH IF NO CARTRIDGE
2023  2724  E8                          INX
2024  2725  D0 F3                       BNE     CKCART          ;LOOP BACK
2025                            ;
2026                            ;
2027                            ;       RESET VERTICAL BLANK VECTORS BEFORE ENTERING CART
2028                            ;
2029  2727  20 76 19                    JSR     INITIO
2030  272A  A9 06                       LDA     #6              ;SET VVBLKI
2031  272C  A2 E4                       LDX     #.LOW.SYVBLH    ;HI BYTE
2032  272E  A0 5F                       LDY     #.LOW.SYVBLL
2033  2730  20 5C E4                    JSR     SETVBV
2034  2733  A9 07                       LDA     #7              ;SET VVBLKD
2035  2735  A2 E4                       LDX     #.LOW.XTVBLH
2036  2737  A0 62                       LDY     #.LOW.XTVBLL
```

```
2037  2739  20 5C E4        JSR     SETVBV
2038  273C  4C 12 19        JMP     CLMJMP
```

```
2039                                    .PAGE
2040  273F  4E 4F 20 43        NCA     .BYTE   'NO CARTRIDGE'
2041  2743  41 52 54 52
2042  2747  49 44 47 45
2043  274B  9B                SCMG    .BYTE   CR
2044  274C                            HILO    NCA
2045  0027                    +NCAH    =      NCA/256
2046  003F                    +NCAL    =      (-256)*NCAH+NCA
2047                            ;
2048                            ;
2049                            ;       ******* RUN AT ADDRESS *******
2050                            ;
2051                            ;
2052                            ;
2053  274C  68 27               BRUN    .WORD   BRMG
2054  274E  20 3C 30                    JSR     GETLIN
2055  2751  20 24 32                    JSR     GETNO
2056  2754  20 C4 30                    JSR     PERX
2057  2757  85 1A                       STA     RAMLO
2058  2759  86 1B                       STX     RAMLO+1
2059  275B  AD 03 1F                    LDA     CTR
2060  275E  C9 04                       CMP     #4
2061  2760  F0 50                       BEQ     MOUT1           ;RETURN TO MENU IF NO RUN ADDRESS GIVEN
2062  2762  20 76 19                    JSR     INITIO          ;CLOSE ALL IOCB'S, THEN REOPEN S/E
2063  2765  4C 20 19                    JMP     LMTR            ;LOAD MEM.SAV & JUMP TO ADDRESS
2064                            ;
2065                            ;
2066  2768  52 55 4E 20         BRMG    .BYTE   'RUN FROM WHAT ADDRESS?',CR
2067  276C  46 52 4F 4D
2068  2770  20 57 48 41
2069  2774  54 20 41 44
2070  2778  44 52 45 53
2071  277C  53 3F 9B
```

```
2072
2073                                ;  ****  CREATE  MEM.SAV  FILE  ON  DISK  ****
2074                                ;
2075                                ;
2076  277F  54 59 50 45             MEMS    .BYTE   'TYPE ',$22,'Y',$22,' TO CREATE MEM.SAV',CR
2077  2783  20 22 59 22
2078  2787  20 54 4F 20
2079  278B  43 52 45 41
2080  278F  54 45 20 4D
2081  2793  45 4D 2E 53
2082  2797  41 56 9B
2083  279A  7F 27                   MEMSAV  .WORD   MEMS
2084  279C  20 7E 30                        JSR     CHRGET          ;GET CHAR (CR)
2085  279F  C9 59                           CMP     #'Y
2086  27A1  D0 0C                           BNE     MOUT            ;BRANCH IF USER'S ANSWER NOT A Y
2087  27A3  20 73 18                        JSR     MEMSVQ          ;TRY TO OPEN MEM.SAV
2088  27A6  30 0D                           BMI     MCONT           ;IF FILE DOESN'T EXIST THEN JUMP
2089  27A8  A9 BD                           LDA     #.LOW.MEMSGL    ;ELSE 'MEMORY.SAVE' AREADY EXIST
2090  27AA  A2 27                           LDX     #.LOW.MEMSGH    ;
2091  27AC  20 B5 31                        JSR     DSPLIN          ;DISPLAY THIS FACT
2092  27AF  20 AA 19                MOUT    JSR     CLOSX           ;EXIT AFTER CLOSING IOCB1
2093  27B2  4C B6 20                MOUT1   JMP     MENUSL          ;
2094                                ;
2095                                ; WRITE MEMORY.SAVE TO DISK
2096                                ;
2097  27B5  20 46 17                MCONT   JSR     MWRITE          ;WRITE FILE
2098  27B8  10 F5                           BPL     MOUT
2099  27BA  4C F5 31                MERR    JMP     CIOER1          ;DISPLAY ERROR
2100                                ;
2101  27BD  4D 45 4D 2E             MEMSG   .BYTE   'MEM.SAV FILE ALREADY EXISTS',CR
2102  27C1  53 41 56 20
2103  27C5  46 49 4C 45
2104  27C9  20 41 4C 52
2105  27CD  45 41 44 59
2106  27D1  20 45 58 49
2107  27D5  53 54 53 9B
2108  27D9                                HILO    MEMSG
2109  0027                          +MEMSGH     =     MEMSG/256
2110  00BD                          +MEMSGL     =     (-256)*MEMSGH+MEMSG
```

```
2111                                    .PAGE
2112                                ;  **** WRITE DOS & DUP ****
2113                                ;
2114                                ;
2115  27D9  75 28                   WBOOT   .WORD   DOSDRV           ;ADDRESS OF DRIVE # PROMPT
2116                                ;
2117                                ;       RETRIEVE DRIVE NUMBER FROM USER.
2118                                ;
2119  27DB  20 3C 30                        JSR     GETLIN           ;GET INPUT
2120  27DE  20 BE 32                        JSR     GETDN            ;GET DRIVE AS NUMBER, VERIFY IT
2121  27E1  20 C4 30                        JSR     PERX             ;EXIT IF ERROR
2122  27E4  8D F6 1E                        STA     UNNO             ;SAVE IT FOR TSTVER2
2123  27E7  09 30                           ORA     #'0              ;TURN BACK TO ASCII REP
2124  27E9  8D CB 28                        STA     DS+1             ;STORE IN DOS.SYS FILE SPEC
2125  27EC  8D C7 28                        STA     QWMG+31          ;& IN PROMPT
2126                                ;
2127  27EF  20 F3 28                        JSR     TSTVER2          ;TST IF VERS. 2 DISK - IF ISNT WONT RTRN
2128                                ;
2129                                ;       ASK USER IF CAN WRITE DOS & DUP TO SPECIFIED DRIVE
2130                                ;
2131  27F2  A9 A8                           LDA     #.LOW.QWMGL      ;PRINT PROMPT
2132  27F4  A2 28                           LDX     #.LOW.QWMGH
2133  27F6  20 B5 31                        JSR     DSPLIN
2134  27F9  20 7E 30                        JSR     CHRGET
2135  27FC  C9 59                           CMP     #'Y
2136  27FE  D0 72                           BNE     WBX              ;EXIT UNLESS Y
2137                                ;
2138                                ;TELL USER WRITING DOS FILES AND WRITE DOS.SYS FIRST- JUST OPEN IT.
2139                                ;
2140  2800  A9 92                           LDA     #.LOW.WBMGL
2141  2802  A2 28                           LDX     #.LOW.WBMGH
2142  2804  20 B5 31                        JSR     DSPLIN
2143                                ;
2144  2807  A9 03                           LDA     #OPEN
2145  2809  A2 10                           LDX     #$10             ;OPEN DOS.SYS ON IOCB #1
2146  280B  9D 42 03                        STA     ICCOM,X          ;WILL CAUSE FMS TO REWRITE BOOT SECTOR
2147  280E  A9 CA                           LDA     #.LOW.DSL        ;& A COPY OF DOS.SYS
2148  2810  9D 44 03                        STA     ICBAL,X
2149  2813  A9 28                           LDA     #.LOW.DSH
2150  2815  9D 45 03                        STA     ICBAH,X
2151  2818  A9 08                           LDA     #8
2152  281A  9D 4A 03                        STA     ICAX1,X
2153  281D  20 EE 31                        JSR     CIOCL            ;DO OPEN, IF ERROR GOTO MENU
2154                                ;
2155  2820  A2 10                           LDX     #$10
2156  2822  A9 0C                           LDA     #CLOSE
2157  2824  9D 42 03                        STA     ICCOM,X
2158  2827  20 EE 31                        JSR     CIOCL            ;DONE CLOSE IT.
2159                                ;
2160                                ;       WRITE DUP.SYS - SWAP AREA FILE
2161                                ;
2162  282A  A2 0B                           LDX     #11              ;MOVE 11 CHARS
2163  282C  BD 2E 18                MDUPBL  LDA     DUPSYS-1,X
2164  282F  9D 7B 1D                        STA     PAR-1,X          ;MOVE FILE NAME TO PARAMETER LIST
```

```
ERR LINE  ADDR  B1 B2 B3 B4

 2165  2832  CA
 2166  2833  D0 F7
 2167  2835  AD CB 28
 2168  2838  8D 7D 1D
 2169
 2170  283B  8E 01 1F
 2171  283E  A2 10
 2172  2840  20 DD 31
 2173  2843  A9 0C
 2174  2845  8D E0 19
 2175  2848  A9 1F
 2176  284A  8D E1 19
 2177  284D  A9 05
 2178  284F  8D E2 19
 2179  2852  A9 F9
 2180  2854  8D F8 2F
 2181  2857  A9 13
 2182  2859  8D FD 2F
 2183  285C  A9 33
 2184  285E  8D E3 19
 2185  2861  48
 2186  2862  A9 75
 2187  2864  8D E0 02
 2188  2867  A9 20
 2189  2869  8D E1 02
 2190  286C  CE BE 18
 2191  286F  4C A0 2F
 2192  2872  4C B6 20
 2193  2875  44 52 49 56
 2194  2879  45 20 54 4F
 2195  287D  20 57 52 49
 2196  2881  54 45 20 44
 2197  2885  4F 53 20 46
 2198  2889  49 4C 45 53
 2199  288D  20 54 4F 3F
 2200  2891  9B
 2201  2892  57 52 49 54
 2202  2896  49 4E 47 20
 2203  289A  4E 45 57 20
 2204  289E  44 4F 53 20
 2205  28A2  46 49 4C 45
 2206  28A6  53 9B
 2207  28A8
 2208  0028
 2209  0092
```

```
        DEX
        BNE     MDUPBL
        LDA     DS+1            ;GET DRIVE NUMBER
        STA     PAR+1           ;PUT IT IN DUP.SYS FILE SPEC

        STX     PTR
        LDX     #$10
        JSR     PIOCB           ;PUT FILE NAME POINTER IN IOCB
        LDA     #.LOW.DTHL
        STA     LDST
        LDA     #.LOW.DTHH
        STA     LDST+1
        LDA     #.LOW.NMDUP
        STA     LDND
        LDA     #.LOW.LENL
        STA     WDRL+1
        LDA     #.LOW.LENH
        STA     WDRH+1
        LDA     #.HIGH.NMDUP
        STA     LDND+1
        PHA                     ;NO /A
        LDA     #.LOW.DOSOS
        STA     RUNAD
        LDA     #.HIGH.DOSOS
        STA     RUNAD+1         ;SET DUP.SYS RUN ADDRESS
        DEC     RUNQ+1          ;SET RUN FLAG
        JMP     NRUNAD          ;WRITE DUP.SYS
WBX     JMP     MENUSL
DOSDRV  .BYTE   'DRIVE TO WRITE DOS FILES TO?',CR



WBMG    .BYTE   'WRITING NEW DOS FILES',CR



        HILO    WBMG
+WBMGH  =       WBMG/256
+WBMGL  =       (-256)*WBMGH+WBMG
```

```
2210                                    .PAGE
2211  28A8  54 59 50 45        QWMG    .BYTE   'TYPE ',$22,'Y',$22,' TO WRITE DOS TO DRIVE   ',CR
2212  28AC  20 22 59 22
2213  28B0  20 54 4F 20
2214  28B4  57 52 49 54
2215  28B8  45 20 44 4F
2216  28BC  53 20 54 4F
2217  28C0  20 44 52 49
2218  28C4  56 45 20 20
2219  28C8  2E 9B
2220  28CA                          HILO    QWMG
2221  0028                  +QWMGH    =      QWMG/256
2222  00A8                  +QWMGL    =      (-256)*QWMGH+QWMG
2223  28CA  44 31 3A 44         DS    .BYTE   'D1:DOS.SYS',CR
2224  28CE  4F 53 2E 53
2225  28D2  59 53 9B
2226  28D5                          HILO    DS
2227  0028                  +DSH      =      DS/256
2228  00CA                  +DSL      =      (-256)*DSH+DS
2229  28D5  45 52 52 4F         WVD   .BYTE   'ERROR - NOT VERSION 2 FORMAT.',CR
2230  28D9  52 20 2D 20
2231  28DD  4E 4F 54 20
2232  28E1  56 45 52 53
2233  28E5  49 4F 4E 20
2234  28E9  32 20 46 4F
2235  28ED  52 4D 41 54
2236  28F1  2E 9B
2237  28F3                          HILO    WVD
2238  0028                  +WVDH     =      WVD/256
2239  00D5                  +WVDL     =      (-256)*WVDH+WVD
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    2240
    2241
    2242
    2243
    2244
    2245
    2246
    2247
    2248
    2249
    2250
    2251
    2252
    2253
    2254
    2255
    2256
    2257
    2258  28F3
    2259  28F3  A9 00
    2260  28F5  8D 08 1F
    2261  28F8  AD F6 1E
    2262  28FB  20 E4 2C
    2263  28FE  B0 05
    2264  2900  A9 80
    2265  2902  8D 08 1F
    2266
    2267
    2268
    2269  2905  20 26 2A
    2270  2908  AD F4 1D
    2271  290B  C9 02
    2272  290D  F0 0A
    2273
    2274
    2275
    2276  290F  A9 D5
    2277  2911  A2 28
    2278  2913  20 B5 31
    2279  2916  4C B6 20
    2280
    2281
    2282
    2283  2919  60
```

```
                .PAGE
;   **** TEST FOR VERSION 2 FORMAT - SUBROUTINE ****
;
;
;
;       SUBROUTINE - TSTVER2
;
;       READS THE DISK'S VTOC AND CHECKS IF VERSION BYTE IS SET AS 2.
;
;               ENTRY - DRIVE # STORED IN UNNO
;               EXIT  - RETURNS ONLY IF IS A VERSION 2 DISK
;                       ELSE DOES AN ERROR EXIT BACK TO MENU
;               CALLS - DRVSTAT AND RVTOC
;               CALLED BY - DELFIL, RENFIL, WBOOT.
;
;
;       GET DRIVE TYPE SO KNOW CORRECT SECTOR SIZE - NEEDED FOR RVTOC
;
TSTVER2 =       *
        LDA     #0              ;GET DRIVE TYPE IN SECSIZ
        STA     SECSIZ          ;ASSUME 256 - NEEDED BY RVTOC
        LDA     UNNO            ;GET DRIVE #
        JSR     DRVSTAT         ;FIND OUT TYPE - CARRY FLAG
        BCS     OKTYP           ;BRANCH IF 256 TYPE
        LDA     #$80            ;ELSE SET AS 128 BYTE DEVICE
        STA     SECSIZ
;
;       READ THE VTOC & CHECK IF VERSION 2
;
;
OKTYP   JSR     RVTOC           ;READ IN VTOC TO DBUF
        LDA     DBUF            ;1ST BYTE IS VERSION #
        CMP     #2              ;IS IT VERSION 2?
        BEQ     SMVRS           ;YES, SAME VERSION - RETURN
;
;       NOT A VERSION 2 DISK - PRINT MSG & GOTO MENU
;
;
        LDA     #.LOW.WVDL      ;ELSE, NOT SAME VERSION
        LDX     #.LOW.WVDH      ;PRINT INCOMPATIBLE MSG
        JSR     DSPLIN
        JMP     MENUSL          ;GOTO MENU
;
;       DISK IS VERSION TWO SO RETURN
;
SMVRS   RTS                     ;RETURN
```

```
2284                                .PAGE
2285                        ;  ****  LOAD  USER  FILE  FUNCTION  ****
2286                        ;
2287                        ;
2288  291A  5B 29           LDFIL   .WORD   LFMG
2289  291C  20 CF 30                JSR     GETIC1
2290  291F  A9 00                   LDA     #0
2291  2921  AE 9E 15                LDX     OPT
2292  2924  8D 9E 15                STA     OPT
2293  2927  E0 4E                   CPX     #'N             ;IS OPTION N FOR DON'T LOAD AND GO?
2294  2929  D0 03                   BNE     NOTN            ;BRANCH IF NOT
2295  292B  CE 9E 15                DEC     OPT
2296  292E  20 C4 30        NOTN    JSR     PERX
2297  2931  20 A9 15                JSR     LOAD
2298  2934  E0 00                   CPX     #0              ;PROCESS LOAD SUBR RESPONSE
2299  2936  F0 12                   BEQ     LDFX            ;BRANCH IF LOAD WAS OK
2300  2938  E0 03                   CPX     #3
2301  293A  F0 04                   BEQ     NLF             ;IF BAD LOAD FILE
2302  293C  98                      TYA                     ;OTHERWISE WE GOT A CIO ERROR
2303  293D  4C F6 31                JMP     CIOER           ;GO SAY WHAT IT IS
2304  2940  A9 4D            NLF     LDA     #.LOW.BLFL
2305  2942  A2 29                   LDX     #.LOW.BLFH
2306  2944  20 B5 31                JSR     DSPLIN          ;BAD LOAD FILE MSG
2307  2947  20 AA 19                JSR     CLOSX           ;CLOSE THE FILE
2308  294A  4C B6 20        LDFX    JMP     MENUSL          ;EXIT
2309  294D  42 41 44 20     BLF     .BYTE   'BAD LOAD FILE',CR
2310  2951  4C 4F 41 44
2311  2955  20 46 49 4C
2312  2959  45 9B
2313  295B                          HILO    BLF
2314  0029                  +BLFH     =     BLF/256
2315  004D                  +BLFL     =     (-256)*BLFH+BLF
2316  295B  4C 4F 41 44      LFMG    .BYTE   'LOAD FROM WHAT FILE?',CR
2317  295F  20 46 52 4F
2318  2963  4D 20 57 48
2319  2967  41 54 20 46
2320  296B  49 4C 45 3F
2321  296F  9B
```

