DISK UTILITIES

Disk UTILITIES

```
                              ; SYSRES  =       0      ***USER RAM RESIDENT VERSION***
        0001                  SYSRES   =       1      ***SYSTEM RAM RESIDENT VERSION***
        E456                  CIO      =       $E456
        E453                  DKIIND   =       $E453
        E45C                  SETVBV   =       $E45C
        E45F                  SYSVBV   =       $E45F
        E462                  XITVBV   =       $E462
        02E5                  MEMTOP   =       $2E5
        D20E                  IRGEN    =       $D20E
        0011                  BRKKEY   =       $11
        0010                  POKMSK   =       $10
        000A                  DOSVEC   =       $A
        0008                  WARMST   =       8
        BFFA                  CARTST   =       $BFFA
        02E7                  MEMLO    =       $2E7
        02BE                  SIIFLOK  =       $2BE
        02E0                  RUNAD    =       $2E0
        0020                  ICHIDZ   =       $20
        0021                  ICDNOZ   =       $21
        0024                  ICBALZ   =       $24
        0025                  ICBAHZ   =       $25
        002E                  ICIDNO   =       $2E
        0021                  MAXDEV   =       $21
        031A                  HATABS   =       $31A
                              ; CIO SUBROUTINES
        E63D                  COMENT   =       $E63D
        E689                  GOHAND   =       $E689
        E69E                  DEVSRC   =       $E69E
                              .TITLE  "COLLEEN DISK UTILITIES"
                              ;
                              ;
                              ;****************************************************************
                              ;        FILENAME =DK1:DUPV.SRC
                              ;****************************************************************
                              ;
                              ;
                              ;        DISK UTILITY ROUTINE FOR COLLEEN O.S.
                              ;
        HILO    .MACRO  &1
        &1H      =       &1/256
        &1L      =       -256*&1H+&1
                              .ENDM
                              ;
                              ;        ZERO PAGE VARIABLES
                              ;
                              *=$18
        001B    JMPTBL       *=*+2
        001A    RAMLO        *=*+2
                              ;
                              ;
                              ;
        1700    USRDOS       =       $1700
        0700    I-MS         =       $700
        0700    FMINIT       =       FMS+$88
```

```
                                    .IF      SYSRES
                           *=DOS
1300 A9 25                 LDA      #DOSDBL
1302 85 0A                 STA      DOSVEC
1304 A9 16                 LDA      #DOSDSH
1306 85 0B                 STA      DOSVEC+1
1308 20 88 07              JSR      FMINIT
130B A5 08                 LDA      WARMST  ON COLDSTART, LOAD AUTO.SYS
130D D0 0D                 BNE      INITX
130F A9 1D                 LDA      #AFL
1311 8D 54 03              STA      ICBAL+$10
1314 A9 13                 LDA      #AFH
1316 8D 55 03              STA      ICBAH+$10
1319 4C 0B 24   LOADV      JMP      LOAD    LOAD VECTOR, ALSO INIT LOAD AND EXIT.
131C 60         INITX      RTS
131D 44 31 3A   AF         .BYTE    "D1:AUTO.SYS",CR
1320 41 55 54
1323 4F 2E 53
1326 59 53 9B
                           HILO     AF
0013            AFH    =    AF/256
001D            AFL    =    -256*AFH+AF
                           .ENDIF
                           .IF      SYSRES-1
                           *=USRDOS
                LOADV  =    $1319
                           .ENDIF
                !
007B            CR     =    $7B
001C            CUP    =    $1C
001D            CDN    =    $1D
001E            CLF    =    $1E
001F            CRT    =    $1F
007C            DLL    =    $9C
007D            CLSCR  =    $7D
                !
0003            OPEN   =    $03
000C            CLOSE  =    $0C
000A            PUTCHR =    $0A
0006            GETCHR =    $06
0004            GETRCD =    $04
000B            PUTRCD =    $0B
0020            RENAME =    $20
0021            DELETE =    $21
00FE            FORMAT =    $FE
0023            LOCK   =    $23
0024            UNLOCK =    $24
                !
0001            TEXT   =    1
                !
0300            DCB    =    $300
0301            DUNIT  =    DCB+1
0302            DCOMND =    DCB+2
0303            DSTATS =    DCB+3
0304            DBUFLO =    DCB+4
0305            DBUFHI =    DCB+5
030A            DCLO   =    DCB+$A
```

```
030B              DBHI    =      DCB+$B
                  ;
0340              IOCB    =      $340
0340              ICHID   =      IOCB+0
0341              ICDNO   =      IOCB+1
0342              ICCOM   =      IOCB+2
0343              ICBTA   =      IOCB+3
0344              ICBAL   =      IOCB+4
0345              ICBAH   =      IOCB+5
0348              ICBLL   =      IOCB+8
0349              ICBLH   =      IOCB+9
034A              ICAX1   =      IOCB+10
034B              ICAX2   =      IOCB+11
1329              PAR     *=*+40  PARAMETER AREA
0013              PARH    =      PAR/256
0029              PARL    =      -256*PARH+PAR
1351              LINE    *=*+80  TYPEIN LINE BUFFER
0013              LBUFH   =      LINE/256
0051              LBUFL   =      -256*LBUFH+LINE
13A1              DBUF    *=*+$100       DATA BUFFER FOR COPY
1421              DB1     =      DBUF+$80
139E              DB3     =      DBUF-3
                  HILO    DBUF
0013              DBUFH   =      DBUF/256
00A1              DBUFL   =      -256*DBUFH+DBUF
                  HILO    DB1
0014              DB1H    =      DB1/256
0021              DB1L    =      -256*DB1H+DB1
                  HILO    DB3
0013              DB3H    =      DB3/256
009E              DB3L    =      -256*DB3H+DB3
0000              DBLL    =      0
0001              DBLH    =      1         DATA BUFFER LENGTH=$100
14A1              MENUSZ  *=*+1
14A2              PER     *=*+1
14A3              UNNO    *=*+1
14A4              RCNT    *=*+1
14A5              SSTAT   *=*+1
14A6              SHDP    *=*+5
14AB              CSRC    *=*+1
14AC              CDES    *=*+1
14AD              SAVX    *=*+1
14AE              PTR     *=*+1
14AF              IPTR    *=*+1
14B0              CTR     *=*+1
14B1              OPT     *=*+1
14B2              T1      *=*+2
14B4              STVEC   *=*+2   A TEMP OF SOME KIND
14B6              MNA     *=*+2   MENU ADDRESS
14B8              MNL     *=*+2   MENU LENGTH
14BA              RDM     *=*+1   FF TO REDISPLAY MENU
14BB              DTH     =*
                  HILO    DTH
0014              DTHH    =      DTH/256
00BB              DTHL    =      -256*DTHH+DTH
14BB              LDST    *=*+2
14BD              LDND    *=*+2
                  ;
                  ;
```

```
14BF 84 09        BAVH     . BYTE    $84, $09
                  HILO     BAVH
0014              BAVHH    =        BAVH/256
00BF              BAVHL    =        -256*BAVHH+BAVH
14C1 45 3A 9B     EDN      . BYTE   "E:", CR
0014              EDH      =        EDN/256
00C1              EDL      =        -256*EDH+EDN
14C4 4B 3A 9B     KDN      . BYTE   "K:", CR
                  HILO     KDN
0014              KDNH     =        KDN/256
00C4              KDNL     =        -256*KDNH+KDN

14C7 7D           DMENU    . BYTE   CLSCR
14C8 44 49 53              . BYTE   "DISK OPERATING SYSTEM   9/24/79", CR
14CB 4B 20 4F
14CE 50 45 52
14D1 41 54 49
14D4 4E 47 20
14D7 53 59 53
14DA 54 45 4D
14DD 20 20 20
14E0 39 2F 32
14E3 34 2F 37
14E6 39 9B
14E8 43 4F 50              . BYTE   "COPYRIGHT 1979 ATARI", CR, CR
14EB 59 52 49
14EE 47 48 54
14F1 20 31 39
14F4 37 39 20
14F7 41 54 41
14FA 52 49 9B
14FD 9B
14FE 41 2E 20              . BYTE   "A. DISK DIRECTORY I. FORMAT DISK", CR
1501 44 49 53
1504 4B 20 44
1507 49 52 45
150A 43 54 4F
150D 52 59 20
1510 49 2E 20
1513 46 4F 52
1516 4D 41 54
1519 20 44 49
151C 53 4B 9B
151F 42 2E 20              . BYTE   "B. RUN CARTRIDGE  J. DUPLICATE DISK", CR
1522 52 55 4E
1525 20 43 41
1528 52 54 52
152B 49 44 47
152E 45 20 20
1531 4A 2E 20
1534 44 55 50
1537 4C 49 43
153A 41 54 45
153D 20 44 49
1540 53 4B 9B
1543 43 2E 20              . BYTE   "C. COPY FILE       K. BINARY SAVE", CR
1546 43 4F 50
1549 59 20 46
```

```
154F 20 20 20
1552 20 20 20
1555 4B 2E 20
1558 42 49 4E
155B 41 52 59
155E 20 53 41
1561 56 45 9B          .BYTE   "D. DELETE FILE(S) L. BINARY LOAD",CR
1564 44 2E 20
1567 44 45 4C
156A 45 54 45
156D 20 46 49
1570 4C 45 28
1573 53 29 20
1576 4C 2E 20
1579 42 49 4E
157C 41 52 59
157F 20 4C 4F
1582 41 44 9B
1585 45 2E 20          .BYTE   "E. RENAME FILE    M. RUN AT ADDRESS",CR
1588 52 45 4E
158B 41 4D 45
158E 20 46 49
1591 4C 45 20
1594 20 20 20
1597 4D 2E 20
159A 52 55 4E
159D 20 41 54
15A0 20 41 44
15A3 44 52 45
15A6 53 53 9B
15A9 46 2E 20          .BYTE   "F. LOCK FILE      N. DEFINE DEVICE",CR
15AC 4C 4F 43
15AF 4B 20 46
15B2 49 4C 45
15B5 20 20 20
15B8 20 20 20
15BB 4E 2E 20
15BE 44 45 46
15C1 49 4E 45
15C4 20 44 45
15C7 56 49 43
15CA 45 9B
15CC 47 2E 20          .BYTE   "G. UNLOCK FILE    O. DUPLICATE FILE",CR
15CF 55 4E 4C
15D2 4F 43 4B
15D5 20 46 49
15D8 4C 45 20
15DB 20 20 20
15DE 4F 2E 20
15E1 44 55 50
15E4 4C 49 43
15E7 41 54 45
15EA 20 46 49
15ED 4C 45 9B
15F0 48 2E 20          .BYTE   "H. WRITE DOS FILE",CR
15F3 57 52 49
15F6 54 45 20
15F9 44 4F 53
15FC 20 46 49