```
2322
2323                                ;   **** LOCK & UNLOCK FILE COMMANDS ****
2324                                ;
2325                                ;
2326  2970  85 29                  LKFIL   .WORD   LKMG             ;DO LOCK
2327  2972  20 CF 30                        JSR     GETIC1
2328  2975  20 C4 30                        JSR     PERX
2329  2978  A9 23                           LDA     #LOCK
2330  297A  A2 10                           LDX     #$10
2331  297C  9D 42 03                        STA     ICCOM,X
2332  297F  20 EE 31                        JSR     CIOCL
2333  2982  4C B6 20                        JMP     MENUSL
2334  2985  57 48 41 54            LKMG    .BYTE   'WHAT FILE TO LOCK?',CR
2335  2989  20 46 49 4C
2336  298D  45 20 54 4F
2337  2991  20 4C 4F 43
2338  2995  4B 3F 9B
2339                                ;
2340  2998  AD 29                  ULFIL   .WORD   ULMG             ;DO UNLOCK
2341  299A  20 CF 30                        JSR     GETIC1
2342  299D  20 C4 30                        JSR     PERX
2343  29A0  A9 24                           LDA     #UNLOCK
2344  29A2  A2 10                           LDX     #$10
2345  29A4  9D 42 03                        STA     ICCOM,X
2346  29A7  20 EE 31                        JSR     CIOCL
2347  29AA  4C B6 20                        JMP     MENUSL
2348  29AD  57 48 41 54            ULMG    .BYTE   'WHAT FILE TO UNLOCK?',CR
2349  29B1  20 46 49 4C
2350  29B5  45 20 54 4F
2351  29B9  20 55 4E 4C
2352  29BD  4F 43 4B 3F
2353  29C1  9B
```

```
2354
2355                                ;  ****  DUPLICATE  DISK  ROUTINE  ****
2356                                ;
2357                                ;
2358  29C2  44 55 50 20             DDMG    .BYTE   'DUP DISK-SOURCE,DEST DRIVES?',CR
2359  29C6  44 49 53 4B
2360  29CA  2D 53 4F 55
2361  29CE  52 43 45 2C
2362  29D2  44 45 53 54
2363  29D6  20 44 52 49
2364  29DA  56 45 53 3F
2365  29DE  9B
2366  29DF  54 59 50 45             OK      .BYTE   'TYPE ',$22,'Y',$22,' IF OK TO USE PROGRAM AREA',CR
2367  29E3  20 22 59 22
2368  29E7  20 49 46 20
2369  29EB  4F 4B 20 54
2370  29EF  4F 20 55 53
2371  29F3  45 20 50 52
2372  29F7  4F 47 52 41
2373  29FB  4D 20 41 52
2374  29FF  45 41 9B
2375  2A02                          HILO    OK
2376  0029                    +OKH  =       OK/256
2377  00DF                    +OKL  =       (-256)*OKH+OK
2378  2A02  43 41 55 54             CMSI    .BYTE   'CAUTION: A ',$22,'Y',$22,' INVALIDATES MEM.SAV. ',CR
2379  2A06  49 4F 4E 3A
2380  2A0A  20 41 20 22
2381  2A0E  59 22 20 49
2382  2A12  4E 56 41 4C
2383  2A16  49 44 41 54
2384  2A1A  45 53 20 4D
2385  2A1E  45 4D 2E 53
2386  2A22  41 56 2E 9B
2387  2A26                          HILO    CMSI
2388  002A                    +CMSIH  =       CMSI/256
2389  0002                    +CMSIL  =       (-256)*CMSIH+CMSI
2390                                ;
2391                                ;       RVTOC READS VOLUME TABLE OF CONTENTS SECTOR
2392                                ;
2393  2A26  A9 01                   RVTOC   LDA     #1
2394  2A28  8D 0B 03                       STA     DSHI            ;READ VTOC SECTOR
2395  2A2B  A9 68                           LDA     #$68
2396  2A2D  8D 0A 03                       STA     DSLO
2397  2A30  A9 1D                           LDA     #.LOW.DBUFH
2398  2A32  8D 05 03                       STA     DBUFHI
2399  2A35  A9 F4                           LDA     #.LOW.DBUFL
2400  2A37  8D 04 03                       STA     DBUFLO          ;POINT DCB AT DBUF
2401  2A3A  20 8D 2C                       JSR     RSEC1
2402  2A3D  A9 00                           LDA     #0
2403  2A3F  8D 01 1F                       STA     PTR
2404  2A42  AD FE 1D                       LDA     DBUF+$A
2405  2A45  8D FE 1E                       STA     CSRC            ;BYTE OF ALLOC MAP
2406  2A48  A9 08                           LDA     #8
2407  2A4A  8D 02 1F                       STA     IPTR            ;COUNT BITS IN BYTE
```

```
2408  2A4D  A9 00        LDA     #0
2409  2A4F  8D 0B 03     STA     DSHI            ;POINT TO SECTOR ONE
2410  2A52  A9 01        LDA     #1
2411  2A54  8D 0A 03     STA     DSLO
2412  2A57  60           RTS
2413                    ;
2414                    ;
2415  2A58  C2 29       DUPDSK  .WORD   DDMG
2416  2A5A  A9 00               LDA     #0              ;ASSUME SINGLE DRIVE
2417  2A5C  8D 0B 1F            STA     TWODRV          ;CLEAR FLAG
2418  2A5F  20 3C 30            JSR     GETLIN
2419  2A62  20 BE 32            JSR     GETDN
2420  2A65  8D F6 1E            STA     UNNO            ;UNIT NO FOR READ
2421  2A68  20 BE 32            JSR     GETDN
2422  2A6B  8D FF 1E            STA     CDES            ;CDES IS THE DEST DRIVE #
2423  2A6E  20 C4 30            JSR     PERX
2424                    ;
2425                    ;       DETERMINE THE DRIVE TYPES OF THE SOURCE AND DESTINATION DRIVES
2426                    ;       IF THEY ARE NOT THE SAME THEN ERROR - PRINT MSG AND GOTO MENU
2427                    ;       IF THEY ARE THE SAME THEN SET SECSIZ FOR BUFFER INCREMENT IN
2428                    ;       DUP PORTION.
2429                    ;
2430  2A71  A9 80               LDA     #$80            ;ASSUME SOURCE IS 128 BYTE/SECTOR
2431  2A73  8D 08 1F            STA     SECSIZ          ;SECSIZ IN LSB,MSB ORDER
2432  2A76  A9 00               LDA     #0
2433  2A78  8D 09 1F            STA     SECSIZ+1
2434                    ;
2435  2A7B  AD F6 1E            LDA     UNNO            ;CHECK SOURCE FIRST, DO STATUS TEST
2436  2A7E  20 E4 2C            JSR     DRVSTAT         ;SETS CARRY IF 256, 128 THEN CARRY CLR
2437  2A81  90 09               BCC     ONE28           ;BRANCH IF 128 DEVICE
2438  2A83  A2 00               LDX     #0              ;ELSE SET SECSIZ AS 256 BYTES
2439  2A85  8E 08 1F            STX     SECSIZ
2440  2A88  E8                  INX
2441  2A89  8E 09 1F            STX     SECSIZ+1
2442                    ;
2443                    ;       CHECK STATUS ON DESTINATION AND SEE IF COMPATIBLE
2444                    ;
2445  2A8C  AD FF 1E    ONE28   LDA     CDES            ;DO STATUS ON DEST
2446  2A8F  20 E4 2C            JSR     DRVSTAT
2447  2A92  90 0F               BCC     IS128           ;128,YES TEST FOR 128 IN SECSIZ
2448  2A94  2C 08 1F            BIT     SECSIZ          ;ELSE CHK FOR 256 IN SECSIZ
2449  2A97  10 0F               BPL     SAME            ;BRANCH IF DES & SRC ARE 256
2450                    ;
2451                    ;       NOT THE SAME THEN PRINT MSG AND GOTO MENU.
2452                    ;
2453  2A99  A9 DE       INCOMP  LDA     #.LOW.NCDRL     ;PRINT INCOMPATIBLE DRIVE
2454  2A9B  A2 2A               LDX     #.LOW.NCDRH     ;MSG
2455  2A9D  20 B5 31            JSR     DSPLIN
2456  2AA0  4C B6 20            JMP     MENUSL          ;GOTO MENU
2457                    ;
2458                    ;       128 BYTE CHECK
2459                    ;
2460  2AA3  2C 08 1F    IS128   BIT     SECSIZ          ;IF LSB NOT 80 HEX THEN 256 SRC
2461  2AA6  10 F1               BPL     INCOMP          ;AND THEN INCOMPATIBLE
```

```
2462                            ;
2463                            ;       CHECK IF TWO DRIVE OR SINGLE DRIVE DUP
2464                            ;
2465  2AA8  AD F6 1E    SAME    LDA     UNNO
2466  2AAB  CD FF 1E            CMP     CDES            ;IF BOTH UNITS THE SAME
2467  2AAE  F0 4B               BEQ     SDD             ;SINGLE DRIVE DUP
2468  2AB0  A2 2A               LDX     #.LOW.IBDH
2469  2AB2  A9 BF               LDA     #.LOW.IBDL
2470  2AB4  20 B5 31            JSR     DSPLIN          ;PROMPT TO INSERT BOTH DISKS
2471  2AB7  20 7E 30            JSR     CHRGET
2472  2ABA  CE 0B 1F            DEC     TWODRV          ;SET TWO DRIVE FLAG
2473  2ABD  30 46               BMI     DODKDP          ;GO DUP DISK
```

```
2474                                        .PAGE
2475  2ABF  49 4E 53 45        IBD          .BYTE   'INSERT BOTH DISKS, TYPE RETURN',CR
2476  2AC3  52 54 20 42
2477  2AC7  4F 54 48 20
2478  2ACB  44 49 53 4B
2479  2ACF  53 2C 20 54
2480  2AD3  59 50 45 20
2481  2AD7  52 45 54 55
2482  2ADB  52 4E 9B
2483  2ADE                              HILO    IBD
2484  002A                     +IBDH      =       IBD/256
2485  00BF                     +IBDL      =       (-256)*IBDH+IBD
2486  2ADE  45 52 52 4F        NCDR       .BYTE   'ERROR - DRIVES INCOMPATIBLE.',CR
2487  2AE2  52 20 2D 20
2488  2AE6  44 52 49 56
2489  2AEA  45 53 20 49
2490  2AEE  4E 43 4F 4D
2491  2AF2  50 41 54 49
2492  2AF6  42 4C 45 2E
2493  2AFA  9B
2494  2AFB                              HILO    NCDR
2495  002A                     +NCDRH     =       NCDR/256
2496  00DE                     +NCDRL     =       (-256)*NCDRH+NCDR
2497                                    ;
2498                                    ;
2499                                    ;USED BY BOTH SINGLE & DOUBLE DRIVE DUP. WILL NOT ASK TO SWAP IF 2 DRIVE
2500                                    ;FLAG (TWODRV) IS SET.
2501                                    ;IF THE TWO DRIVE FLAG IS CLEAR WILL
2502                                    ;FILL FROM SOURCE DISK, SWAP, EMPTY, SWAP, REPEAT.
2503                                    ;
2504  2AFB  A9 16              SDD       LDA     #.LOW.ISDL      ;TELL USER TO INSERT SOURCE
2505  2AFD  A2 2C                        LDX     #.LOW.ISDH      ;FOR INITIAL READ - USED ONLY FOR SINGLE
2506  2AFF  20 B5 31                     JSR     DSPLIN          ;DRIVE DUPLICATE
2507  2B02  20 7E 30                     JSR     CHRGET
2508  2B05  A9 05              DODKDP    LDA     #.LOW.NMDUPL    ;SET BUFFER AT END OF DUP
2509  2B07  8D 06 1F                     STA     STVEC
2510  2B0A  A9 33                        LDA     #.LOW.NMDUPH
2511  2B0C  8D 07 1F                     STA     STVEC+1
2512                                    ;
2513                                    ;       BUFFER BOTTOM MOVES FROM NMDUP TO MEMTOP
2514                                    ;       SET END OF BUFFER TO MEMTOP MINUS 1 SECTOR IN BYTES
2515                                    ;       WHEN BUFFER BOTTOM IS LESS THAN OR EQUAL TO BUFFER END, AT
2516                                    ;       LEAST ONE MORE SECTOR WILL FIT IN MEMORY.
2517                                    ;
2518  2B0F  AD E5 02                     LDA     MEMTOP
2519  2B12  38                          SEC
2520  2B13  ED 08 1F                     SBC     SECSIZ          ;T1 IS END OF BUFFER
2521  2B16  8D 04 1F                     STA     T1
2522  2B19  AD E6 02                     LDA     MEMTOP+1
2523  2B1C  ED 09 1F                     SBC     SECSIZ+1
2524  2B1F  8D 05 1F                     STA     T1+1            ;T1 IS MEMTOP MINUS SECTOR SIZE
2525                                    ;
2526                                    ;SEE IF ROOM FOR AT LEAST ONE SECTOR!
2527                                    ;
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    2528  2B22  AD 04 1F
    2529  2B25  CD 06 1F
    2530  2B28  AD 05 1F
    2531  2B2B  ED 07 1F
    2532  2B2E  B0 0A
    2533  2B30  A9 06
    2534  2B32  A2 2C
    2535  2B34  20 B5 31
    2536  2B37  4C B6 20
    2537
    2538  2B3A  20 BE 2C
    2539  2B3D  A9 00
    2540  2B3F  8D 9E 15
    2541  2B42  20 26 2A
    2542  2B45  AD 0A 03
    2543  2B48  8D F9 1E
    2544  2B4B  AD 0B 03
    2545  2B4E  8D FA 1E
    2546  2B51  AD 01 1F
    2547  2B54  8D FB 1E
    2548  2B57  AD 02 1F
    2549  2B5A  8D FC 1E
    2550  2B5D  AD FE 1E
    2551  2B60  8D FD 1E
    2552  2B63  4C 7A 2B
    2553
    2554
    2555
    2556  2B66  A9 00
    2557  2B68  8D 9E 15
    2558  2B6B  2C 0B 1F
    2559  2B6E  30 0A
    2560  2B70  A9 16
    2561  2B72  A2 2C
    2562  2B74  20 B5 31
    2563  2B77  20 7E 30
    2564
    2565
    2566
    2567  2B7A  20 D2 2B
    2568
    2569
    2570
    2571  2B7D  20 59 2C
    2572  2B80  30 21
    2573  2B82  2C 9E 15
    2574  2B85  30 06
    2575  2B87  20 8D 2C
    2576  2B8A  4C 90 2B
    2577  2B8D  20 98 2C
    2578  2B90  AD 04 03
    2579  2B93  18
    2580  2B94  6D 08 1F
    2581  2B97  8D 04 03
```