```

```
1602 1D 1D 1D              .BYTE   CDN, CDN, CDN, CDN, CDN
1605 1D 1D
1607              DMEND     =*
0140              DULEN     =       DMEND-DMENU
                  HILO      DULEN
0001              DULENH    =       DULEN/256
0040              DULENL    =       -256*DULENH+DULEN
                  HILO      DMENU
0014              DMENUH    =       DMENU/256
00C7              DMENUL    =       -256*DMENUH+DMENU
                  ;
1607 2D 17 F9     DUJPT     .WORD   DIRLST, STCAR, CPYFIL, DELFIL, RENFIL, LKFIL, ULFIL
160A 1A 36 19
160D BA 17 54
1610 1A 08 1D
1613 30 1D
1615 39 1C 87              .WORD   WBOOT, FMTDSK, DUPDSK, SAVFIL, LDFIL, BRUN, DEFDEV
1618 1A CC 1D
161B DB 20 B1
161E 1C 2C 1B
1621 76 1D
1623 1F 20                 .WORD   DUPFIL
                  HILO      DUJPT
0016              DUJPTH    =       DUJPT/256
0007              DUJPTL    =       -256*DUJPTH+DUJPT
000F              DUNUM     =       15      NUMBER OF FUNCTIONS

                  ;**************************************************************
                  ;       DISK UTILITY ROUTINE STARTS HERE
                  ;**************************************************************



                  ;
                  ;
                  ;       DISK OPERATING SYS MONITOR
                  ;
1625 A9 FF        DOSOS     LDA     #$FF
                  HILO      DOSOS
0016              DOSOSH    =       DOSOS/256
0025              DOSOSL    =       -256*DOSOSH+DOSOS
1627 85 11                  STA     BRKKEY
1629 85 08                  STA     WARMST
162B A5 10                  LDA     POKMSK  ENABLE BREAK INTERRRUPTS
162D 09 80                  ORA     #$80
162F 85 10                  STA     POKMSK
1631 8D 0E D2               STA     IRGEN
1634 20 BC 21               JSR     INITIO  CLOSE FILES
1637 A9 03                  LDA     #OPEN
1639 8D 72 03               STA     ICCOM+$30       K: IOCB
163C A2 30                  LDX     #$30    K: IOCB
163E A9 04                  LDA     #$4
1640 9D 4A 03               STA     ICAX1,X OPEN READ
1643 A9 14                  LDA     #KDNH
1645 9D 45 03               STA     ICBAH,X
1648 A9 C4                  LDA     #KDNL
164A 9D 4A 03               STA     ICBAL,X
```

```
                                    DISK UTILITY MONITOR

        1650                 DSKUTL
        1650 A9 OF           DU1     LDA     #DUNUM
        1652 8D A1 14                STA     MENUSZ          SET MENU SIZE.
        1655 A9 07                   LDA     #DUJPTL
        1657 85 18                   STA     JMPTBL
        1659 A9 16                   LDA     #DUJPTH
        165B 85 19                   STA     JMPTBL+1        SET UP JUMP TABLE ADDRESS
        165D A9 C7                   LDA     #DMENUL
        165F 8D B6 14                STA     MNA             SET UP FOR MENU COPY
        1662 A9 14                   LDA     #DMENUH
        1664 8D B7 14                STA     MNA+1
        1667 A9 40                   LDA     #DULENL
        1669 8D B8 14                STA     MNL
        166C A9 01                   LDA     #DULENH
        166E 8D B9 14                STA     MNL+1
                                    FALL THRU TO MENU SELECT

                                    MENU SELECT MONITOR -- VECTORS TO ROUTINE SELECTED FROM MENU.

        1671 A9 00           SHMEN   LDA     #0              CLEAN PAUSE AND SHOW MENU FLAG
        1673 8D BA 14                STA     RDM
        1676 AD B6 14                LDA     MNA
        1679 8D 44 03                STA     ICBAL
        167C AD B7 14                LDA     MNA+1
        167F 8D 45 03                STA     ICBAH
        1682 AD B8 14                LDA     MNL
        1685 8D 48 03                STA     ICBLL
        1688 AD B9 14                LDA     MNL+1
        168B 8D 49 03                STA     ICBLH
        168E 20 56 23                JSR     DSPMSG          SHOW MENU

                                    SELECT ITEM FROM MENU
                                    FUNCTION COME HERE WHEN THEY ARE DONE.

        1691 A2 FF           MENUSL  LDX     #$FF            RESET STACK AT THIS POINT
        1693 9A                      TXS
        1694 2C BA 14                BIT     RDM             IF WE SHOULD SHOW MENU
        1697 30 62                   BMI     PSMEN           GO DO IT
        1699 A9 EF                   LDA     #SITL           SELECT ITEM MESSAGE
        169B A2 16                   LDX     #SITH
        169D 20 61 23                JSR     DSPLIN
        16A0 A9 40                   LDA     #$40            MAKE SURE UPPER CASE
        16A2 8D BE 02                STA     SHFLOK
        16A5 20 33 22                JSR     CHRGET          GO GET KEYBOARD CHAR.
        16A8 38                      SEC
        16A9 E9 41                   SBC     #'A             CONVERT ASCII CHAR. TO BINARY # & SUBTRACT 1
        16AB 30 2B                   BMI     RANGE           IF ASCII CHAR NOT A #, GO READ AGAIN
        16AD CD A1 14                CMP     MENUSZ          IS THE # ENTERED > MENU SIZE?