```
        LDA     T1              ;DO DOUBLE PRECISION TEST
        CMP     STVEC           ;TO SEE IF ROOM
        LDA     T1+1            ;IF T1 IS = STVEC THEN ENUF ROOM
        SBC     STVEC+1         ;FOR ONE SECTOR
        BCS     ENUF            ;BRANCH IF (T1)>=(STVEC)
NORM    LDA     #NRML
        LDX     #NRMH
        JSR     DSPLIN
        JMP     MENUSL
;
ENUF    JSR     CKMEM           ;SEE IF OK TO USE USER AREA
        LDA     #0
        STA     OPT             ;SET UP FOR READ HERE FIRST PASS
        JSR     RVTOC           ;READ VTOC
        LDA     DSLO            ;COPY INITIAL WRITE POINTERS
        STA     SWDP            ;TO INITIAL READ POINTERS
        LDA     DSHI
        STA     SWDP+1
        LDA     PTR
        STA     SWDP+2
        LDA     IPTR
        STA     SWDP+3
        LDA     CSRC
        STA     SWDP+4
        JMP     LRS1            ;SKIP FIRST READ PROMPT
;
;READ FROM SOURCE DISK TIL BUF FULL OR END OF DATA.
;
DORD    LDA     #0              ;FLAG WE ARE READING
        STA     OPT
        BIT     TWODRV          ;TEST FOR 2 DRIVES
        BMI     LRS1            ;YES, SKIP THE SWAP
        LDA     #.LOW. ISDL     ;INSERT SRC DISK
        LDX     #.LOW. ISDH
XBLK    JSR     DSPLIN
        JSR     CHRGET
;
;SWAP POINTERS TO WHERE WE ARE
;
LRS1    JSR     DOSWDP          ;SWAP SECTOR AND BITMAP POINTERS
;
;LOOP READING/WRITING SECTORS TO BUFFER AREA
;
LRS     JSR     AAM             ;ADVANCE ALLOCATION MAP
        BMI     ASPT            ;IF FREE, ADV SECTR POINTER & TRY AGIN
        BIT     OPT             ;SEE WHAT MODE
        BMI     DOW             ;BR IF WRITE
        JSR     RSEC1           ;DO READ
        JMP     IOD
DOW     JSR     DKWRT           ;DO WRITE
IOD     LDA     DBUFLO          ;ADVANCE BUFFER POINTER
        CLC
        ADC     SECSIZ          ;ADD SECTOR SIZE TO BOTTOM OF BUFFER
        STA     DBUFLO          ;SO POINT TO NEXT FREE BLOCK
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    2582  2B9A  AD 05 03          LDA     DBUFHI
    2583  2B9D  6D 09 1F          ADC     SECSIZ+1
    2584  2BA0  8D 05 03          STA     DBUFHI
    2585  2BA3  20 76 2C  ASPT    JSR     ASP             ;GO ADVANCE SECTOR POINTER
    2586  2BA6  F0 22             BEQ     STDD1           ;ALL SECTORS DONE, SWAP TO DEST DISK
    2587  2BA8  AD 04 1F          LDA     T1              ;SEE IF ROOM FOR ANOTHER
    2588  2BAB  CD 04 03          CMP     DBUFLO          ;SECTOR BELOW MEMTOP
    2589  2BAE  AD 05 1F          LDA     T1+1
    2590  2BB1  ED 05 03          SBC     DBUFHI
    2591  2BB4  B0 C7             BCS     LRS             ;BRANCH IF (DBUF)<=(T1)  - ROOM FOR MORE
    2592                        ;
    2593                        ;SWAP DISKS AND CONTINUE
    2594                        ;
    2595  2BB6  AD 9E 15  STDD    LDA     OPT
    2596  2BB9  30 AB             BMI     DORD            ;IF WAS WRITE, GO READ
    2597  2BBB  CE 9E 15  STDD2   DEC     OPT             ;CHANGE TO WRITE
    2598  2BBE  2C 0B 1F          BIT     TWODRV          ;ARE 2 DRIVES BEING USED?
    2599  2BC1  30 B7             BMI     LRS1            ;YES, SKIP THE SWAP
    2600  2BC3  A9 35             LDA     #.LOW. IDDL     ;INSERT DEST DISK
    2601  2BC5  A2 2C             LDX     #.LOW. IDDH
    2602  2BC7  4C 74 2B          JMP     XBLK            ;GO DO WRITE
    2603  2BCA  AD 9E 15  STDD1   LDA     OPT             ;END OF DATA
    2604  2BCD  10 EC             BPL     STDD2           ;IF READ GO WRITE
    2605  2BCF  4C B6 20          JMP     MENUSL          ;IF WRITE WE ARE DONE
    2606                        ;
    2607                        ;DOSWDP - EXCHANGE CURRENT AND SAVED BITMAP & SECTOR POINTERS
    2608                        ;ALSO INIT BUFFER POINTER
    2609                        ;
    2610  2BD2  A0 04   DOSWDP  LDY     #4
    2611  2BD4  B9 FC 2B SWLOP   LDA     SWATL,Y
    2612  2BD7  85 1A             STA     RAMLO
    2613  2BD9  B9 01 2C          LDA     SWATH,Y
    2614  2BDC  85 1B             STA     RAMLO+1         ;GET ADDRESS FROM TABLE TO RAMLO
    2615  2BDE  A2 00             LDX     #0
    2616  2BE0  A1 1A             LDA     (RAMLO,X)       ;GET WHAT'S THERE
    2617  2BE2  48                PHA
    2618  2BE3  B9 F9 1E          LDA     SWDP,Y
    2619  2BE6  81 1A             STA     (RAMLO,X)
    2620  2BE8  68                PLA
    2621  2BE9  99 F9 1E          STA     SWDP,Y
    2622  2BEC  88                DEY
    2623  2BED  10 E5             BPL     SWLOP
    2624  2BEF  AD 06 1F          LDA     STVEC
    2625  2BF2  8D 04 03          STA     DBUFLO
    2626  2BF5  AD 07 1F          LDA     STVEC+1
    2627  2BF8  8D 05 03          STA     DBUFHI
    2628  2BFB  60                RTS
    2629                        ;
    2630                        ;       WHAT A MESS
    2631                        ;
    2632  2BFC                    HILO    DSLO
    2633  0003              +DSLOH  =       DSLO/256
    2634  000A              +DSLOL  =       (-256)*DSLOH+DSLO
    2635  2BFC                    HILO    DSHI
```

```
ERR LINE   ADDR  B1 B2 B3 B4

    2636   0003
    2637   000B
    2638   2BFC
    2639   001F
    2640   0001
    2641   2BFC
    2642   001F
    2643   0002
    2644   2BFC
    2645   001E
    2646   00FE

+DSHIH      =       DSHI/256
+DSHIL      =       (-256)*DSHIH+DSHI
            HILO    PTR
+PTRH       =       PTR/256
+PTRL       =       (-256)*PTRH+PTR
            HILO    IPTR
+IPTRH      =       IPTR/256
+IPTRL      =       (-256)*IPTRH+IPTR
            HILO    CSRC
+CSRCH      =       CSRC/256
+CSRCL      =       (-256)*CSRCH+CSRC
```

```
2647                                        .PAGE
2648  2BFC  0A 0B 01 02    SWATL   .BYTE   DSLOL,DSHIL,PTRL,IPTRL,CSRCL
2649  2C00  FE
2650  2C01  03 03 1F 1F    SWATH   .BYTE   DSLOH,DSHIH,PTRH,IPTRH,CSRCH
2651  2C05  1E
2652                        ;
2653                        ;
2654  2C06  4E 4F 54 20    NRM     .BYTE   'NOT ENOUGH ROOM',CR
2655  2C0A  45 4E 4F 55
2656  2C0E  47 48 20 52
2657  2C12  4F 4F 4D 9B
2658  2C16  49 4E 53 45    ISD     .BYTE   'INSERT SOURCE DISK,TYPE RETURN',CR
2659  2C1A  52 54 20 53
2660  2C1E  4F 55 52 43
2661  2C22  45 20 44 49
2662  2C26  53 4B 2C 54
2663  2C2A  59 50 45 20
2664  2C2E  52 45 54 55
2665  2C32  52 4E 9B
2666  2C35  49 4E 53 45    IDD     .BYTE   'INSERT DESTINATION DISK,TYPE RETURN',CR
2667  2C39  52 54 20 44
2668  2C3D  45 53 54 49
2669  2C41  4E 41 54 49
2670  2C45  4F 4E 20 44
2671  2C49  49 53 4B 2C
2672  2C4D  54 59 50 45
2673  2C51  20 52 45 54
2674  2C55  55 52 4E 9B
2675  2C59                        HILO    NRM
2676  002C                +NRMH   =       NRM/256
2677  0006                +NRML   =       (-256)*NRMH+NRM
2678  2C59                        HILO    ISD
2679  002C                +ISDH   =       ISD/256
2680  0016                +ISDL   =       (-256)*ISDH+ISD
2681  2C59                        HILO    IDD
2682  002C                +IDDH   =       IDD/256
2683  0035                +IDDL   =       (-256)*IDDH+IDD
2684                        ;
2685                        ;
2686                        ; AAM - ADVANCE ALLOCATION MAP ONE BIT.
2687                        ; RETURN MINUS IF FREE.
2688                        ;
2689  2C59  0E FE 1E        AAM     ASL     CSRC            ;NEXT BIT OF ALLOC MAP
2690  2C5C  CE 02 1F                DEC     IPTR
2691  2C5F  D0 11                   BNE     CBIT            ;IF DONE WITH THIS BYTE
2692  2C61  EE 01 1F                INC     PTR             ;GET NEXT ONE
2693  2C64  AE 01 1F                LDX     PTR
2694  2C67  BD FE 1D                LDA     DBUF+$A,X       ;VTOC IS DBUF & BITMAP STRTS IN 10TH BYT
2695  2C6A  8D FE 1E                STA     CSRC
2696  2C6D  A9 08                   LDA     #8
2697  2C6F  8D 02 1F                STA     IPTR
2698  2C72  AD FE 1E        CBIT    LDA     CSRC            ;CHECK THE BIT
2699  2C75  60                      RTS
2700                        ;
```

```
2701
2702                    ;
2703                    ; ASP - ADVANCE SECTOR POINTER IN DCB.
2704  2C76  AD 0A 03    ; RETURN EQ IF AT END.
2705  2C79  C9 D0       ASP     LDA     DSLO            ;SEE IF END
2706  2C7B  D0 07               CMP     #208
2707  2C7D  AD 0B 03            BNE     NXS
2708  2C80  C9 02               LDA     DSHI
2709  2C82  F0 08               CMP     #2
2710  2C84  EE 0A 03            BEQ     ASPX            ;ALL DONE
2711  2C87  D0 03       NXS     INC     DSLO
2712  2C89  EE 0B 03            BNE     ASPX
2713  2C8C  60                  INC     DSHI
2714                    ASPX    RTS
2715                    ;
2716                    ; RSEC1 - READ A SECTOR WHOSE NUMBER IS IN DCB
2717  2C8D  AD F6 1E    ;
2718  2C90  8D 01 03    RSEC1   LDA     UNNO
2719  2C93  18                  STA     DUNIT
2720  2C94  08                  CLC                     ;TELL DISK HANDLER DOING A GET SECTOR
2721  2C95  4C A0 2C            PHP                     ;SAVE FLAG
2722                            JMP     CLDKH
2723                    ;
2724                    ; DKWRT - WRITE A SECTOR
2725  2C98  AD FF 1E    ;
2726  2C9B  8D 01 03    DKWRT   LDA     CDES            ;PUT DEST UNIT #
2727  2C9E  38                  STA     DUNIT           ;IN DCB
2728  2C9F  08                  SEC                     ;TELL DISK HANDLER DOING WRITE SECTOR
2729  2CA0  A9 02               PHP                     ;SAVE FLAG
2730  2CA2  8D F7 1E    CLDKH   LDA     #2              ;SET RETRY COUNT
2731  2CA5  A2 01               STA     RCNT
2732  2CA7  2C 08 1F    CLD1    LDX     #1              ;SET DRIVE TYPE- ASSUME 128
2733  2CAA  30 01               BIT     SECSIZ          ;TEST FOR 128
2734  2CAC  E8                  BMI     NOT256          ;IF IS BRANCH
2735  2CAD  28                  INX                     ;ELSE SET FOR 256
2736  2CAE  08          NOT256  PLP
2737  2CAF  20 72 07            PHP                     ;SET ACTION FLAG & SAVE IT FOR RETRY
2738  2CB2  10 08               JSR     BSIOR           ;GOTO FMS DISK HANDLER
2739                            BPL     DRTS            ;RETURN IF GOOD STATUS
2740  2CB4  CE F7 1E    ;
2741  2CB7  10 EC               DEC     RCNT            ;ELSE SEE IF MORE RETRIES
2742  2CB9  4C F5 31            BPL     CLD1            ;YES, DO AGAIN
2743  2CBC  28                  JMP     CIOER1          ;CIO ERROR, GO SAY WHICH
2744  2CBD  60          DRTS    PLP                     ;EVEN OUT STACK
2745                            RTS                     ;RETURN
2746                    ;
2747                    ;       CKMEM - ASK IF OK TO USE USER AREA
2748  2CBE  A5 08       ;
2749  2CC0  F0 1C       CKMEM   LDA     WARMST          ;IF MEMORY WAS INTACT
2750  2CC2  A9 DF               BEQ     CPTR1           ;QUERY TO BOMB IT
2751  2CC4  A2 29               LDA     #.LOW.OKL
2752  2CC6  20 B5 31            LDX     #.LOW.OKH       ;PRINT PROMPT
2753  2CC9  A9 02               JSR     DSPLIN
2754  2CCB  A2 2A               LDA     #.LOW.CMSIL     ;PRINT CAUTION MSG
                                LDX     #.LOW.CMSIH     ;Y RESPONSE WILL INVALIDATE MEM.SAV
```

```
2755  2CCD  20 B5 31          JSR     DSPLIN
2756  2CD0  20 7E 30          JSR     CHRGET
2757  2CD3  C9 59             CMP     #'Y             ;TEST FOR OK TO BOMB USER AREA
2758  2CD5  D0 08             BNE     DDXT            ;IF SAY NO THEN DON'T DO DUP
2759  2CD7  A9 00             LDA     #0
2760  2CD9  85 08             STA     WARMST          ;TELL CART NO GOOD USER MEMORY
2761  2CDB  8D 9E 17          STA     MEMFLG          ;TELL LOADER NO GOOD MEM SAV
2762  2CDE  60          CPTR1 RTS
2763                     ;
2764  2CDF  68           DDXT  PLA                     ;POP RETURN ADDRESS
2765  2CE0  68                 PLA
2766  2CE1  4C B6 20           JMP     MENUSL          ;GOTO MENU, DON'T DO DUP
2767                     ;
2768                     ;
2769                     ;       DRVSTAT - SUBROUTINE TO DO STATUS ON DISK DRIVE SPECIFIED
2770                     ;                 BY THE NUMBER IN REG. A.
2771                     ;       RETURNS - CARRY SET = DEVICE HAS 256 BYTE SECTORS
2772                     ;                 CARRY CLR = DEVICE HAS 128 BYTE SECTORS
2773                     ;
2774  2CE4  8D 01 03     DRVSTAT STA     DUNIT           ;STORE UNIT NUMBER IN DCB
2775  2CE7  A9 53             LDA     #STAREQ         ;STORE STATUS COMMAND IN DCB
2776  2CE9  8D 02 03          STA     DCOMND
2777  2CEC  A9 02             LDA     #2              ;SET RETRY COUNT
2778  2CEE  8D F7 1E          STA     RCNT
2779  2CF1  20 53 E4     DOSTAT  JSR     DKHND           ;DO STATUS WITH OS HANDLER
2780  2CF4  10 08             BPL     CHKTYP          ;IF GOOD RETURN, DETERMINE TYPE
2781                     ;
2782  2CF6  CE F7 1E          DEC     RCNT            ;ELSE SEE IF ANOTHER RETRY
2783  2CF9  10 F6             BPL     DOSTAT          ;YES, DO AGAIN
2784  2CFB  4C F5 31          JMP     CIOER1          ;ELSE ERROR EXIT
2785                     ;
2786  2CFE  18           CHKTYP  CLC                   ;ASSUME 128 BYTE DEVICE
2787  2CFF  AD EA 02           LDA     DVSTAT          ;GET COMMAND STATUS BYTE
2788  2D02  29 20              AND     #$20            ;MASK FOR DRIVE TYPE BIT- D5
2789  2D04  F0 01              BEQ     RETSTAT         ;128 IF = 0
2790  2D06  38                 SEC                     ;256 IF = 1
2791  2D07  60           RETSTAT RTS
2792                     ;   ****  DUPLICATE  FILE  COMMAND  ****
2793                     ;
2794                     ;
2795                     ;       DUPLICATE FILE FROM ONE DISK TO ANOTHER
2796                     ;USING ONE DRIVE.  FILENAME FOR DUPLICATE FILE IS SAME AS
2797                     ;SOURCE NAME.  USER CAN ENTER ONLY THE SOURCE FILE SPECIFICATION.
2798                     ;USER HAS OPTION OF USING PROGRAM AREA FOR COPY OR A 250 BYTE
2799                     ;DATA BUFFER TO GET PROGRAM AREA USER MUST RESPOND WITH
2800                     ;'Y' AS 1ST CHAR OR THEY WILL GET THE DATA BUFFER.  WILL DUPLICATE
2801                     ;FILE OF ANY SIZE.  IF ERROR, PRINTS MSG, CLOSES FILE(S) OPEN AND
2802                     ;RETURNS TO MENU.   TO PREVENT POSSIBLE DAMAGE TO DESTINATION
2803                     ;DISK, FILE IS OPENED AND CLOSED FOR EACH WRITE.
2804                     ;MAKES BUFR LEN AN EVEN MULTIPLE OF 125.   THIS PREVENTS FRAGMENTATION
2805                     ;OF THE FILE DUE TO THE APPEND OPEN FUNC.   125 IS USED BECAUSE IT IS THE
2806                     ;SIZE OF DATA PORTION IN A SECTOR.   IF THIS CHANGES THE VALUE IN THE PGM
2807                     ;MUST BE CHANGED.
2808                     ;       K.B.  5/7/80
```

```
2809                                ;
2810  2D08  4E 41 4D 45            DPFM    .BYTE   'NAME OF FILE TO MOVE?',CR
2811  2D0C  20 4F 46 20
2812  2D10  46 49 4C 45
2813  2D14  20 54 4F 20
2814  2D18  4D 4F 56 45
2815  2D1C  3F 9B
2816                                ;
2817  2D1E  08 2D                   DUPFIL  .WORD   DPFM             ;DUPLICATE FILE PROMPT
2818  2D20  20 CF 30                        JSR     GETIC1          ;GET FILENAME TO DUPLICATE ON SAME DRIVE
2819  2D23  20 C4 30                        JSR     PERX            ;DON'T COME BACK IF PARAMETER ERRORS
2820  2D26  AD 7C 1D                        LDA     PAR
2821  2D29  C9 44                           CMP     #'D             ;DUPLICATE FILE ONLY FOR DISK DEVICE
2822  2D2B  F0 03                           BEQ     ISDISK
2823  2D2D  4C 74 25                        JMP     ODMS            ;IF NOT -- SAY CANNOT DO & EXIT
2824                                ;
2825  2D30  20 41 2E                ISDISK  JSR     USEPGM          ;ASK USER IF TO USE PROG AREA OR BUFFER
2826                                ;
2827                                ;       HAVE USER INSERT SOURCE FILE AND HIT <CR> WHEN DONE
2828                                ;
2829  2D33  A2 2C                           LDX     #.LOW.ISDH      ;ARG: LINE TO BE DISPLAYED ADDR
2830  2D35  A9 16                           LDA     #.LOW.ISDL      ;IN REG. A & X
2831  2D37  20 B5 31                        JSR     DSPLIN          ;PRINT INSERT SOURCE MSG
2832  2D3A  20 3C 30                        JSR     GETLIN          ;GOTO SCREEN & WAIT FOR <CR>
2833  2D3D  20 C4 30                        JSR     PERX            ;GOTO MENU IF BREAK KEY HIT
2834                                ;
2835  2D40  20 D7 2E                        JSR     LOOKWC          ;SEE IF FILE SPEC. USES WILDCARDS
2836  2D43  D0 05                           BNE     NOWC            ;BRANCH IF NO WILD CARDS USED - USE OLD
2837  2D45  A9 40                           LDA     #$40            ;SET 'DUPLICATE WILDCARD' MODE
2838  2D47  4C 96 23                        JMP     WCINIT          ;OPEN WILDCARD DIRECTORY FILE, ETC.
2839                                ;
2840  2D4A                          NOWC    =       *
2841                                ;
2842                                ;       MAKE SURE DEST NOT DOS.SYS
2843                                ;
2844  2D4A  A2 00                           LDX     #0              ;ENTRY-INDEX TO FIRST CHAR OF FILE NAME
2845  2D4C  20 ED 2E                        JSR     TSTDOS          ;WON'T RETURN IF IS DOS.SYS
2846                                ;
2847                                ;       OPEN SOURCE FILE - ADDR OF FILENAME STRING IN PARAM LIST IS
2848                                ;       ALREADY ASSIGNED TO IOCB # 2
2849                                ;
2850  2D4F  A2 10                   WCDUPS  LDX     #$10            ;USE IOCB #2
2851  2D51  A9 03                           LDA     #OPEN           ;OPEN COMMAND
2852  2D53  9D 42 03                        STA     ICCOM,X
2853  2D56  A9 04                           LDA     #4              ;READ ONLY
2854  2D58  9D 4A 03                        STA     ICAX1,X
2855  2D5B  20 EE 31                        JSR     CIOCL           ;CALL CIO - IF ERR PRNT MSG,CLOSE, GOTO
2856                                ;
2857                                ; EOFFLG - SOURCE FILE EOF FLAG   FTRF - FLAG TO SHOW IF 1ST TIME SOURCE
2858                                ; FILE WAS READ
2859                                ;
2860  2D5E  A9 00                           LDA     #0
2861  2D60  8D 0A 1F                        STA     EOFFLG          ;CLEAR EOF FLAG
2862  2D63  8D 0B 1F                        STA     FTRF            ;CLEAR MEANS FIRST TIME
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    2863
    2864
    2865
    2866
    2867
    2868  2D66  A2 10
    2869  2D68  A5 1A
    2870  2D6A  9D 44 03
    2871  2D6D  A5 1B
    2872  2D6F  9D 45 03
    2873  2D72  AD 04 1F
    2874  2D75  9D 48 03
    2875  2D78  AD 05 1F
    2876  2D7B  9D 49 03
    2877  2D7E  A9 07
    2878  2D80  9D 42 03
    2879  2D83  20 56 E4
    2880
    2881
    2882
    2883
    2884  2D86  10 0A
    2885  2D88  C0 88
    2886  2D8A  F0 03
    2887  2D8C  4C F5 31
    2888  2D8F  CE 0A 1F
    2889
    2890
    2891
    2892
    2893  2D92  A2 2C
    2894  2D94  A9 35
    2895  2D96  20 B5 31
    2896  2D99  20 3C 30
    2897  2D9C  2C F5 1E
    2898  2D9F  10 03
    2899  2DA1  4C 1F 2E
    2900
    2901
    2902
    2903
    2904  2DA4  A2 20
    2905  2DA6  A0 09
    2906  2DA8  AD 0B 1F
    2907  2DAB  D0 05
    2908  2DAD  A0 08
    2909  2DAF  EE 0B 1F
    2910
    2911  2DB2  98
    2912  2DB3  9D 4A 03
    2913  2DB6  A9 03
    2914  2DB8  9D 42 03
    2915
    2916
```

```
;
;   DO UNTIL (SOURCE EOF FLAG (EOFFLG) IS SET)
;   SET UP IOCB#2 TO DO GET CHAR.  ZP LOC BUFADR HAS BUFFER ADDRESS
;   BUFLEN HAS BUFFER LENGTH
;
DODUP   LDX     #$10            ;USE IOCB #2
        LDA     BUFADR          ; IN LSB,MSB ORDER
        STA     ICBAL,X         ;SET BUFFER ADDR IN IOCB #2
        LDA     BUFADR+1
        STA     ICBAH,X
        LDA     BUFLEN          ; IN LSB,MSB ORDER
        STA     ICBLL,X         ;STORE BUFFER LENGTH
        LDA     BUFLEN+1        ; IN IOCB #2
        STA     ICBLH,X
        LDA     #GETCHR         ;COMMAND TO GET CHAR - IGNORE EOL'S (9B)
        STA     ICCOM,X
        JSR     CIO             ;CALL CIO
;
;       CHECK FOR ENDFILE.  IF YES, THEN SET FLG.  CHECK FOR ERR  IF ERR
;       THEN PRINT MSG, CLOSE FILE, AND RETURN TO MENU
;
        BPL     INSDES          ;IF GOOD READ WRITE BUFFER
        CPY     #EOF            ;WAS IT EOF?
        BEQ     SETFLG          ;YES, THEN SET FLAG
        JMP     CIOER1          ;WAS ERR - PRINT MSG,CLOSE,GOTO MENU
SETFLG  DEC     EOFFLG          ;SET ENDFILE FLAG
;
;       WHEN GOOD READ OR EOF GET HERE.  ASK USER TO INSERT DESTINATION
;       DISK AND ATTEMPT TO WRITE TO DESTINATION FILE.
;
;
INSDES  LDX     #.LOW.IDDH      ;ARG:  ADDRESS OF LINE TO BE PRINTED
        LDA     #.LOW.IDDL      ;IN REGS A AND X
        JSR     DSPLIN          ;SAY TO SWAP DISKS
        JSR     GETLIN          ;WAIT TIL USER HITS <CR>
        BIT     PER             ;WAS BREAK KEY HIT?
        BPL     DODEST          ;NO, TRY WRITE
        JMP     CLSSRC          ;YES, CLOSE & GOTO MENU
;
;       CHECK IF FIRST TIME SRC WAS READ.  IF YES, THEN OPEN FOR OUTPUT
;       ONLY.  OTHERWISE, OPEN FOR OUTPUT APPEND.
;
;
DODEST  LDX     #$20            ;USE IOCB #3 FOR DESTINATION
        LDY     #9              ;ASSUME APPEND
        LDA     FTRF            ;IS FLAG CLEAR?
        BNE     OPNDES          ;NO,NOT FIRST TIME - OPEN APPEND
        LDY     #8              ;YES, THEN OPEN OUT ONLY
        INC     FTRF            ;SET TO SHOW NOT FIRST TIME NEXT TIME
;
OPNDES  TYA                     ;GET OPEN TYPE CODE
        STA     ICAX1,X         ;SET AUX1 BYTE
        LDA     #OPEN           ;OPEN COMMAND
        STA     ICCOM,X
;
;         THE FILENAME IS THE FIRST FILE IN THE PARAMETER LIST-PAR.
;
```

```
ERR LINE  ADDR  B1 B2 B3 B4

2917
2918  2DBB  A9 7C
2919  2DBD  A0 1D
2920  2DBF  2C 41 23
2921  2DC2  50 04
2922
2923  2DC4  A9 64
2924  2DC6  A0 23
2925
2926  2DC8  9D 44 03
2927  2DCB  98
2928  2DCC  9D 45 03
2929  2DCF  20 EE 31
2930
2931
2932
2933
2934  2DD2  A0 10
2935  2DD4  A2 20
2936  2DD6  A9 00
2937  2DD8  D9 48 03
2938  2DDB  D0 05
2939  2DDD  D9 49 03
2940  2DE0  F0 1E
2941
2942  2DE2  A9 0B
2943  2DE4  9D 42 03
2944  2DE7  A5 1A
2945  2DE9  9D 44 03
2946  2DEC  A5 1B
2947  2DEE  9D 45 03
2948  2DF1  B9 48 03
2949  2DF4  9D 48 03
2950  2DF7  B9 49 03
2951  2DFA  9D 49 03
2952  2DFD  20 EE 31
2953
2954
2955
2956  2E00  A9 0C
2957  2E02  9D 42 03
2958  2E05  20 EE 31
2959
2960
2961
2962
2963  2E08  AD 0A 1F
2964  2E0B  D0 12
2965
2966
2967
2968  2E0D  A2 2C
2969  2E0F  A9 16
2970  2E11  20 B5 31
```

```
;
        LDA     #PARL           ;SET BUFR ADDR TO FILE SPEC TO BE OPENED
        LDY     #PARH
        BIT     WCFLAG          ;IF WLDCARD-WLDCARD BUFR INSTEAD OF PAR
        BVC     SKIPWC
;
        LDA     #.LOW.WCBUF2
        LDY     #.HIGH.WCBUF2
;
SKIPWC  STA     ICBAL,X
        TYA
        STA     ICBAH,X
        JSR     CIOCL           ;CALL CIO, IF ERROR GOTO MENU
;
;       CHECK IF SOURCE BUFR LEN IS NOT EQUAL TO ZERO.  IF NOT = ZERO
;       THEN WRITE BUFFER TO THE DESTINATION FILE.
;
        LDY     #$10            ;SOURCE IS AT IOCB #2
        LDX     #$20            ;DEST IS AT IOCB #3
        LDA     #0              ;CHECK LENGTH LOW FOR ZERO
        CMP     ICBLL,Y         ;LOW=0
        BNE     DOWRIT          ;NO THEN WRITE BUFFER
        CMP     ICBLH,Y         ;IS HI=0?
        BEQ     CLSDES          ;YES, DON'T WRITE EMPTY BUFFER
;
DOWRIT  LDA     #PUTCHR         ;PUT CHAR COMMAND CODE
        STA     ICCOM,X         ;IGNORE EOLS (9B)
        LDA     BUFADR          ;GET BUFFER ADDRESS
        STA     ICBAL,X
        LDA     BUFADR+1
        STA     ICBAH,X
        LDA     ICBLL,Y         ;GET BUFFER LENGTH TO WRITE
        STA     ICBLL,X         ;FROM IOCB OF SOURCE FILE
        LDA     ICBLH,Y         ;SET BY GET TO ACTUAL BYTE
        STA     ICBLH,X         ;COUNT READ INTO BUFFER
        JSR     CIOCL           ;DO WRITE - IF ERR GOTO MENU
;
;         CLOSE DESTINATION FILE
;
CLSDES  LDA     #CLOSE          ;CLOSE COMMAND CODE
        STA     ICCOM,X         ;CALL CIO -  IF ERROR GOTO
        JSR     CIOCL           ;MENU AFTER PRINT MSG
;
;       TEST ENDFILE FLAG.  IF IT IS SET THEN COMPLETED DUPLICATION
;       OTHERWISE, DO LOOP BODY AGAIN (READ THEN WRITE).
;
        LDA     EOFFLG          ;IS SOURCE AT ENDFILE?
        BNE     CLSSRC          ;YES, THEN DONE
;
;       ASK USER TO INSERT SOURCE FOR NEXT READ & THEN REPEAT LOOP
;
        LDX     #.LOW.ISDH      ;ARGS:  ADDRESS OF LINE TO PRINT IN
        LDA     #.LOW.ISDL      ;REGS A AND X
        JSR     DSPLIN          ;SAY TO INSERT SOURCE
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    2971  2E14  20 3C 30
    2972  2E17  2C F5 1E
    2973  2E1A  30 03
    2974  2E1C  4C 66 2D
    2975
    2976
    2977
    2978
    2979
    2980  2E1F  A2 10
    2981  2E21  A9 0C
    2982  2E23  9D 42 03
    2983  2E26  20 56 E4
    2984
    2985  2E29  2C 41 23
    2986  2E2C  50 10
    2987  2E2E  A2 2C
    2988  2E30  A9 16
    2989  2E32  20 B5 31
    2990  2E35  20 3C 30
    2991  2E38  20 C4 30
    2992  2E3B  4C 9E 23
    2993  2E3E
    2994
    2995  2E3E  4C B6 20
```

```
                JSR     GETLIN          ;WAIT TIL USER HITS <CR>
                BIT     PER             ;WAS BREAK KEY HIT?
                BMI     CLSSRC          ;YES, CLOSE & GOTO MENU
                JMP     DODUP           ;REPEAT LOOP
;
;               ******************END OF LOOP******************
;
;
;                CLOSE SOURCE AND RETURN TO MENU
;
CLSSRC  LDX     #$10                    ;SOURCE AT IOCB #2
        LDA     #CLOSE                  ;CLOSE COMMAND CODE
        STA     ICCOM,X
        JSR     CIO                     ;CALL CIO
;
        BIT     WCFLAG                  ;TEST IF 'DUPLICATE WILDCARD' MODE
        BVC     DUPFEX                  ;BR IF NOT 'DUPLICATE WILDCARD' MODE
        LDX     #.LOW. ISDH             ;INSERT SOURCE MESSAGE
        LDA     #.LOW. ISDL
        JSR     DSPLIN                  ;NEEDED TO GET NEXT WILDCARD DIR ENTRY
        JSR     GETLIN                  ;WAIT FOR CR
        JSR     PERX                    ;IF BREAK-KEY ABORT - EXIT TO MENU
        JMP     WCOPYL                  ;JUMP TO WILDCARD LOOP
DUPFEX  =       *
;
        JMP     MENUSL                  ;GO TO THE MENU
```

```
2996                                .PAGE
2997                        ;   ****  ASK  IF  OK  TO  USE  PROGRAM  AREA  ROUTINE  ****
2998                        ;
2999                        ;ASK USER IF CAN USE PROGRAM AREA.  IF SAY YES ('Y') THEN
3000                        ;ASSIGN BUFFER ADDRESS AS ALL AVAILABLE MEMORY.  OTHERWISE, USE
3001                        ;DBUF (250 BYTES) AS THE BUFFER.   ASSIGNS BUFFER LENGTH.
3002                        ;
3003                        ;NO PARAMETERS
3004                        ;RETURNS:  BUFADR-BUFFER ADDRESS
3005                        ;          BUFLEN-BUFFER LENGTH
3006                        ;
3007 2E41  A5 08    USEPGM  LDA     WARMST          ;CHECK IF PGM AREA ALREADY
3008 2E43  F0 15            BEQ     USEDB4          ;USED-YES, USE IT AGAIN
3009 2E45  A9 DF            LDA     #.LOW.OKL       ;ARGS:  IN A AND X ADDR
3010 2E47  A2 29            LDX     #.LOW.OKH       ;OF LINE TO DISPLAY
3011 2E49  20 B5 31         JSR     DSPLIN          ;ASK TO USE PGM AREA
3012 2E4C  A9 02            LDA     #.LOW.CMSIL     ;SAY A Y RESPONSE WILL
3013 2E4E  A2 2A            LDX     #.LOW.CMSIH     ;INVALIDATE MEM.SAV
3014 2E50  20 B5 31         JSR     DSPLIN          ;PRINT CAUTION
3015 2E53  20 7E 30         JSR     CHRGET          ;GET 1ST CHAR OF
3016 2E56  C9 59            CMP     #'Y             ;USERS RESPONSE
3017 2E58  D0 6A            BNE     USEBUF          ;NO, THEN USE DBUFF
3018                        ;
3019                        ;USE ALL MEMORY AVAILABLE-PROGRAM AREA
3020                        ;MEMLO,MEMTOP,BUFADR,BUFLEN ARE IN LSB,MSB FORM
3021                        ;
3022 2E5A  A9 00    USEDB4  LDA     #0              ;CLEAR WARMSTART FLAG
3023 2E5C  85 08            STA     WARMST          ;TO SHOW PGM AREA USED
3024 2E5E  8D 9E 17         STA     MEMFLG          ;SHOW NO USER AREA GOOD-MEM.SAV ALSO
3025 2E61  A9 05            LDA     #.LOW.NMDUPL    ;USE ALL AVAILABLE
3026 2E63  85 1A            STA     BUFADR          ;MEMORY-FROM END OF DUP TO MEMTOP
3027 2E65  A9 33            LDA     #.LOW.NMDUPH    ;BUFADR HAS BUFFER
3028 2E67  85 1B            STA     BUFADR+1        ;ADDRESS
3029 2E69  AD E5 02         LDA     MEMTOP          ;GET LENGTH OF
3030 2E6C  38               SEC                     ;PGM AREA
3031 2E6D  E9 05            SBC     #.LOW.NMDUPL
3032 2E6F  8D 04 1F         STA     BUFLEN          ;LSB,MSB ORDER
3033 2E72  AD E6 02         LDA     MEMTOP+1
3034 2E75  E9 33            SBC     #.LOW.NMDUPH
3035 2E77  8D 05 1F         STA     BUFLEN+1
3036                        ;
3037                        ;       FIND THE GREATEST MULTIPLE OF 125 LESS THAN THE PROGRAM AREA
3038                        ;       THEN SET BUFR LEN TO IT.  THIS PREVENTS FRAGMENTATION TO FILE
3039                        ;       WHEN APPEND IS USED IN DUPFIL.
3040                        ;
3041 2E7A  A9 00            LDA     #0              ;INIT MULTIPLE OF 125 (MLT125) TO ZERO
3042 2E7C  8D 06 1F         STA     MLT125
3043 2E7F  8D 07 1F         STA     MLT125+1
3044                        ;
3045                        ;       DO UNTIL (MLT125 > BUFLEN)
3046                        ;
3047 2E82  A9 7D    FINDGM  LDA     #125            ;INC THE MULTIPLE OF 125 BY 125
3048 2E84  18               CLC                     ;TO GET THE NEXT HIGHER MULTIPLE
3049
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    3050  2E85  6D 06 1F          ADC     MLT125
    3051  2E88  8D 06 1F          STA     MLT125
    3052  2E8B  A9 00             LDA     #0
    3053  2E8D  6D 07 1F          ADC     MLT125+1        ;MLT125 IS IN LSB,MSB ORDER
    3054  2E90  8D 07 1F          STA     MLT125+1
    3055                        ;
    3056                        ;       TEST FOR MLT125 > BUFLEN - LOOP TEST
    3057                        ;
    3058  2E93  AD 05 1F          LDA     BUFLEN+1        ;IS MSB OF MLT125 > MSB OF BUFLEN?
    3059  2E96  CD 07 1F          CMP     MLT125+1
    3060  2E99  90 0A             BCC     GETMLT          ;YES, THEN END LOOP
    3061  2E9B  D0 E5             BNE     FINDGM          ;IF MLT<BUFLEN, REPEAT LOOP
    3062  2E9D  AD 04 1F          LDA     BUFLEN          ;ELSE MSB'S ARE =. CHECK THE LSB'S.
    3063  2EA0  CD 06 1F          CMP     MLT125          ;IS LSB MLT125 > LSB BUFLEN?
    3064  2EA3  B0 DD             BCS     FINDGM          ;NO, REPEAT LOOP
    3065                        ;                       ;ELSE END LOOP.
    3066                        ;********************** END OF LOOP************************
    3067                        ;
    3068                        ;       CHECK IF MULTIPLE = TO 125.  IF IS, THEN LEAVE BUFLEN AS IS.  IF
    3069                        ;       ISN'T THEN SET BUFLEN TO THAT NULTIPLE OF 125 MINUS 125.
    3070                        ;
    3071  2EA5  AD 07 1F        GETMLT  LDA     MLT125+1        ;IS MSB NOT = ZERO?
    3072  2EA8  D0 08             BNE     REPLAC          ;YES, VALUE IS > 125
    3073  2EAA  A9 7D             LDA     #125            ;IS LSB > 125?
    3074  2EAC  CD 06 1F          CMP     MLT125
    3075  2EAF  90 01             BCC     REPLAC          ;YES, REPLACE BUFLEN WITH MLT125
    3076  2EB1  60                RTS                     ;ELSE LEAVE BUFLEN AS IS
    3077                        ;
    3078  2EB2  AD 06 1F        REPLAC  LDA     MLT125          ;SUBTRACT 125 FROM MLT125 TO GET
    3079  2EB5  38                SEC                     ;GREATEST MULTIPLE LESS THAN OR EQUAL
    3080  2EB6  E9 7D             SBC     #125            ;TO THE PROGRAM AREA.
    3081  2EB8  8D 04 1F          STA     BUFLEN          ;USE IT AS THE BUFFER LENGTH.
    3082  2EBB  AD 07 1F          LDA     MLT125+1
    3083  2EBE  E9 00             SBC     #0
    3084  2EC0  8D 05 1F          STA     BUFLEN+1
    3085  2EC3  60                RTS                     ;RETURN
    3086                        ;
    3087                        ;USE BUFFER DBUF (250 BYTES) INSTEAD OF PROGRAM AREA
    3088                        ;
    3089  2EC4  A9 F4           USEBUF  LDA     #.LOW.DBUFL     ;USE DBUF AS
    3090  2EC6  85 1A             STA     BUFADR          ;BUFFER ADDRESS
    3091  2EC8  A9 1D             LDA     #.LOW.DBUFH     ;IN LSB,MSB ORDER
    3092  2ECA  85 1B             STA     BUFADR+1
    3093  2ECC  A9 FA             LDA     #EDBLL          ;STORE DATA
    3094  2ECE  8D 04 1F          STA     BUFLEN          ;BUFFER LENGTH
    3095  2ED1  A9 00             LDA     #EDBLH          ;=TO 256(100HEX)
    3096  2ED3  8D 05 1F          STA     BUFLEN+1        ;IN LSB,MSB ORDER
    3097  2ED6  60                RTS                     ;RETURN
```