        16B0 10 26                   BPL     RANGE           IF YES, GO READ AGAIN.
        16B2 0A                      ASL     A
        16B3 A8                      TAY                     SET INDEX TO (MENU # - 1) * 2
        16B4 B1 18                   LDA     (JMPTBL),Y
        16B6 C8                      INY
        16B7 85 1A                   STA     RAMLO           GET STRING POINTER
```

```
16BB 85 1B              STA     RAMLO+1
16BD A0 01              LDY     #1        LOAD STRING POINTER INTO REGISTERS
16BF B1 1A              LDA     (RAMLO),Y       FOR DSPLIN
16C1 AA                 TAX
16C2 88                 DEY
16C3 B1 1A              LDA     (RAMLO),Y
16C5 20 61 23           JSR     DSPLIN   PRINT MODULES INITIAL STRING
16C8 A5 1A              LDA     RAMLO    INC BY 2 TO POINT PAST STRING POINTER
16CA 18                 CLC
16CB 69 02              ADC     #2
16CD 85 1A              STA     RAMLO
16CF A5 1B              LDA     RAMLO+1
16D1 69 00              ADC     #0       CARRY
16D3 85 1B              STA     RAMLO+1        PUT HI BYTE.
16D5 6C 1A 00           JMP     (RAMLO)        JUMP TO ROUTINE SELECTED BY MENU.
16D8 A9 E2      RANGE   LDA     #NSIL
16DA A2 16              LDX     #NSIH
16DC 20 61 23           JSR     DSPLIN   NO SUCH ITEM MESSAGE
16DF 4C 91 16           JMP     MENUSL
16E2 4E 4F 20   NSI     .BYTE   "NO SUCH ITEM",CR
16E5 53 55 43
16E8 48 20 49
16EB 54 45 4D
16EE 9B
16EF 53 45 4C   SIT     .BYTE   "SELECT ITEM",CR
16F2 45 43 54
16F5 20 49 54
16F8 45 4D 9B
                HILO    NSI
0016            NSIH    =       NSI/256
00E2            NSIL    =       -256*NSIH+NSI
                HILO    SIT
0016            SITH    =       SIT/256
00EF            SITL    =       -256*SITH+SIT
1691            MNSL    =       MENUSL
                HILO    MNSL
0016            MNSLH   =       MNSL/256
0091            MNSLL   =       -256*MNSLH+MNSL
                ;
                ; PAUSE AND REDISPLAY MENU
                ;
16FB A9 00      PSMEN   LDA     #0
16FD 8D 49 03           STA     ICBLH
1700 A9 15              LDA     #21
1702 8D 48 03           STA     ICBLL
1705 A9 17              LDA     #PMESH
1707 8D 45 03           STA     ICBAH
170A A9 18              LDA     #PMESL
170C 8D 44 03           STA     ICBAL
170F 20 56 23           JSR     DSPMSG
1712 20 33 22           JSR     CHRGET
1715 4C 71 16           JMP     SHMEN
1718 9B 54 59   PMES    .BYTE   CR,"TYPE RETURN FOR MENU"
171B 50 45 20
171E 52 45 54
1721 55 52 4E
1724 20 46 4F
1727 52 20 4D
172A 45 4E 55
                HILO    PMES
```

```
0017                   PMESH    =      PMES/256
0018                   PMESL    =      -256*PMESH+PMES
                   ;
                   ;
                   ;         DIRECTORY LISTING ROUTINE
                   ;