```
3098                                    .PAGE
3099                            ;  **** CHECK FILENAME FOR WILDCARD CHARACTERS ****
3100                            ;
3101                            ;
3102                            ;       CHECKS THE STRING AT PAR,X FOR WLDCARD CHARACTERS (* OR ?). IF
3103                            ;       THEY ARE FOUND THE ROUTINE SETS THE = FLAG. IF A <CR> IS FOUND
3104                            ;       RETURNS TO THE CALLING ROUTINE WITH THE EQUAL FLAG RESET.
3105                            ;
3106  2ED7  BD 7C 1D            LOOKWC  LDA     PAR,X
3107  2EDA  E8                          INX
3108  2EDB  C9 2A                       CMP     #'*
3109  2EDD  F0 0D                       BEQ     LOOKW2
3110  2EDF  C9 3F                       CMP     #'?
3111  2EE1  F0 09                       BEQ     LOOKW2
3112  2EE3  C9 9B                       CMP     #CR
3113  2EE5  F0 04                       BEQ     LOOKW1
3114  2EE7  C9 2C                       CMP     #',             ;TERMINATE WITH CR OR COMMA
3115  2EE9  D0 EC                       BNE     LOOKWC
3116                            ;
3117  2EEB  E8                  LOOKW1  INX
3118  2EEC  60                  LOOKW2  RTS
```

```
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135  2EED  E8
3136  2EEE  BD 7C 1D
3137  2EF1  C9 3A
3138  2EF3  F0 01
3139  2EF5  E8
3140  2EF6  E8
3141
3142
3143
3144  2EF7  A0 00
3145
3146  2EF9  B9 CD 28
3147  2EFC  DD 7C 1D
3148  2EFF  D0 10
3149  2F01  C8
3150  2F02  E8
3151  2F03  C0 07
3152  2F05  D0 F2
3153
3154
3155
3156  2F07  A9 12
3157  2F09  A2 2F
3158  2F0B  20 B5 31
3159  2F0E  4C B6 20
3160
3161
3162
3163  2F11  60
3164
3165  2F12  44 45 53 54
3166  2F16  49 4E 41 54
3167  2F1A  49 4F 4E 20
3168  2F1E  43 41 4E 54
3169  2F22  20 42 45 20
3170  2F26  44 4F 53 2E
3171  2F2A  53 59 53 9B
3172  2F2E
```

```
        .PAGE
;  **** TEST FILE SPEC FOR DOS.SYS ****
;
;
;       SUBROUTINE - TSTDOS
;
;       CHECKS A FILE SPEC IN THE STORAGE LOC FOR DOS.SYS.  USED TO
;       PREVENT COPYING TO A FILE NAMED DOS.SYS.  IF DOS SYS IS OPENED
;       OUTPUT FMS WILL WRITE A COPY OF DOS OUT TO THE FILE
;
;       ENTRY - REG X HAS INDEX INTO PAR TO FIRST CHAR OF FILE SPEC
;               ASSUMES COMPLETE FILE SPEC.
;       EXIT  - WILL NOT RETURN IF FILE NAME = DOS.SYS, BUT GOES TO MENU
;
;       FIND END OF DEVICE ID - COLON
;
TSTDOS  INX                     ;NEVER IS FIRST CHAR
        LDA     PAR,X           ;GET 2ND CHAR
        CMP     #':             ;IS IT A COLON?
        BEQ     GOTCOL          ;YES, THEN NAME STARTS AT CHAR 3
        INX                     ;ELSE NAME STARTS AT CHAR 4
GOTCOL  INX                     ;POINT AT FIRST CHAR OF NAME
;
;       COMPARE FILE NAME IN PAR WITH DOS.SYS
;
        LDY     #0              ;INDEX INTO DOS.SYS FILE SPEC
;
NXTCHAR LDA     DS+3,Y          ;GET NEXT DOS.SYS CHAR
        CMP     PAR,X           ;TEST IF FILE NAME IS SAME
        BNE     NOTSAM          ;NO, THEN RETURN
        INY
        INX                     ;ELSE TRY NEXT CHAR
        CPY     #7              ;ARE THERE MORE CHARS TO TRY?
        BNE     NXTCHAR         ;YES, DO AGAIN
;
;       FILE NAME EQUALS DOS.SYS - ERROR EXIT
;
        LDA     #.LOW.DCDSL     ;PRINT MSG - DEST CAN'T BE DOS.SYS
        LDX     #.LOW.DCDSH
        JSR     DSPLIN
        JMP     MENUSL          ;GOTO MENU
;
;       NOT EQUAL TO DOS.SYS - RETURN TO CALLER
;
NOTSAM  RTS
;
DCDS    .BYTE   'DESTINATION CANT BE DOS.SYS',CR



        HILO    DCDS
```

```
3173  002F                    +DCDSH      =       DCDS/256
3174  0012                    +DCDSL      =       (-256)*DCDSH+DCDS
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    3175
    3176
    3177
    3178
    3179  2F2E  18 30
    3180  2F30  A9 00
    3181  2F32  8D A0 18
    3182  2F35  8D BE 18
    3183  2F38  20 CF 30
    3184  2F3B  AD 9E 15
    3185  2F3E  48
    3186  2F3F  AE 01 1F
    3187  2F42  A9 9B
    3188  2F44  9D 7B 1D
    3189  2F47  20 24 32
    3190  2F4A  8D E0 19
    3191  2F4D  8E E1 19
    3192  2F50  E0 32
    3193  2F52  B0 03
    3194  2F54  CE 94 18
    3195  2F57  20 24 32
    3196  2F5A  8D E2 19
    3197  2F5D  8E E3 19
    3198  2F60  38
    3199  2F61  ED E0 19
    3200  2F64  8D F8 2F
    3201  2F67  8A
    3202  2F68  ED E1 19
    3203  2F6B  10 03
    3204  2F6D  4C B6 20
    3205  2F70  8D FD 2F
    3206  2F73  C0 9B
    3207  2F75  F0 29
    3208  2F77  20 24 32
    3209  2F7A  8D E2 02
    3210  2F7D  8E E3 02
    3211  2F80  0D E3 02
    3212  2F83  F0 03
    3213  2F85  CE A0 18
    3214  2F88  C0 9B
    3215  2F8A  F0 14
    3216  2F8C  20 24 32
    3217  2F8F  20 C4 30
    3218  2F92  8D E0 02
    3219  2F95  8E E1 02
    3220  2F98  0D E1 02
    3221  2F9B  F0 03
    3222  2F9D  CE BE 18
    3223  2FA0  A9 00
    3224  2FA2  8D 9E 15
    3225  2FA5  68
    3226  2FA6  C9 41
    3227  2FA8  D0 03
    3228  2FAA  CE 9E 15
```

```
          .PAGE
;   ****  SAVE  FILE  ROUTINE  ****
;
;
SAVFIL    .WORD   SFMG
          LDA     #0
          STA     INITQ+1
          STA     RUNQ+1
          JSR     GETIC1
          LDA     OPT
          PHA
          LDX     PTR             ;PUT EOL ON FILENAME
          LDA     #CR
          STA     PAR-1,X
          JSR     GETNO           ;GET HEX PARAMETER
          STA     LDST
          STX     LDST+1
          CPX     #.LOW.NDSH
          BCS     DSLMFG          ;BRANCH IF NOT SAVING DUP AREA
          DEC     WDR1+1
DSLMFG    JSR     GETNO           ;END ADDRESS
          STA     LDND
          STX     LDND+1
          SEC
          SBC     LDST
          STA     WDRL+1
          TXA
          SBC     LDST+1
          BPL     ADDOK           ;BR IF ENDING ADDR > THAN STARTING
          JMP     MENUSL          ;ELSE BACK TO MENU
ADDOK     STA     WDRH+1
          CPY     #CR
          BEQ     NRUNAD          ;BRANCH IF NO MORE PARAMS
          JSR     GETNO           ;GET A RUN ADDRESS IF ANY
          STA     INITAD
          STX     INITAD+1
          ORA     INITAD+1
          BEQ     NINTAD          ;BRANCH IF NO INIT ADDRESS GIVEN
          DEC     INITQ+1         ;SET FLAG
NINTAD    CPY     #CR
          BEQ     NRUNAD          ;BRANCH IF NO RUN ADDRESS GIVEN
          JSR     GETNO           ;GET RUN ADDRESS
          JSR     PERX            ;CHECK FOR ERRORS
          STA     RUNAD
          STX     RUNAD+1
          ORA     RUNAD+1
          BEQ     NRUNAD          ;BRANCH IF NO RUN ADDRESS
          DEC     RUNQ+1          ;SET FLAG
NRUNAD    LDA     #0
          STA     OPT             ;OPTION CHAR FROM FILENAME
          PLA
          CMP     #'A             ;IF APPEND
          BNE     *+5
          DEC     OPT             ;SET OT=$FF
```

```
3229                            ;
3230                            ; OPEN THE FILE
3231                            ;
3232  2FAD  A2 10               LDX     #$10
3233  2FAF  A9 03               LDA     #OPEN
3234  2FB1  9D 42 03            STA     ICCOM,X
3235  2FB4  2C 9E 15            BIT     OPT             ;IF APPEND
3236  2FB7  30 04               BMI     *+6
3237  2FB9  A9 08               LDA     #8
3238  2FBB  D0 02               BNE     *+4
3239  2FBD  A9 09               LDA     #9
3240  2FBF  9D 4A 03            STA     ICAX1,X
3241  2FC2  20 EE 31            JSR     CIOCL
3242                            ;
3243                            ; WRITE SAVE FILE HEADER
3244                            ;
3245  2FC5  A9 0B               LDA     #PUTCHR
3246  2FC7  9D 42 03            STA     ICCOM,X
3247  2FCA  A9 DE               LDA     #.LOW.SAVHL
3248  2FCC  9D 44 03            STA     ICBAL,X
3249  2FCF  A9 19               LDA     #.LOW.SAVHH
3250  2FD1  9D 45 03            STA     ICBAH,X
3251  2FD4  A9 06               LDA     #6
3252  2FD6  9D 48 03            STA     ICBLL,X
3253  2FD9  A9 00               LDA     #0
3254  2FDB  9D 49 03            STA     ICBLH,X
3255  2FDE  2C 9E 15            BIT     OPT
3256  2FE1  10 0F               BPL     WHEAD           ;BRANCH IF NOT APPEND
3257  2FE3  A9 04               LDA     #4
3258  2FE5  9D 48 03            STA     ICBLL,X
3259  2FE8  A9 E0               LDA     #.LOW.LDSTL
3260  2FEA  9D 44 03            STA     ICBAL,X
3261  2FED  A9 19               LDA     #.LOW.LDSTH
3262  2FEF  9D 45 03            STA     ICBAH,X
3263  2FF2  20 EE 31    WHEAD   JSR     CIOCL
3264                            ;
3265                            ; WRITE DATA RECORD
3266                            ;
3267  2FF5  A2 10       WDR     LDX     #$10
3268  2FF7  A9 00       WDRL    LDA     #0              ;THIS IMMEDIATE VALUE MODIFIED
3269  2FF9  9D 48 03            STA     ICBLL,X
3270  2FFC  A9 00       WDRH    LDA     #0              ;THIS IMMEDIATE VALUE MODIFIED
3271  2FFE  9D 49 03            STA     ICBLH,X
3272  3001  FE 48 03            INC     ICBLL,X
3273  3004  D0 03               BNE     *+5
3274  3006  FE 49 03            INC     ICBLH,X
3275  3009  AD E0 19            LDA     LDST
3276  300C  9D 44 03            STA     ICBAL,X
3277  300F  AD E1 19            LDA     LDST+1
3278  3012  9D 45 03            STA     ICBAH,X
3279  3015  4C 93 18    WEX     JMP     WDR1
3280  3018  53 41 56 45 SFMG    .BYTE   'SAVE-GIVE FILE,START,END(,INIT,RUN)',CR
3281  301C  2D 47 49 56
3282  3020  45 20 46 49
```

```
    3283  3024  4C 45 2C 53
    3284  3028  54 41 52 54
    3285  302C  2C 45 4E 44
    3286  3030  28 2C 49 4E
    3287  3034  49 54 2C 52
    3288  3038  55 4E 29 9B
```

```
3289
3290
3291
3292
3293  303C  A9 9B
3294  303E  A2 4F
3295  3040  9D A4 1D
3296  3043  CA
3297  3044  10 FA
3298  3046  A9 00
3299  3048  8D 01 1F
3300  304B  8D 02 1F
3301  304E  8D F5 1E
3302  3051  20 58 30
3303  3054  20 BB 31
3304  3057  60
3305
3306
3307
3308
3309
3310  3058  A9 05
3311  305A  8D 42 03
3312  305D  A9 A4
3313  305F  8D 44 03
3314  3062  A9 1D
3315  3064  8D 45 03
3316  3067  A9 50
3317  3069  8D 48 03
3318  306C  A9 00
3319  306E  8D 49 03
3320  3071  A2 00
3321  3073  20 56 E4
3322  3076  C0 80
3323  3078  D0 03
3324  307A  CE F5 1E
3325  307D  60
3326
3327
3328
3329
3330  307E  A9 00
3331  3080  8D F5 1E
3332  3083  20 58 30
3333  3086  AD 48 03
3334  3089  8D F7 1E
3335  308C  20 BB 31
3336  308F  AD F5 1E
3337  3092  10 06
3338  3094  20 AA 19
3339  3097  4C B6 20
3340  309A  AD F7 1E
3341  309D  C9 03
3342  309F  30 0A

;  **** MISC. SUBROUTINES ****
;
;
GETLIN  LDA     #CR
        LDX     #79
        STA     LINE,X
        DEX
        BPL     *-4
        LDA     #0
        STA     PTR
        STA     IPTR
        STA     PER
        JSR     CIOGET
        JSR     SCROL
        RTS
;
;
;
; CIOGET - GET LINE OF INPUT FROM SCREEN EDITOR
;
CIOGET  LDA     #GETREC
        STA     ICCOM           ;SCREEN EDIT IOCB
        LDA     #LBUFL
        STA     ICBAL
        LDA     #LBUFH
        STA     ICBAH
        LDA     #80
        STA     ICBLL
        LDA     #0
        STA     ICBLH
        LDX     #0
        JSR     CIO             ;READ RECORD FROM SCREEN EDITOR
        CPY     #$80            ;CHECK FOR BREAK ABORT STATUS
        BNE     *+5
        DEC     PER             ;PARAM ERROR FLAG IS SET IF SO
        RTS
;
;
; CHRGET - GET 1 CHAR FROM EDITOR IN A.
;
CHRGET  LDA     #0
        STA     PER
CHRG1   JSR     CIOGET          ;GET A LINE FROM E:
        LDA     ICBLL           ;SAVE CHAR COUNT
        STA     RCNT
        JSR     SCROL
        LDA     PER
        BPL     CHRG2           ;IF BREAK, CLOSE AND EXIT
        JSR     CLOSX
        JMP     MENUSL
CHRG2   LDA     RCNT            ;EXPECT 1 OR 2 CHARACTERS
        CMP     #3
        BMI     CHRG3           ;IF OK
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    3343  30A1  A9 AF
    3344  30A3  A2 30
    3345  30A5  20 B5 31
    3346  30A8  4C 83 30
    3347  30AB  AD A4 1D
    3348  30AE  60
```

```
          LDA     #.LOW.OLL
          LDX     #.LOW.OLH
          JSR     DSPLIN
          JMP     CHRG1           ;TRY AGAIN
CHRG3     LDA     LINE            ;GET 1ST CHAR
          RTS
```

```
3349                                        .PAGE
3350  30AF  50 4C 45 41                OL    .BYTE   'PLEASE TYPE 1 LETTER',CR
3351  30B3  53 45 20 54
3352  30B7  59 50 45 20
3353  30BB  31 20 4C 45
3354  30BF  54 54 45 52
3355  30C3  9B
3356  30C4                              HILO    OL
3357  0030                        +OLH  =       OL/256
3358  00AF                        +OLL  =       (-256)*OLH+OL
3359                                    ;
3360                                    ; PERX - EXIT IF PARAMETER ERRORS
3361                                    ;
3362  30C4  2C F5 1E                    PERX    BIT     PER
3363  30C7  30 01                       BMI     PERX1
3364  30C9  60                          RTS
3365  30CA  68                          PLA
3366  30CB  68                          PLA
3367  30CC  4C B6 20                    JMP     MENUSL
3368                                    ;
3369                                    ; GETIC1 - READ LINE, GET FILENAME, POINT TO IT IN IOCB1
3370                                    ;
3371  30CF  20 3C 30                    GETIC1  JSR     GETLINE
3372  30D2  A2 10                       GETIC2  LDX     #$10
3373  30D4  20 DD 31                    JSR     PIOCB
3374  30D7  4C E8 30                    JMP     GETFIL
3375                                    ;
3376                                    ;
3377  30DA  A9 08                       GETNAME LDA     #8              ;ENTRY TO GETFIL USED BY RENAME
3378  30DC  8D 03 1F                    STA     CTR             ;WHICH DOES NOT HAVE A DEVICE ID
3379  30DF  AC 01 1F                    LDY     PTR             ;FOR THE SECOND FILE SPEC
3380  30E2  AE 02 1F                    LDX     IPTR
3381  30E5  4C 41 31                    JMP     CFTE
3382                                    ;
3383                                    ;       SUBROUTINE - GETFIL
3384                                    ;       REMOVES ONE FILE SPECIFICATION FROM THE INPUT LINE.  WILL SET UP
3385                                    ;       THE SPEC FOR DEFAULTS FOR INCOMPLETE DRIVE ID.  DEFAULT DRIVE #
3386                                    ;       IS 1.
3387                                    ;
3388                                    ;
3389                                    ;GET FILESPEC FROM INPUT LINE
3390  30E8  AC 01 1F                    GETFIL  LDY     PTR
3391  30EB  AE 02 1F                    LDX     IPTR
3392  30EE  A9 0B                       LDA     #11
3393  30F0  8D 03 1F                    STA     CTR
3394                                    ;
3395                                    ;AVOID GETTING JUNK ON VERY SHORT PARAMS
3396                                    ;
3397  30F3  BD A4 1D                    LDA     LINE,X
3398  30F6  C9 2C                       CMP     #',
3399  30F8  F0 3B                       BEQ     ADDC
3400  30FA  C9 9B                       CMP     #CR
3401  30FC  F0 37                       BEQ     ADDC
3402  30FE  BD A5 1D                    LDA     LINE+1,X
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    3403  3101  C9 2C
    3404  3103  F0 22
    3405  3105  C9 9B
    3406  3107  F0 1E
    3407  3109  A9 3A
    3408  310B  DD A6 1D
    3409  310E  F0 31
    3410  3110  DD A5 1D
    3411  3113  D0 12
    3412  3115  CE 03 1F
    3413  3118  BD A4 1D
    3414  311B  C9 41
    3415  311D  10 22
    3416
    3417
    3418
    3419  311F  A9 44
    3420  3121  99 7C 1D
    3421  3124  C8
    3422  3125  10 1A
    3423  3127  CE 03 1F
    3424  312A  CE 03 1F
    3425  312D  DD A4 1D
    3426  3130  F0 ED
    3427  3132  CE 03 1F
    3428  3135  A9 44
    3429  3137  99 7C 1D
    3430  313A  C8
    3431  313B  A9 3A
    3432  313D  99 7C 1D
    3433  3140  C8
    3434  3141  A9 00
    3435  3143  8D 9E 15
    3436  3146  BD A4 1D
    3437  3149  99 7C 1D
    3438  314C  E8
    3439  314D  C8
    3440  314E  C9 9B
    3441  3150  F0 2C
    3442  3152  C9 2C
    3443  3154  F0 28
    3444  3156  C9 2F
    3445  3158  F0 2B
    3446  315A  C9 2E
    3447  315C  D0 05
    3448  315E  A9 04
    3449  3160  8D 03 1F
    3450  3163  CE 03 1F
    3451  3166  10 DE
    3452
    3453
    3454
    3455  3168  A9 95
    3456  316A  A2 31
```

```
        CMP     #',
        BEQ     GT1
        CMP     #CR
        BEQ     GT1
        LDA     #':             ;LOOK FOR : IN FILESPEC
        CMP     LINE+2,X        ;SEE IF HAVE COMPLETE FILESPEC ALREADY
        BEQ     CFTE
        CMP     LINE+1,X
        BNE     GT1
        DEC     CTR
        LDA     LINE,X
        CMP     #'A
        BPL     CFTE            ;HAVE X FILE, COMPLETE FILESPEC
;
; IF FALLS THRU, IS UNIT:FILE, ADD D
;
GT2     LDA     #'D
        STA     PAR,Y
        INY
        BPL     CFTE
GT1     DEC     CTR
        DEC     CTR
        CMP     LINE,X          ;AN UNLIKELY CASE (:FILE)
        BEQ     GT2             ;TREAT FILE AS D:FILE
        DEC     CTR
ADDC    LDA     #'D
        STA     PAR,Y
        INY
        LDA     #':
        STA     PAR,Y
        INY
CFTE    LDA     #0
        STA     OPT
CFTE1   LDA     LINE,X
        STA     PAR,Y
        INX
        INY
        CMP     #CR             ;LOOK FOR TERMINATOR
        BEQ     EOC
        CMP     #',
        BEQ     EOC
        CMP     #'/
        BEQ     POPT
        CMP     #'.             ;LOOK FOR START OF .EXT
        BNE     CFTE2
        LDA     #4              ;FOUND, 4 MORE CHARS MAX
        STA     CTR
CFTE2   DEC     CTR
        BPL     CFTE1
;
;GETS HERE IF TOO MANY CHARS IN FILENAME
;
        LDA     #.LOW.NTLL
        LDX     #.LOW.NTLH
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    3457  316C  20 B5 31
    3458  316F  CE F5 1E
    3459  3172  BD A4 1D
    3460  3175  E8
    3461  3176  C9 2C
    3462  3178  F0 04
    3463  317A  C9 9B
    3464  317C  D0 F4
    3465  317E  8E 02 1F
    3466  3181  8C 01 1F
    3467  3184  60
    3468  3185  BD A4 1D
    3469  3188  8D 9E 15
    3470  318B  E8
    3471  318C  BD A4 1D
    3472  318F  99 7B 1D
    3473  3192  E8
    3474  3193  10 E9
    3475  3195  4E 41 4D 45
    3476  3199  20 54 4F 4F
    3477  319D  20 4C 4F 4E
    3478  31A1  47 9B
    3479  31A3
    3480  0031
    3481  0095
    3482
    3483
    3484
    3485
    3486  31A3  A9 0B
    3487  31A5  8D 42 03
    3488  31A8  A2 00
    3489
    3490  31AA  20 56 E4
    3491  31AD  C0 80
    3492  31AF  D0 03
    3493  31B1  4C B6 20
    3494  31B4  60
    3495
    3496
    3497
    3498  31B5  20 BE 19
    3499  31B8  4C BB 31
    3500
    3501
    3502
    3503
    3504  31BB  A9 00
    3505  31BD  AA
    3506  31BE  9D 49 03
    3507  31C1  A9 0A
    3508  31C3  9D 48 03
    3509  31C6  A9 31
    3510  31C8  9D 45 03
```

```
          JSR     DSPLIN          ;NAME TOO LONG
          DEC     PER             ;SET PARAMETER ERROR FLAG
STE       LDA     LINE,X          ;SKIP TO END
          INX
          CMP     #',
          BEQ     EOC
          CMP     #CR
          BNE     STE
EOC       STX     IPTR
          STY     PTR
          RTS
POPT      LDA     LINE,X
          STA     OPT
          INX
          LDA     LINE,X
          STA     PAR-1,Y         ;CHANGE STORED TERMINATOR TO , OR CR I H
          INX
          BPL     EOC
NTL       .BYTE   'NAME TOO LONG',CR



          HILO    NTL
+NTLH      =      NTL/256
+NTLL      =      (-256)*NTLH+NTL
;
; DSPMSG - DISPLAY N BYTES
; BUFFER POINTER AND LENGTH ARE ALREADY IN IOCB0
;
DSPMSG    LDA     #PUTCHR
          STA     ICCOM
          LDX     #0
;
CIO1      JSR     CIO             ;CALL CIO AND GO TO MENUSL
          CPY     #$80            ;IF BREAK KEY ABORT
          BNE     *+5
          JMP     MENUSL
          RTS
;
; DSPLIN - DISPLAY ONE LINE OF TEXT
;        A=LO, X=HI ADDRESS
DSPLIN    JSR     PRNTMSG         ;USE RESIDENT DUP SUBROUTINE
          JMP     SCROL           ;SCROLL SCREEN BELOW MENU & RETURN
;
;
; SCROL - DO SCROLLING OF AREA BELOW MENU
;
SCROL     LDA     #0
          TAX
          STA     ICBLH,X
          LDA     #10
          STA     ICBLL,X
          LDA     #.LOW.ZAPH
          STA     ICBAH,X
```

```
    3511  31CB  A9 D3          LDA     #.LOW. ZAPL
    3512  31CD  9D 44 03       STA     ICBAL, X
    3513  31D0  4C A3 31       JMP     DSPMSG
```

```
3514                                    .PAGE
3515  31D3  1C 1C 1C 1C        ZAP     .BYTE   CUP,CUP,CUP,CUP,CUP
3516  31D7  1C
3517  31D8  9C 1D 1D 1D                 .BYTE   DLL,CDN,CDN,CDN,CDN
3518  31DC  1D
3519  31DD                              HILO    ZAP
3520  0031                    +ZAPH      =      ZAP/256
3521  00D3                    +ZAPL      =      (-256)*ZAPH+ZAP
3522                                    ;
3523                                    ; PIOCB - POINT IOCB AT PAR(PTR)
3524                                    ;
3525  31DD  A9 7C               PIOCB   LDA     #PARL
3526  31DF  18                          CLC
3527  31E0  6D 01 1F                    ADC     PTR
3528  31E3  9D 44 03                    STA     ICBAL,X
3529  31E6  A9 1D                       LDA     #PARH
3530  31E8  69 00                       ADC     #0
3531  31EA  9D 45 03                    STA     ICBAH,X
3532  31ED  60                          RTS
3533                                    ;
3534                                    ; CIOCL - CALL CIO AND PROCESS ANY ERRORS
3535                                    ;
3536  31EE  20 56 E4            CIOCL   JSR     CIO             ;CALL CIO
3537  31F1  98                          TYA
3538  31F2  30 01                       BMI     *+3
3539  31F4  60                          RTS                     ;OK, RETURN
3540  31F5  98                  CIOER1  TYA                     ;ERROR STATUS
3541  31F6  38                  CIOER   SEC
3542  31F7  E9 64                       SBC     #100            ;ERROR NUMS ALWAYS ARE 1XX DEC
3543  31F9  A2 2F                       LDX     #'0-1           ;CONVERT TENS
3544  31FB  E8                  CTNS    INX
3545  31FC  38                          SEC
3546  31FD  E9 0A                       SBC     #10
3547  31FF  10 FA                       BPL     CTNS            ;THE EASY (SLOW) WAY
3548  3201  18                          CLC
3549  3202  69 3A                       ADC     #10+'0          ;CONVERT
3550  3204  8D 22 32                    STA     EUN
3551  3207  8E 21 32                    STX     ETN
3552  320A  A2 32                       LDX     #.LOW.CIEH
3553  320C  A9 17                       LDA     #.LOW.CIEL
3554  320E  20 B5 31            CIEX    JSR     DSPLIN
3555  3211  20 AA 19                    JSR     CLOSX           ;CLOSE IOCBS 10,20
3556  3214  4C B6 20                    JMP     MENUSL
3557  3217  45 52 52 4F        CIE     .BYTE   'ERROR-   1'
3558  321B  52 2D 20 20
3559  321F  20 31
3560  3221  00                  ETN     .BYTE   0
3561  3222  00                  EUN     .BYTE   0
3562  3223  9B                          .BYTE   CR
3563  3224                              HILO    CIE
3564  0032                    +CIEH      =      CIE/256
3565  0017                    +CIEL      =      (-256)*CIEH+CIE
3566                                    ;
3567                                    ;
```

```
3568                    ;
3569                    ; GETNO - GET HEX NUMERIC PARAMETER FROM LINE(IPTR).
3570                    ;         RETURN A=LO, X=HI.  PER SET MINUS IF ERROR.
3571                    ;         INC IPTR PAST PARAM.
3572  3224  A9 04       ;
3573  3226  8D 03 1F    GETNO   LDA     #4              ;MAX NO DIGITS
3574  3229  A9 00               STA     CTR
3575  322B  8D 04 1F           LDA     #0
3576  322E  8D 05 1F           STA     T1
3577  3231  AE 02 1F           STA     T1+1            ;INIT TEMP TO BUILD NUMBER IN
3578  3234  BD A4 1D    GHB     LDX     IPTR
3579  3237  EE 02 1F           LDA     LINE,X          ;GET CHAR
3580  323A  C9 9B               INC     IPTR
3581  323C  F0 2B               CMP     #CR             ;SEE IF TERMINATOR
3582  323E  C9 2C               BEQ     GND
3583  3240  F0 27               CMP     #','
3584  3242  20 A5 32            BEQ     GND
3585  3245  30 2A               JSR     HEXCON          ;CONVERT ASCII TO NIBBLE
3586  3247  A0 03               BMI     ERRX            ;IF ERROR
3587  3249  18                 LDY     #3              ;SHIFT T1,T1+1 BY 4
3588  324A  2E 05 1F    SHT1    CLC
3589  324D  2E 04 1F           ROL     T1+1
3590  3250  88                 ROL     T1
3591  3251  10 F6               DEY
3592  3253  0D 05 1F           BPL     SHT1
3593  3256  8D 05 1F           ORA     T1+1            ;OR IN NEW NIBBLE
3594  3259  CE 03 1F           STA     T1+1
3595  325C  10 D3               DEC     CTR             ;COUNT DIGIT
3596  325E  A9 77               BPL     GHB             ;LOOP UNLESS TOO MANY DIGITS
3597  3260  A2 32               LDA     #.LOW.TMDL
3598  3262  20 B5 31            LDX     #.LOW.TMDH
3599  3265  CE F5 1E    ERRX1   JSR     DSPLIN
3600  3268  60                 DEC     PER
3601  3269  A8                 RTS
3602  326A  AD 05 1F    GND     TAY
3603  326D  AE 04 1F           LDA     T1+1
3604  3270  60                 LDX     T1
3605  3271  A9 87               RTS
3606  3273  A2 32       ERRX    LDA     #.LOW.IHPL      ;INVALID HEX PARAM
3607  3275  D0 EB               LDX     #.LOW.IHPH
3608  3277  54 4F 4F 20        BNE     ERRX1
3609  327B  4D 41 4E 59  TMD     .BYTE   'TOO MANY DIGITS',CR
3610  327F  20 44 49 47
3611  3283  49 54 53 9B
3612  3287                    HILO    TMD
3613  0032              +TMDH   =       TMD/256
3614  0077              +TMDL   =       (-256)*TMDH+TMD
3615  3287  49 4E 56 41  IHP     .BYTE   'INVALID HEXADECIMAL PARAMETER',CR
3616  328B  4C 49 44 20
3617  328F  48 45 58 41
3618  3293  44 45 43 49
3619  3297  4D 41 4C 20
3620  329B  50 41 52 41
3621  329F  4D 45 54 45
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    3622  32A3  52 9B
    3623  32A5
    3624  0032
    3625  0087
    3626
    3627
    3628
    3629
    3630
    3631  32A5  38
    3632  32A6  E9 30
    3633  32A8  30 0F
    3634  32AA  C9 0A
    3635  32AC  30 0D
    3636  32AE  38
    3637  32AF  E9 07
    3638  32B1  C9 0A
    3639  32B3  30 04
    3640  32B5  C9 10
    3641  32B7  30 02
    3642  32B9  A9 FF
    3643  32BB  C9 00
    3644  32BD  60
    3645
    3646
    3647
    3648
    3649  32BE  2C F5 1E
    3650  32C1  30 27
    3651  32C3  AE 02 1F
    3652  32C6  BD A4 1D
    3653  32C9  E8
    3654  32CA  C9 44
    3655  32CC  F0 F8
    3656  32CE  38
    3657  32CF  E9 30
    3658  32D1  F0 18
    3659  32D3  30 16
    3660  32D5  C9 05
    3661  32D7  10 12
    3662  32D9  48
    3663  32DA  BD A4 1D
    3664  32DD  E8
    3665  32DE  C9 2C
    3666  32E0  F0 04
    3667  32E2  C9 9B
    3668  32E4  D0 F4
    3669  32E6  8E 02 1F
    3670  32E9  68
    3671  32EA  60
    3672  32EB  CE F5 1E
    3673  32EE  A9 F5
    3674  32F0  A2 32
    3675  32F2  4C B5 31