                   ;*****************
172D 98 17         DIRLST   .WORD  DLMG
172F 20 84 22               JSR    GETICI
1732 AE AE 14               LDX    PTR
1735 A9 9B                  LDA    #CR
1737 9D 28 13               STA    PAR-1,X ASSURE GOOD TERM
173A BD 27 13               LDA    PAR-2,X LAST CHAR OF SEARCH SPEC
173D C9 3A                  CMP    #':      IF COLON, ADD *,*
173F D0 18                  BNE    OLF
1741 A9 2A                  LDA    #'*
1743 9D 28 13               STA    PAR-1,X
1746 9D 2A 13               STA    PAR+1,X
1749 A9 2E                  LDA    #'.
174B 9D 29 13               STA    PAR,X
174E A9 9B                  LDA    #CR
1750 9D 2B 13               STA    PAR+2,X
1753 E8                     INX
1754 E8                     INX
1755 E8                     INX
1756 8E AE 14               STX    PTR
1759 8E AD 14      OLF      STX    SAVX
175C A2 20                  LDX    #$20
175E 20 B1 23               JSR    PIOCB
1761 20 9E 22               JSR    GETFIL
1764 20 79 22               JSR    PERX
1767 A9 06                  LDA    #6       READ DIR INFO
1769 A2 10                  LDX    #$10
176B 9D 4A 03               STA    ICAX1,X
176E A9 03                  LDA    #OPEN    OPEN
1770 9D 42 03               STA    ICCOM,X
1773 8E AB 14               STX    CSRC     COPY SOURCE-DIRECTORY INFO
1776 E0 10                  CPX    #$10
1778 D0 01                  BNE    *+3
177A 20 C2 23               JSR    CIOCL
177D AD AE 14               LDA    PTR
1780 38                     SEC
1781 ED AD 14               SBC    SAVX
1784 C9 03                  CMP    #3       IF ONLY 3 CHARS, IS "D:"CR, USE DEFAULT
1786 F0 03                  BEQ    DLST1
1788 4C 98 19      DLST0    JMP    PDES     GO INTO COPY
178B AE AD 14      DLST1    LDX    SAVX
178E BD 29 13               LDA    PAR,X
1791 C9 44                  CMP    #'D
1793 D0 F3                  BNE    DLST0
1795 4C A2 19               JMP    PDES1    GO INTO COPY WITH DES='E:'
1798 44 49 52      DLMG     .BYTE  "DIRECTORY--SEARCH SPEC,LIST FILE?",CR
179B 45 43 54
179E 4E 52 59
17A1 2D 2D 53
17A4 49 41 52
17A7 43 48 20
17AA 53 50 45
17AD 43 2C 4C
```

```
          17B3 20 46 49
          17B6 4C 45 3F
          17B9 9B

                        ; *****************
                        ;
                        ;
                        ;           DELETE FILE ROUTINE
                        ;
                        ; *****************
          17BA FC 18    DELFIL  .WORD   DEMO
          17BC 20 84 22         JSR     GETIC1
          17BF 20 79 22         JSR     PERX    EXIT IF PARAM ERRORS
          17C2 AD 29 13         LDA     PAR     GET DEVICE
          17C5 C9 44            CMP     #'D     ONLY ALLOW DELETE FOR D:
          17C7 F0 1A            BEQ     DF1
          17C9 A9 D3            LDA     #NDFL
          17CB A2 17            LDX     #NDFH
          17CD 20 61 23         JSR     DSPLIN
          17D0 4C 91 16         JMP     MENUSL
          17D3 4E 4F 54    NDF  .BYTE   "NOT A DISK FILE",CR
          17D6 20 41 20
          17D9 44 49 53
          17DC 4B 20 46
          17DF 49 4C 45
          17E2 9B
                        HILO    NDF
          0017          NDFH    =       NDF/256
          00D3          NDFL    =       -256*NDFH+NDF
          17E3 A2 10    DF1     LDX     #$10
          17E5 AD B1 14         LDA     OPT
          17E8 C9 4E            CMP     #'N     IF OPTION=N, NO QUERY
          17EA D0 0B            BNE     DWQ     NO, DELETE WITH QUERY
          17EC A9 21            LDA     #DELETE
          17EE 9D 42 03         STA     ICCOM,X
          17F1 20 C2 23         JSR     CIOCL
          17F4 4C 91 16         JMP     MENUSL
          17F7 A9 E6    DWQ     LDA     #TYQL
          17F9 A2 18            LDX     #TYQH
          17FB 20 61 23         JSR     DSPLIN  SAY TYPE Y TO DELETE...
          17FE A9 00            LDA     #0
          1800 8D AF 14         STA     IPTR    HOW MANY FILES TO SKIP, NONE AT FIRST
          1803 A2 20            LDX     #$20    SET UP DELETE IOCB
          1805 A9 21            LDA     #DELETE
          1807 9D 42 03         STA     ICCOM,X
          180A A9 9E            LDA     #DB3L
          180C 9D 44 03         STA     ICBAL,X
          180F A9 13            LDA     #DB3H
          1811 9D 45 03         STA     ICBAH,X
          1814 A9 44            LDA     #'D
          1816 8D 9E 13         STA     DBUF-3
          1819 A9 3A            LDA     #':
          181B 8D A0 13         STA     DBUF-1
          181E AD 2A 13         LDA     PAR+1   DEVICE NUMBER OR : FROM OP INPUT
          1821 C9 3A            CMP     #':
          1823 D0 02            BNE     *+4
          1825 A9 31            LDA     #'1
          1827 8D 9F 13         STA     DBUF-2  KLUDGE KLUDGE KLUDGE
          182A A2 10    IBRD    LDX     #$10
```

```
182E 9D 42 03              STA     ICCOM,X
1831 A9 06                 LDA     #6
1833 9D 4A 03              STA     ICAX1,X  DIR READ OPEN
1836 A9 29                 LDA     #PARL
1838 9D 44 03              STA     ICBAL,X
183B A9 13                 LDA     #PARH
183D 9D 45 03              STA     ICBAH,X
1840 20 C2 23              JSR     CIOCL
1843 A9 A1                 LDA     #DBUFL
1845 9D 44 03              STA     ICBAL,X
1848 A9 13                 LDA     #DBUFH
184A 9D 45 03              STA     ICBAH,X
184D A9 04                 LDA     #GETRCD
184F 9D 42 03              STA     ICCOM,X
1852 A9 00                 LDA     #0
1854 8D AE 14              STA     PTR     HOW MANY FILES WE HAVE SKIPPED
                    ; READ FILENAME FROM DIR, QUERY AND DELETE
1857 A2 10          RDFN   LDX     #$10
1859 A9 00                 LDA     #0
185B 9D 48 03              STA     ICBLL,X
185E A9 01                 LDA     #1
1860 9D 49 03              STA     ICBLH,X
1863 20 C2 23              JSR     CIOCL   READ A LINE FROM DIRECTORY
1866 AD A2 13              LDA     DBUF+1  IF FILE LINE, THIS IS BLANK
1869 C9 20                 CMP     #'
186B D0 68                 BNE     DELX    THIS IS FREE BLOCKS LINE
186D EE AE 14              INC     PTR     COUNT THIS FILE
1870 AD AE 14              LDA     PTR     HAVE WE SKIPPED ENUF YET
1873 CD AF 14              CMP     IPTR
1876 30 DF                 BMI     RDFN    BR IF NO
1878 A2 00                 LDX     #0      PUT PTR
187A A0 02                 LDY     #2      GET PTR
                    ;MASSAGE DELETE FILE NAMES
187C B9 A1 13       MDN1   LDA     DBUF,Y
187F C9 20                 CMP     #'      END OF FILENAME
1881 F0 09                 BEQ     MDN2
1883 9D A1 13              STA     DBUF,X
1886 E8                    INX
1887 C8                    INY
1888 E0 08                 CPX     #8
188A 30 F0                 BMI     MDN1
                    ;FILENAME IS MOVED, PUT .EXT
188C A9 2E          MDN2   LDA     #'.
188E 9D A1 13              STA     DBUF,X
1891 E8                    INX
1892 A0 0A                 LDY     #$10    WHERE EXT IS
1894 B9 A1 13       MDN3   LDA     DBUF,Y
1897 9D A1 13              STA     DBUF,X
189A C8                    INY
189B E8                    INX
189C C0 0D                 CPY     #13
189E 30 F4                 BMI     MDN3
18A0 8E AD 14              STX     SAVX    PUT CR HERE LATER
18A3 A9 3F                 LDA     #'?     FOR QUERY
18A5 9D A1 13              STA     DBUF,X
18A8 E8                    INX
18A9 A9 9D                 LDA     #CR
```

```
        18B2 20 61 23          JSR     DSPLIN   GO ASK ABOUT THIS FILE
        18B5 20 33 22          JSR     CHROET
        18B8 C9 59             CMP     #'Y
        18BA D0 9B             BNE     RDFN     GO DO NEXT FILENAME
        18BC AD AE 14          LDA     PTR      NUMBER FILES WE HAVE GONE THRU SO FAR
        18BF 8D AF 14          STA     IPTR     IS NEW NUMBER TO SKIP.
        18C2 AE AD 14          LDX     SAVX
        18C5 A9 9B             LDA     #CR
        18C7 9D A1 13          STA     DBUF, X
        18CA A2 20             LDX     #$20     DELETE IOCB
        18CC 20 C2 23          JSR     CIOCL
        18CF 20 DB 18          JSR     CLOS1
        18D2 4C 2A 18          JMP     IDRD     CLOSE AND REOPEN DIR READ FILE
        18D5 20 DB 18   DELX   JSR     CLOS1    CLOSE DIR READ FILE
        18DB 4C 91 16          JMP     MENUSL
        18DB A2 10      CLOS1  LDX     #$10
        18DD A9 0C             LDA     #CLOSE
        18DF 9D 42 03          STA     ICCOM, X
        18E2 20 C2 23          JSR     CIOCL
        18E5 60               RTS
        18E6 54 59 50   TYG    .BYTE   "TYPE ",$22,"Y",$22," TO DELETE...",CR
        18E9 45 20 22
        18EC 59 22 20
        18EF 54 4F 20
        18F2 44 45 4C
        18F5 45 54 45
        18F8 2E 2E 2E
        18FB 9B
                        HILO   TYG
        0018     TYGH   =       TYG/256
        00E6     TYGL   =       -256*TYGH+TYG
        18FC 44 45 4C   DEMO   .BYTE   "DELETE FILE SPEC",CR
        18FF 45 54 45
        1902 20 46 49
        1905 4C 45 20
        1908 53 50 45
        190B 43 9B
                        ; ****************
                        ;
                        ;
                        ;        COPY FILE ROUTINE
                        ;
                        ; ****************
        190D 43 4F 50   CPMO   .BYTE   "COPY--FROM, TO?",CR
        1910 59 2D 2D
        1913 46 52 4F
        1916 4D 2C 20
        1919 54 4F 3F
        191C 9B
        191D 4F 50 54   OE     .BYTE   "OPTION DOESNT MAKE SENSE",CR
        1920 49 4F 4E
        1923 20 44 4F
        1926 45 53 4E
        1929 54 20 4D
        192C 41 4B 45
        192F 20 53 45
        1932 4E 53 45
        1935 9B
                        HILO   OE
```

```
          001D              OEL      =       -256*0EH+DE
     1936 0D 19             CPYFIL   .WORD   CPMO
     1938 20 94 22                   JBR     GETIC1
     193B AD B1 14                   LDA     OPT
     193E 48                         PHA
     193F AE AE 14                   LDX     PTR
     1942 CA                         DEX
     1943 A9 9B                      LDA     #CR     TERM FIRST FILENAME
     1945 9D 29 13                   STA     PAR.X
     1948 A2 20                      LDX     #$20    IOCB 3
     194A 20 B1 23                   JBR     PIOCB
     194D AD AE 14                   LDA     PTR
     1950 8D AD 14                   STA     SAVX
     1953 20 9E 22                   JBR     GETFIL  GET SECOND FILENAME
     1956 20 79 22                   JBR     PERX    IF PARAM ERRS, EXIT
     1959 68             PBRC        PLA
     195A AA                         TAX
     195B AD 29 13                   LDA     PAR     GET 1ST LETR OF PARAM
     195E C9 4B                      CMP     #'K     K AND E AS SOURCE ARE SPECIAL
     1960 F0 4B                      BEG     ODMB    K: GETS 'OPTION DOESNT MAKE SENSE' FOR NOW
     1962 C9 45                      CMP     #'E
     1964 D0 0B                      BNE     OPSRC   IF NO THEN OPEN SOURCE FILE
     1966 CE BA 14                   DEC     RDM     E: SO RESHOW MENU AFTER COPY
     1969 A2 00                      LDX     #0
     196B 8E AB 14                   STX     CBRC
     196E 4C 98 19                   JMP     PDEB