                HILO    IHP
+IHPH           =       IHP/256
+IHPL           =       (-256)*IHPH+IHP
                ;
                ;
                ; HEXCON - CONVERT ASCII CHAR IN A TO HEX NIBBLE IN A. RETURN
                ;          MINUS CONDITION, A=FF IF ERROR.
                ;
HEXCON  SEC
        SBC     #'0
        BMI     ERRX2           ;ASCII BELOW '0'
        CMP     #10
        BMI     OKX             ;0-9 CONVERTED SO EXIT
        SEC
        SBC     #'A-'0-10
        CMP     #10             ;CONVERTED VALUE MUST BE 10 OR MORE
        BMI     ERRX2           ;BETWEEN '9' AND 'A'
        CMP     #$10
        BMI     OKX             ;A-F CONVERTED
ERRX2   LDA     #$FF
OKX     CMP     #0              ;SET STATUS BY VALUE IN A
        RTS
                ;
                ;       GETDN - GET A DEVICE NUMBER FROM LINE(IPTR)
                ;       RETURN IT IN A
                ;
GETDN   BIT     PER             ;SEE IF PARAM ERROR ALREADY
        BMI     GDR             ;IF SO DON'T BOTHER
        LDX     IPTR
GETD    LDA     LINE,X
        INX
        CMP     #'D             ;IF DN
        BEQ     GETD            ;GO GET DIGIT
        SEC
        SBC     #'0             ;CONVERT DIGIT
        BEQ     BDS             ;CAN'T BE ZERO
        BMI     BDS             ;IF NOT DIGIT
        CMP     #5
        BPL     BDS             ;TOO LARGE
        PHA
GD1     LDA     LINE,X
        INX
        CMP     #',
        BEQ     GDX             ;IF TERMINATOR
        CMP     #CR
        BNE     GD1             ;KEEP LOOKING
GDX     STX     IPTR            ;ADVANCE POINTER
        PLA
GDR     RTS
BDS     DEC     PER
        LDA     #.LOW.NDSL      ;NEED DEVICE SPEC MSG
        LDX     #.LOW.NDSH
        JMP     DSPLIN
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    3676  32F5  4E 45 45 44
    3677  32F9  20 44 31 20
    3678  32FD  54 48 52 55
    3679  3301  20 44 34 9B
    3680  3305  00
    3681  13F9
    3682  3306
    3683  0013
    3684  00F9
    3685  1589
    3686  3306
    3687  0015
    3688  0089
    3689  3306
    3690  0032
    3691  00F5
    3692  3306
    3693  0033
    3694  0005
    3695  3306

ASSEMBLY ERRORS =     0
```

```
NDS       .BYTE   'NEED D1 THRU D4',CR


NMDUP     .BYTE   0
LEN       =       NMDUP-EDN
          HILO    LEN
+LENH      =      LEN/256
+LENL      =      (-256)*LENH+LEN
 MLEN     =       NMDUP-NDOS
          HILO    MLEN
+MLENH     =       MLEN/256
+MLENL     =       (-256)*MLENH+MLEN
          HILO    NDS
+NDSH      =      NDS/256
+NDSL      =      (-256)*NDSH+NDS
          HILO    NMDUP
+NMDUPH      =      NMDUP/256
+NMDUPL      =      (-256)*NMDUPH+NMDUP
          .END
```

CROSS REFERENCE

LABEL	VALUE	REFERENCE
AAM	2C59	2571 -2689
ADDC	3135	3399 3401 -3428
ADDOK	2F70	3203 -3205
ADOK	168B	295 298 -315
AF	170C	-383 388 389
AFH	0017	147 -388 389
AFL	00C	145 -389
ANWD	16AE	-331
ASP	2C76	2585 -2704
ASPT	2BA3	2572 -2585
ASPX	2CBC	2709 2711 -2713
AWD	16BB	324 327 332 -336
AWDQ	16FA	323 326 -369
AWDQR	1704	370 -375
BDS	32EB	3658 3659 3661 -3672
BFENHI	0035	-797 846 968
BFENLO	0034	-796 844 966
BLF	294D	-2309 2314 2315
BLFH	0029	2305 -2314 2315
BLFL	004D	2304 -2315
BRKKEY	0011	-27 1180
BRMG	2768	2053 -2066
BRUN	274C	1162 -2053
BSIOR	0772	-51 2737
BUFADR	001A	-119 1834 1837 2869 2871 2944 2946 3027 3029 3090 3092
BUFLEN	1F04	-1042 1841 1843 2873 2875 3033 3036 3058 3062 3081 3084 3094 3096
BUFRFL	003B	-802 927 982
BUFRHI	0033	-795 839 845 961 967
BUFRLO	0032	-794 837 843 877 952 959 965
CARTST	BFFA	-33 653
CBIT	2C72	2691 -2698
CDES	1EFF	-1036 1798 1821 1830 1847 1869 1874 2422 2445 2466 2725
CDN	001D	-57 1147 1147 1147 1147 1147 3517 3517 3517 3517
CDSK	26EB	1940 -1972
CDTMF3	022A	-53 723 725
CDTMV3	021C	-52 719 720
CFTE	3141	3381 3409 3415 3422 -3434
CFTE1	3146	-3436 3451
CFTE2	3163	3447 -3450
CHKDON	1A0E	857 -869 885
CHKERR	008F	-818 932
CHKSNT	003B	-799 849 856
CHKSUM	0031	-798 854 881 883 930 955 957
CHKTYP	2CFE	2780 -2786
CHKVER	266E	1361 1899 -1920

CHRG1	3083	-3332	3346						
CHRG2	309A	3337	-3340						
CHRG3	30AB	3342	-3347						
CHRGET	307E	1231	1477	1945	2084	2134	2471	2507	2563
		2756	3016	-3330					
CIE	3217	-3557	3564	3565					
CIEH	0032	3552	-3564	3565					
CIEL	0017	3553	-3565						
CIEX	320E	-3554							
CIO	E456	-20	202	218	234	356	439	452	506
		516	582	596	644	687	698	701	716
		735	742	771	1845	1868	1873	2879	2983
		3321	3490	3536					
CIO1	31AA	773	-3490						
CIOCL	31EE	1331	1385	1417	1434	1486	1494	1570	1583
		1593	1608	1686	1785	1822	1855	1906	1955
		2153	2158	2332	2346	2855	2929	2952	2958
		3241	3263	-3536					
CIOER	31F6	1860	2303	-3541					
CIOER1	31F5	2099	2742	2784	2887	-3540			
CIOGET	3058	3302	-3310	3332					
CIOINV	E46E	-25	705						
CKCART	271A	-2019	2024						
CKMDOS	157D	144	-154						
CKMEM	2CBE	2538	-2748						
CKRS	25FA	1854	-1856						
CLD1	2CA5	-2731	2741						
CLDKH	2CA0	2721	-2729						
CLDSET	1599	158	-169						
CLF	001E	-58							
CLFX	1646	205	248	-268	275	281			
CLMJMP	1912	-649	2038						
CLOC	2606	-1861							
CLOOP	25CB	-1840	1857						
CLOS1	22ED	1487	1489	-1491					
CLOS2	196E	683	-699						
CLOS20	19B4	573	584	-739					
CLOSE	000C	-66	514	699	732	740	1492	1591	1606
		1866	1871	2156	2956	2981			
CLOSX	19AA	152	241	273	278	420	441	455	621
		-732	1714	1803	2092	2307	3338	3555	
CLSCR	007D	-61	1060						
CLSDES	2E00	2940	-2956						
CLSSRC	2E1F	2899	2964	2973	-2980				
CMSI	2A02	-2378	2388	2389					
CMSIH	002A	-2388	2389	2754	3014				
CMSIL	0002	-2389	2753	3013					
COMPR1	2429	-1632	1640						
COMPR2	2434	1634	-1638						
COMPR3	2446	-1648	1655						
COMPR4	2451	1650	-1653						
COMPR5	2456	1644	-1657						
CPMG	231E	-1514	1541						

CPTR1	2CDE
CPYFIL	2378
CPYFL1	2394
CR	009B
CRT	001F
CSRC	1EFE
CSRCH	001E
CSRCL	00FE
CTNS	31FB
CTR	1F03
CUP	001C
DB1	1E74
DB1H	001E
DB1L	0074
DB3	1DF1
DB3H	001D
DB3L	00F1
DBLH	0001
DBLL	00C0
DBUF	1DF4
DBUFH	001D
DBUFHI	0305
DBUFL	00F4
DBUFLO	0304
DCB	0300
DCDS	2F12
DCDSH	002F
DCDSL	0012
DCOMND	0302
DDMG	29C2
DDSK	26E8
DDXT	2CDF
DELETE	0021
DELFIL	21C9
DELX	22E7
DEMG	230D
DF1	21F5
DINIT	16F7
DIRLST	2139
DKHND	E453

```
 2749 -2762
 1158 -1541
 1550 -1552
  -55   383   390   414   539   546   550   559
 1053  1061  1071  1071  1077  1086  1096  1105
 1114  1123  1133  1142  1233  1268  1281  1302
 1312  1344  1372  1472  1483  1495  1504  1514
 1518  1657  1745  1908  1957  1970  1973  2043
 2066  2076  2101  2193  2201  2211  2223  2229
 2309  2316  2334  2348  2358  2366  2378  2475
 2486  2654  2658  2666  2810  3112  3165  3187
 3206  3214  3280  3293  3350  3400  3405  3440
 3463  3475  3562  3580  3608  3615  3667  3676
  -59
-1035  1328  1685  1770  1782  1829  1840  1848
 1861  2405  2550  2645  2646  2689  2695  2698
-2645  2646  2650
-2646  2648
-3544  3547
-1040  2059  3378  3393  3412  3423  3424  3427
 3449  3450  3573  3594
  -56  3515  3515  3515  3515  3515
-1014  1020  1021
-1020  1021
-1021
-1015  1023  1024
-1023  1024  1397  1475
-1024  1395  1474
-1026
-1025
  221   223 -1013  1014  1015  1017  1018  1400
 1402  1407  1435  1447  1450  1459  1462  1463
 1470  1473  1484  2270  2404  2694
  209 -1017  1018  1420  2397  3091
  -88  2398  2582  2584  2590  2627
  207 -1018  1418  2399  3089
  -87  2400  2578  2581  2588  2625
  -83    84    85    86    87    88    89    90
-3165  3173  3174
 3157 -3173  3174
 3156 -3174
  -85  2776
-2358  2415
 1876  1877  1939 -1969
 2758 -2764
  -72   456  1383  1393
 1158 -1357
 1437 -1489
 1357 -1504
 1367 -1379
  360  -362
 1158 -1298
  -21  2779
```

DKWRT	2C98	2577	-2725					
DLL	009C	-60	3517					
DLM	16DB	340	342	-351				
DLM1	16EF	357	-359					
DLMG	21A7	1298	-1344					
DLSTO	2197	-1337	1341					
DLST1	219A	1336	-1338					
DMEND	2057	-1149	1150					
DMENU	1F0F	-1060	1150	1155	1156			
DMENUH	001F	-1155	1156	1210				
DMENUL	000F	-1156	1208					
DOCPY	25AB	1799	-1828					
DODEST	2DA4	2898	-2904					
DODKDP	2B05	2473	-2508					
DODUP	2D66	-2868	2974					
DORD	2B66	-2556	2596					
DOS	1540	-49	127	523	525			
DOSDRV	2875	2115	-2193					
DOSINI	000C	-29	518	520	524	526	667	669
DOSOS	2075	627	-1175	1177	1178	2186	2188	
DOSOSH	0020	-1177	1178					
DOSOSL	0075	-1178						
DOSTAT	2CF1	-2779	2783					
DOSVEC	000A	-28	131	133				
DOSWDP	2BD2	2567	-2610					
DOTSYS	2415	1613	-1619					
DOW	2B8D	2574	-2577					
DOWRIT	2DE2	2938	-2942					
DPFM	2D08	-2810	2817					
DRRDUP	18EC	624	-627					
DRTS	2CBC	2738	-2743					
DRUN	1621	243	-245					
DRUN1	1635	250	-258					
DRUN2	1644	259	-267					
DRV1	267A	1923	-1926					
DRVSTA	2CE4	2262	2436	2446	-2774			
DS	28CA	2124	2167	-2223	2227	2228	3146	
DSH	0028	2149	-2227	2228				
DSHI	030B	-90	2394	2409	2544	2636	2637	2707 2712
DSHIH	0003	-2636	2637	2650				
DSHIL	000B	-2637	2648					
DSKUTL	2092	-1195						
DSL	00CA	2147	-2228					
DSLMFG	2F57	3193	-3195					
DSLO	030A	-89	2396	2411	2542	2633	2634	2704 2710
DSLOH	0003	-2633	2634	2650				
DSLOL	000A	-2634	2648					
DSPLIN	31B5	1253	1265	1370	1389	1476	1663	1743 1802
		1944	2010	2091	2133	2142	2278	2306 2455
		2470	2506	2535	2562	2752	2755	2831 2895
		2970	2989	3012	3015	3158	3345	3457 -3498
		3554	3598	3675				
DSPMSG	31A3	1216	-3486	3513				

DSTATS	0303	-86						
DTH	1F0C	-1049	1051	1052				
DTHH	001F	-1051	1052	2175				
DTHL	000C	-1052	2173					
DU1	2092	-1196						
DU3	2620	1870	-1874					
DU4	2613	1862	-1869					
DU5	2634	1882	-1884					
DU6	262C	1875	-1879					
DUJPT	2057	-1158	1168	1169				
DUJPTH	0020	-1168	1169	1200				
DUJPTL	0057	-1169	1198					
DULEN	0148	-1150	1152	1153				
DULENH	0001	-1152	1153	1214				
DULENL	0048	-1153	1212					
DUNIT	0301	-84	2718	2726	2774			
DUNUM	000F	-1170	1196					
DUPDSK	2A58	1162	-2415					
DUPFEX	2E3E	2986	-2993					
DUPFIL	2D1E	1166	-2817					
DUPFLG	159D	154	166	-183	258	346	536	651	660
		768						
DUPSYS	182F	528	531	-546	2163			
DVSTAT	02EA	-80	2787					
DWQ	2209	1382	-1387					
EC	182C	-539	541	542				
ECH	0018	-541	542	712				
ECL	002C	-542	710					
EDBLH	0000	-1028	3095					
EDBLL	00FA	-1027	3093					
EDH	001F	-1054	1055					
EDL	000C	-1055						
EDN	1F0C	-1053	1054	1055	3681			
ENUF	2B3A	2532	-2538					
EOC	317E	3441	3443	3462	-3465	3474		
EOF	0088	-62	2885					
EOFFLG	1F0A	-1046	2861	2888	2963			
ERR	185B	490	491	-559				
ERRMES	183A	486	487	-550				
ERROR	17C2	482	-486					
ERRWR	178B	424	440	442	-454			
ERRX	3271	3585	-3605					
ERRX1	3262	-3598	3607					
ERRX2	32B9	3633	3639	-3642				
ERST	164F	219	237	-276	358			
ETN	3221	3551	-3560					
EUN	3222	3550	-3561					
FDP	26EA	-1971	1981	1982				
FDPH	0026	1951	-1981	1982				
FDPL	00EA	1948	-1982					
FINAL	17F7	478	484	-513				
FINDGM	2E82	-3048	3061	3064				
FMINIT	07E0	-48	142					

FMS	0700	-47 48 49
FMTDSK	2680	1162 -1934
FMX	26B6	1947 -1956
FORMAT	00FE	-73 1953
FRMERR	008C	-816 916
FTRF	1F0B	-1047 1048 2862 2906 2909
GC1	25AD	-1829
GD1	32DA	-3663 3668
GDR	32EA	3650 -3671
GDX	32E6	3666 -3669
GETCHR	0007	-68 216 688 1828 2877
GETD	32C6	-3652 3655
GETDN	32BE	1936 2120 2419 2421 -3649
GETFIL	30E8	1321 1695 1731 3374 -3390
GETIC1	30CF	1299 1358 1542 1895 2289 2327 2341 2818 3183 -3371
GETIC2	30D2	-3372
GETLIN	303C	1935 2054 2119 2418 2832 2896 2971 2990 -3293 3371
GETMLT	2EA5	3060 -3071
GETNAM	30DA	1896 -3377
GETNO	3224	2055 3189 3195 3208 3216 -3572
GETREC	0005	-69 496 1422 1573 3310
GHB	3231	-3577 3595
GLF	2168	1306 -1318
GND	3269	3581 3583 -3601
GOOD	17B8	253 475 -481
GOON	1A78	974 -982
GOTCOL	2EF6	3138 -3140
GT1	3127	3404 3406 3411 -3423
GT2	311F	-3419 3426
HATABS	031A	-45
HDBUF	15A0	-186 188 189 287 290 302 303 304 305 306 308 315 317 319 320 322 325
HDBUFH	0015	-188 189 228
HDBUFL	00A0	-189 226
HEXCON	32A5	3584 -3631
IBD	2ABF	-2475 2484 2485
IBDH	002A	2468 -2484 2485
IBDL	00BF	2469 -2485
ICAX1	034A	-101 201 422 581 715 1325 1412 1563 1680 1781 1818 2152 2854 2912 3240
ICAX2	034B	-102 1824
ICBAH	0345	-98 148 210 229 291 309 434 451 501 532 579 639 697 713 754 1211 1398 1416 1421 1569 1582 1672 1684 1724 1838 1839 1952 2150 2872 2928 2947 3250 3262 3278 3315 3510 3531
ICBAHZ	0025	-42
ICBAL	0344	-97 146 208 227 288 307 432 449 499 530 577 637 695 711 753 1209 1396 1414 1419 1567 1580 1670 1682 1722

		1835 1836 1950 2148 2870 2926 2945 3248
		3260 3276 3313 3512 3528
ICBALZ	0024	-41
ICBLH	0349	-100 214 233 321 338 438 505 643
		693 761 1215 1433 1578 1844 1851 1852
		2876 2939 2950 2951 3254 3271 3274 3319
		3506
ICBLL	0348	-99 212 231 318 336 436 503 641
		691 759 1213 1431 1576 1842 1849 1850
		1853 2874 2937 2948 2949 3252 3258 3269
		3272 3317 3333 3508
ICCOM	0342	-95 199 217 430 447 457 497 515
		575 686 689 700 709 734 741 763
		1327 1384 1394 1410 1423 1493 1565 1574
		1592 1607 1678 1779 1820 1831 1833 1867
		1872 1905 1954 2146 2157 2331 2345 2852
		2878 2914 2943 2957 2982 3234 3246 3311
		3487
ICDNO	0341	-94
ICDNOZ	0021	-40
ICHID	0340	-93
ICHIDZ	0020	-39
ICIDNO	002E	-43
ICSTA	0343	-96
IDD	2C35	-2666 2682 2683
IDDH	002C	2601 -2682 2683 2893
IDDL	0035	2600 -2683 2894
IDRD	223C	-1408 1488
IHP	3287	-3615 3624 3625
IHPH	0032	3606 -3624 3625
IHPL	0087	3605 -3625
INCOMP	2A99	-2453 2461
INISAV	179C	-465 519 521 666 668
INITAD	02E2	-37 353 355 378 600 602 604 607
		3209 3210 3211
INITIO	1976	472 -705 1191 2029 2062
INITQ	189F	-597 599 3181 3213
INITX	1593	149 155 162 -165 170
INMEM	19DB	769 -773
INSDES	2D92	2884 -2893
INTRVE	020A	-34 135 137 139 141
IOCB	0340	-92 93 94 95 96 97 98 99
		100 101 102
IOCB1	0010	-78
IOD	2B90	2576 -2578
IPTR	1F02	-1039 1391 1440 1481 1693 1697 2407 2548
		2642 2643 2690 2697 3300 3380 3391 3465
		3577 3579 3651 3669
IPTRH	001F	-2642 2643 2650
IPTRL	0002	-2643 2648
IRQEN	D20E	-811 865 1190
IS128	2AA3	2447 -2460
ISD	2C16	-2658 2679 2680

ISDH	002C
ISDISK	2D30
ISDL	0016
ISRODN	19E6
ISRSIR	1A23
JMPINT	1705
JMPNWC	2391
JMPRUN	1708
JMPTBL	0018
LBUFH	001D
LBUFL	00A4
LDFIL	291A
LDFX	294A
LDMEM	1939
LDMEM1	193F
LDMEM2	194A
LDND	19E2
LDST	19E0
LDSTH	0019
LDSTL	00E0
LEN	13F9
LENH	0013
LENL	00F9
LFMG	295B
LINE	1DA4
LKFIL	2970
LKMG	2985
LMARGN	0052
LMTR	1920
LNLF	1648
LOAD	15A9
LOADFG	159F
LOCK	0023
LOOKW1	2EEB
LOOKW2	2EEC
LOOKWC	2ED7
LRS	2B7D
LRS1	2B7A
MAXDEV	0021
MCONT	27B5
MDEND	1A7C
MDENDH	001A
MDENDL	007C
MDN1	228E
MDN2	229E
MDN3	22A6
MDUPBL	282C
MEMFLG	179E
MEMLDD	170B