     1971 C9 53          OPBRC       CMP     #'B
     1973 F0 38                      BEG     ODMB    B: AS SOURCE GETS O. D. M. B.  FOR NOW
     1975 E0 53                      CPX     #'B
     1977 D0 09                      BNE     OPBRC1
     1979 C9 43                      CMP     #'C
     197B D0 30                      BNE     ODMB
     197D A9 80                      LDA     #$80
     197F 8D 58 03                   STA     ICAX2+$10
     1982 A2 10          OPBRC1      LDX     #$10
     1984 A9 03                      LDA     #OPEN
     1986 9D 42 03                   STA     ICCOM.X
     1989 A9 04                      LDA     #4       OPEN IN
     198B 9D 4A 03                   STA     ICAX1.X
     198E 8E AB 14                   STX     CBRC
     1991 E0 10                      CPX     #$10
     1993 D0 1F                      BNE     *+33
     1995 20 C2 23                   JSR     CIOCL   OPEN SOURCE FILE HERE
     1998 AE AD 14       PDEB        LDX     SAVX
     199B BD 29 13                   LDA     PAR.X
     199E C9 45                      CMP     #'E     CHECK FOR SPECIAL CASE
     19A0 D0 15                      BNE     OPDEB   IF NOT
     19A2 A9 00          PDEB1       LDA     #0      SPECIAL CASE - DONT OPEN, USE EXISTING IOCB
     19A4 8D AC 14                   STA     CDEB
     19A7 CE BA 14                   DEC     RDM     SET FLAG TO RESHOW MENU
     19AA 4C DA 19                   JMP     DOCPY
     19AD A9 1D          ODMB        LDA     #OEL
     19AF A2 19                      LDX     #OEH
     19B1 20 61 23                   JSR     DSPLIN
     19B4 4C 91 16                   JMP     MENUBL
     19B7 E0 41          OPDEB       CPX     #'A     APPEND TO DISK FILE
     19B9 D0 08                      BNE     OPDEB1
     19BB C9 44                      CMP     #'D
     19BD D0 EE                      BNE     ODMB
```

```
19C1 DO 02                   BNE      OPDES3
19C3 A9 08        OPDES1     LDA      #8
19C5 A2 20        OPDES3     LDX      ##20
19C7 9D 4A 03                STA      ICAX1,X  OPEN TYPE OUT
19CA A9 03                   LDA      #OPEN
19CC 9D 42 03                STA      ICCOM,X  OPEN
19CF 8E AC 14                STX      CDES
19D2 20 C2 23                JSR      CIOCL
19D5 A9 00                   LDA      #0
19D7 9D 48 03                STA      ICAX2,X
                  ; COPY FROM CSRC TO CDES
19DA A9 07        DOCPY      LDA      #GETCHR+TEXT
19DC AE AB 14     GC1        LDX      CBRC
19DF AC AC 14                LDY      CDES
19E2 9D 42 03                STA      ICCOM,X
19E5 A9 0B                   LDA      #PUTCHR+TEXT
19E7 99 42 03                STA      ICCOM,Y
19EA A9 A1                   LDA      #DBUFL
19EC 9D 44 03                STA      ICBAL,X
19EF 99 44 03                STA      ICBAL,Y
19F2 A9 13                   LDA      #DBUFH
19F4 9D 45 03                STA      ICBAH,X
19F7 99 45 03                STA      ICBAH,Y
19FA AE AB 14     CLOOP      LDX      CBRC
19FD A9 00                   LDA      ##00
19FF 9D 48 03                STA      ICBLL,X
1A02 A9 01                   LDA      ##01
1A04 9D 49 03                STA      ICBLH,X
1A07 20 56 E4                JSR      CIO      READ FROM INPUT
1A0A BC A5 14                STY      SSTAT
1A0D AE AC 14                LDX      CDES
1A10 AC AB 14                LDY      CBRC
1A13 B9 48 03                LDA      ICBLL,Y
1A16 9D 48 03                STA      ICBLL,X
1A19 B9 49 03                LDA      ICBLH,Y
1A1C 9D 49 03                STA      ICBLH,X
1A1F 19 48 03                ORA      ICBLL,Y  IF SOURCE FILE LENGTH = 0
1A22 FO 03                   BEQ      CKRS     DONT DO WRITE
1A24 20 C2 23                JSR      CIOCL    WRITE, ABORT IF ERROR
1A27 AD A5 14     CKRS       LDA      SSTAT    GET READ OPERATION STATUS BACK
1A2A 10 CE                   BPL      CLOOP    IF OK, GO READ SOME MORE
1A2C C9 88                   CMP      ##88     EOF STATUS
1A2E FO 03                   BEQ      *+5
1A30 4C CA 23                JMP      CIOER    IF NOT, ABORT
1A33 AE AB 14     CLOC       LDX      CBRC
1A36 FO 0C                   BEQ      DU4      IF E:, DONT CLOSE
1A38 E0 30                   CPX      ##30     OR K:
1A3A FO 08                   BEQ      DU4
                  ; CLOSE SOURCE FILE
1A3C A9 0C                   LDA      #CLOSE
1A3E 9D 42 03                STA      ICCOM,X