```
 2505  2561 -2679  2680  2829  2968  2987
 2822 -2825
 2504  2560 -2680  2830  2969  2988
  138   140  -834
  134   136  -905
  361  -378
 1547 -1551
  244  -379
 -117  1199  1201  1243  1246
-1011  1012  3314
-1012  3312
 1162 -2288
 2299 -2308
  344   594   649   658  -676
  159   677  -679
  680  -685
  633   634  -784  2178  2184  3196  3197
  606   618   631   636   638  -780   782   783
 2174  2176  3190  3191  3199  3202  3275  3277
 -782   783  3261
 -783  3259
-3681  3683  3684
 2181 -3683  3684
 2179 -3684
 2288 -2316
-1010  1011  1012  3295  3347  3397  3402  3408
 3410  3413  3425  3436  3459  3468  3471  3578
 3652  3663
 1158 -2326
 2326 -2334
  -31  1184
 -658  2063
  222   224  -273
 -193  2297
 -185   192   246   247   334   335   339   469
 1182
  -74  2329
 3113 -3117
 3109  3111 -3118
 1549  1739  2835 -3106  3115
-2571  2591
 2552  2559 -2567  2599
  -44
 2088 -2097
 -986   988   989  1001
 -988   989   991
 -989   991
-1447  1454
 1449 -1458
-1462  1467
-2163  2166
  157  -466   468   483   622   676  2761  3025
  215   331   341   343  -382
```

```
MEMLO       02E7          -35
MEMORY    M 0000            0
MEMS        277F       -2076  2083
MEMSAV      279A        1162 -2083
MEMSG       27BD       -2101  2109  2110
MEMSGH      0027        2090 -2109  2110
MEMSGL      00BD        2089 -2110
MEMSVQ      1873         249   474  -573   679  2087
MEMTOP      02E5         -26  2518  2522  3030  3034
MENUSL      2006       -1222  1266  1290  1371  1386  1490  1594  1744
                        1804  1884  1907  1956  2011  2093  2192  2279
                        2308  2333  2347  2456  2536  2605  2766  2995
                        3159  3204  3339  3367  3493  3556
MENUSZ      1EF4       -1029  1197  1239
MERR        27BA       -2099
MES         249D       -1691
MLEN        1589       -3685  3687  3688
MLENH       0015         437   692 -3687  3688
MLENL       0089         435   690 -3688
MLT125      1F06       -1044  3043  3044  3050  3051  3053  3054  3059
                        3063  3071  3074  3078  3082
MNDUP       179F        -467   544   545
MNDUPH      0017         132  -544   545
MNDUPL      009F         130  -545
MNSL        20B6       -1290  1292  1293
MNSLH       0020       -1292  1293
MNSLL       00B6       -1293
MOUT        27AF        2086 -2092  2098
MOUT1       27B2        2061 -2093
MWRITE      1746        -420   481  2097
NAME        1730        -414   418   419
NAMEH       0017        -418   419   450   578
NAMEL       003B        -419   448   576
NARG        0000           0
NCA         273F       -2040  2045  2046
NCAH        0027        2009 -2045  2046
NCAL        003F        2008 -2046
NCDR        2ADE       -2486  2495  2496
NCDRH       002A        2454 -2495  2496
NCDRL       00DE        2453 -2496
NDF         21E5       -1372  1377  1378
NDFH        0021        1369 -1377  1378
NDFL        00E5        1368 -1378
NDOS        1D7C       -1001  1003  1004  1005  3685
NDOSH       001D         369   433   696 -1003  1004
NDOSL       007C         431   694 -1004
NDS         32F5       -3676  3690  3691
NDSH        0032        3192  3674 -3690  3691
NDSL        00F5        3673 -3691
NINTAD      2F88        3212 -3214
NLF         2940        2301 -2304
NMDUP       3305        2177  2183 -3680  3681  3685  3693  3694
NMDUPH      0033         371  2510  3028  3035 -3693  3694
```

NMDUPL	0005	2508	3026	3032	-3694				
NMSF	171B	-390	399	400					
NMSFH	0017	261	-399	400					
NMSFL	001B	260	-400						
NOCART	270A	-2008	2016	2020	2022				
NOCKSM	003C	-800	973	977					
NORM	2B30	-2533							
NORNAD	18DB	610	-621						
NOSYS	2418	1614	-1621						
NOT256	2CAD	2733	-2735						
NOTEND	1A12	847	-876						
NOTN	292E	2294	-2296						
NOTRAM	2714	2001	2005	-2015					
NOTSAM	2F11	3148	-3163						
NOTWC	24E1	1551	-1728						
NOTYET	1A50	928	-950						
NOWC	2D4A	2836	-2840						
NOWRPO	19EE	838	-843						
NRM	2C06	-2654	2676	2677					
NRMH	002C	2534	-2676	2677					
NRML	0006	2533	-2677						
NRUNAD	2FA0	2191	3207	3215	3221	-3223			
NSI	210D	-1268	1285	1286					
NSIH	0021	1264	-1285	1286					
NSIL	000D	1263	-1286						
NTFRAM	1A31	915	-919						
NTL	3195	-3475	3480	3481					
NTLH	0031	3456	-3480	3481					
NTLL	0095	3455	-3481						
NTOVRN	1A39	920	-927						
NTWRP1	1A64	960	-965						
NWA	2501	-1745	1756	1757					
NWAH	0025	1742	-1756	1757					
NWAL	0001	1741	-1757						
NWCIND	2527	1740	-1758						
NXS	2C84	2706	-2710						
NXTCHA	2EF9	-3146	3152						
ODMS	2574	1716	1764	1766	1773	1793	-1800	1807	1813
		2823							
OE	232E	-1518	1524	1525					
OEH	0023	-1524	1525	1801					
OEL	002E	-1525	1800						
OK	29DF	-2366	2376	2377					
OKH	0029	-2376	2377	2751	3011				
OKL	00DF	-2377	2750	3010					
OKTYP	2905	2263	-2269						
OKX	32BB	3635	3641	-3643					
OL	30AF	-3350	3357	3358					
OLH	0030	3344	-3357	3358					
OLL	00AF	3343	-3358						
ONE28	2A8C	2437	-2445						
OPDES	2581	1796	-1806						
OPDES1	2594	1725	1811	-1816					

OPDES3	2596	1815 -1817
OPEN	0003	-65 198 446 574 685 708 1326 1409 1564 1677 1778 1819 2144 2851 2913 3233
OPNDES	2DB2	2907 -2911
OPSRC	2544	1768 -1772
OPT	159E	129 -184 242 359 534 1380 1808 2291 2292 2295 2540 2557 2573 2595 2597 2603 3184 3224 3228 3235 3255 3435 3469
ORDWRT	000C	-106 580 714
OREST	1779	423 -446 458
OVRRUN	308E	-817 921
OWRIT	0008	-105 421
PAR	1D7C	-1007 1008 1009 1303 1304 1308 1309 1311 1313 1339 1365 1403 1545 1566 1568 1622 1701 1707 1762 1790 1921 2164 2168 2820 3106 3136 3147 3188 3420 3429 3432 3437 3472
PARH	001D	-1008 1009 1415 2919 3529
PARL	007C	-1009 1413 2918 3525
PDES	255E	1337 1704 1771 -1789
PDES1	256C	1342 -1797
PER	1EF5	-1030 2897 2972 3301 3324 3331 3336 3362 3458 3599 3649 3672
PERX	30C4	1322 1359 1759 1897 1941 2056 2121 2296 2328 2342 2423 2819 2833 2991 3217 -3362
PERX1	30CA	3363 -3365
PIOCB	31DD	1320 1689 1730 2172 3373 -3525
POKMSK	0010	-804 862 864 1187 1189
POPT	3186	3445 -3468
PRNTMS	19BE	262 488 492 -753 1228 1878 3498
PSRC	252D	-1761
PTR	1F01	-1038 1301 1317 1332 1425 1438 1439 1480 1543 1690 1699 2170 2403 2546 2639 2640 2692 2693 3186 3299 3379 3390 3466 3527
PTRH	001F	-2639 2640 2650
PTRL	0001	-2640 2648
PUTCHR	000B	-67 429 1832 2942 3245 3486
PUTREC	0009	-70 762
QWMG	28A8	2125 -2211 2221 2222
QWMGH	0028	2132 -2221 2222
QWMGL	00A8	2131 -2222
RAMLO	001A	-118 119 662 1245 1247 1249 1252 1255 1258 1259 1261 1262 2057 2058 2612 2614 2616 2619
RANGE	2103	1238 1240 -1263
RCNT	1EF7	-1032 2730 2740 2778 2782 3334 3340
RDDRC	15F7	-225 362
RDDRC1	1605	-231 311
RDFN	2269	-1429 1441 1479
RDLF	15C8	203 -206
RECVDN	0039	-803 938
RELDIN	192E	161 347 652 661 -666
RELONE	1A05	850 -862

```
RENAME      0020          -71  1903
RENFIL      2637         1158 -1894
REPLAC      2EB2         3072  3075 -3078
RET         1778         -444
RETSTA      2D07         2789 -2791
RMARGN      0053          -32  1186
RNMG        2652         1894 -1908
ROMTST      BFFD        -1996  1997  1999  2000  2003  2004  2007  2019
                         2021
RRDUP       1801          263  -518
RRDUP1      1813         -528   626
RSEC1       2C8D         2401  2575 -2717
RTCART      182B          509  -538
RTS         1647          193   195  -269   352   354
RUNAD       02E0          -38   194   196   379   612   614   616   619
                         2187  2189  3218  3219  3220
RUNQ        18BD          598  -609   611  2190  3182  3222
RVTOC       2A26         2269 -2393  2541
SAME        2AA8         2449 -2465
SAVFIL      2F2E         1162 -3179
SAVH        19DE         -776   778   779
SAVHH       0019         -778   779  3249
SAVHL       00DE         -779  3247
SAVX        1F00        -1037  1318  1334  1338  1468  1482  1544  1700
                         1735  1738  1789
SCMG        274B         1995 -2043
SCROL       31BB         1254  3303  3335  3499 -3504
SDD         2AFB         2467 -2504
SECSIZ      1F08        -1045  2260  2265  2431  2433  2439  2441  2448
                         2460  2520  2523  2580  2583  2732
SERIN       D20D         -810   929   950
SEROUT      D20D         -809   810   855   878
SETFLG      2D8F         2886 -2888
SETVBV      E45C          -22   724  2033  2037
SFLOAD      15A4         -191   537
SFMG        3018         3179 -3280
SHFLOK      02BE          -36  1230
SHMEN       209F        -1208  1234
SHT1        3249        -3587  3591
SIT         211A        -1275  1288  1289
SITH        0021         1227 -1288  1289
SITL        001A         1226 -1289
SKIP1       23FB         1599 -1604
SKIPWC      2DC8         2921 -2926
SKRES       D20A         -808   911
SKSTAT      D20F         -812   910
SMVRS       2919         2272 -2283
SRETRN      1A48          931  -937   978
SSTAT       1EF8        -1033  1846  1856
STACK     S 0000            0
STAK        0100          507  -993   995   996
STAKH       0001          500  -995   996
STAKL       0000          498  -996
```

STAREQ	0053	-76	2775						
STATUS	0030	-801	917	922	933				
STCAR	26EE	1158	-1995						
STDD	2BB6	-2595							
STDD1	2BCA	2586	-2603						
STDD2	2BBB	-2597	2604						
STE	3172	-3459	3464						
STLOAD	15A6	151	-192						
STOK	1658	235	-286						
STVEC	1F06	-1043	1044	2509	2511	2529	2531	2624	2626
SUSUAL	1A4C	-942	969	983					
SWATH	2C01	2613	-2650						
SWATL	2BFC	2611	-2648						
SWDP	1EF9	-1034	2543	2545	2547	2549	2551	2618	2621
SWLOP	2BD4	-2611	2623						
SYSED	0000	-104							
SYSLOP	2408	-1612	1616						
SYSVBV	E45F	-23	1987						
SYVBL	E45F	-1987	1989	1990					
SYVBLH	00E4	-1989	1990	2031					
SYVBLL	005F	-1990	2032						
T1	1F04	-1041	1042	2521	2524	2528	2530	2587	2589
		3575	3576	3588	3589	3592	3593	3602	3603
TEMP	1799	454	-459						
TMD	3277	-3608	3613	3614					
TMDH	0032	3597	-3613	3614					
TMDL	0077	3596	-3614						
TSTDOS	2EED	1736	2845	-3135					
TSTVER	28F3	1928	2127	-2258					
TWODRV	1F0B	-1048	2417	2472	2558	2598			
TYQ	22F7	-1495	1502	1503					
TYQH	0022	1388	-1502	1503					
TYQL	00F7	1387	-1503						
ULFIL	2998	1158	-2340						
ULMG	29AD	2340	-2348						
UNLOCK	0024	-75	2343						
UNNO	1EF6	-1031	1926	2122	2261	2420	2435	2465	2717
USEBUF	2EC4	1300	3018	-3089					
USEDB4	2E5A	3009	-3023						
USEPGM	2E41	1675	1760	2825	-3008				
USRDOS	1700	-46							
VECTR	19E4	601	603	613	615	-785			
VFM	26D0	-1963	1978	1979					
VFMH	0026	1943	-1978	1979					
VFML	00D0	1942	-1979						
WAITIM	19A4	-725	726						
WARMST	0008	-30	143	169	471	477	511	682	2748
		2760	3008	3024					
WBMG	2892	-2201	2208	2209					
WBMGH	0028	2141	-2208	2209					
WBMGL	0092	2140	-2209						
WBOOT	27D9	1162	-2115						
WBX	2872	2136	-2192						

WCBUF	2344	-1535 1579 1581 1585 1612 1632 1642 1648
WCBUF2	2364	-1539 1626 1635 1646 1651 1658 1669 1671
		1681 1683 1712 1720 1721 1723 2923 2924
WCBUFL	0014	-1534 1575
WCDUPS	2D4F	1673 -2850
WCFLAG	2341	1225 -1531 1555 1665 1881 2920 2985
WCGOT	23EE	1587 1589 -1597
WCGOT1	2422	1624 -1626
WCINIT	2396	-1555 2838
WCOPY	2476	1666 -1675
WCOPY0	24BA	1703 -1706
WCOPY1	24C4	1709 -1712
WCOPY2	24CF	1713 -1719
WCOPYL	239E	-1559 1617 1883 2992
WCOPYM	2358	-1536 1661 1662
WCOPYR	23BC	-1573 1602
WCSKP1	2342	-1532 1557 1597 1604
WCSKP2	2343	-1533 1560 1598 1601
WDR	2FF5	-3267
WDR1	1893	-592 623 625 3194 3279
WDR2	189A	593 -595
WDRH	2FFC	2182 3205 -3270
WDRL	2FF7	2180 3200 -3268
WEX	3015	-3279
WHD	26B9	1934 -1957 1975 1976
WHDH	0026	-1975 1976
WHDL	00B9	-1976
WHEAD	2FF2	3256 -3263
WRMSTR	E474	-50 163
WRVEC	18EF	608 620 -631
WVD	28D5	-2229 2238 2239
WVDH	0028	-2238 2239 2277
WVDL	00D5	-2239 2276
XBLK	2B74	-2562 2602
XITVBV	E462	-24 1991
XTVBL	E462	-1991 1993 1994
XTVBLH	00E4	-1993 1994 2035
XTVBLL	0062	-1994 2036
ZAP	31D3	-3515 3520 3521
ZAPH	0031	3509 -3520 3521
ZAPL	00D3	3511 -3521