1A41 20 56 E4                JSR      CIO
1A44 AE AC 14     DU4        LDX      CDES
1A47 FO 08                   BEQ      DU3      IF DES=E:
1A49 A9 0C                   LDA      #CLOSE
1A4B 9D 42 03                STA      ICCOM,X
1A4E 20 56 E4                JSR      CIO
1A51 4C 91 16     DU3        JMP      MENUSL
                  ; ****************
                  ;
```

```
                              ;    RENAME FILE ROUTINE
                              ;
                              ;**************
     1A54 6C 1A        RENFIL  .WORD   RNMO
     1A56 20 84 22             JSR     GETIC1
     1A59 20 90 22             JSR     GETNAME
     1A5C 20 79 22             JSR     PERX
     1A5F A9 20               LDA     #RENAME
     1A61 A2 10               LDX     #$10
     1A63 9D 42 03            STA     ICCOM,X
     1A66 20 C2 23            JSR     CIOCL
     1A69 4C 91 16            JMP     MENUSL
     1A6C 52 45 4E     RNMO    .BYTE   "RENAME, GIVE OLD NAME, NEW",CR
     1A6F 41 4D 45
     1A72 2C 20 47
     1A75 49 56 45
     1A78 20 4F 4C
     1A7B 44 20 4E
     1A7E 41 4D 45
     1A81 2C 20 4E
     1A84 45 57 98
                              ;**************
                              ;
                              ;    FORMAT DISK ROUTINE
                              ;
                              ;**************
     1A87 C0 1A        FMTDSK  .WORD   WHD
     1A89 20 F1 21             JSR     GETLIN
     1A8C 20 53 25             JSR     GETDN
     1A8F 18                  CLC
     1A90 69 30               ADC     #'0
     1A92 8D EF 1A            STA     DDSK
     1A95 8D F2 1A            STA     CDSK
     1A98 20 79 22            JSR     PERX
     1A9B A9 D7               LDA     #VFML   QUERY TO VERIFY DRIVE NUMBER
     1A9D A2 1A               LDX     #VFMH
     1A9F 20 61 23            JSR     DSPLIN
     1AA2 20 33 22            JSR     CHRGET
     1AA5 C9 59               CMP     #'Y     SEE IF OK
     1AA7 D0 14               BNE     FMX
     1AA9 A9 F1               LDA     #FDPL
     1AAB A2 10               LDX     #$10
     1AAD 9D 44 03            STA     ICBAL,X
     1AB0 A9 1A               LDA     #FDPH
     1AB2 9D 45 03            STA     ICBAH,X
     1AB5 A9 FE               LDA     #FORMAT
     1AB7 9D 42 03            STA     ICCOM,X
     1ABA 20 C2 23            JSR     CIOCL   CALL CIO TO DO FORMAT
     1ABD 4C 91 16    FMX     JMP     MENUBL  EXIT.
     1AC0 57 48 49    WHD     .BYTE   "WHICH DRIVE TO FORMAT?",CR
     1AC3 43 48 20
     1AC6 44 52 49
     1AC9 56 45 20
     1ACC 54 4F 20
     1ACF 46 4F 52
     1AD2 4D 41 54
```

```
1AD7 54 59 50    VFM       .BYTE    "TYPE ",$22,"V",$22," TO FORMAT DISK "
1ADA 45 20 22
1ADD 59 22 20
1AEO 54 4F 20
1AE3 46 4F 52
1AE6 4D 41 54
1AE9 20 44 49
1AEC 53 4B 20
1AEF             DDSK      *=*+1
1AFO 9B          FDM'      .BYTE    CR
1AF1 44          FDM'      .BYTE    "D"
1AF2             CDSK      *=*+1
1AF3 3A 9B                 .BYTE    ":",CR
                 HILO      WHD
001A             WHDH      =        WHD/256
00C0             WHDL      =        -256*WHDH+WHD
                 HILO      VFM
001A             VFMH      =        VFM/256
00D7             VFML      =        -256*VFMH+VFM
                 HILO      FDP
001A             FDPH      =        FDP/256
00F1             FDPL      =        -256*FDPH+FDP
                 ;**********
                 ;
                 ;         START CART ROUTINE
                 ;
                 ;********************
E45F             SYVBL     =        SYSVBV
                 HILO      SYVBL
00E4             SYVBLH    =        SYVBL/256
005F             SYVBLL    =        -256*SYVBLH+SYVBL
E462             XTVBL     =        XITVBV
                 HILO      XTVBL
00E4             XTVBLH    =        XTVBL/256
0062             XTVBLL    =        -256*XTVBLH+XTVBL
1AF5 2B 1B       STCAR     .WORD    SCMO
1AF7 AD FC BF              LDA      #BFFC    SEE IF CART
1AFA FO OA                 BEQ      ISCAR    00 IF THERE IS
1AFC A9 1E                 LDA      #NCAL
1AFE A2 1B                 LDX      #NCAH    SAY NO CART
1B00 20 61 23              JSR      DSPLIN
1B03 4C 91 16              JMP      MENUSL
1B06 20 BC 21    ISCAR     JSR      INITIO
                 ;         RESET VERTICAL BLANK VECTORS BEFORE ENTERING CART
1B09 A9 06                 LDA      #6       SET VVBLKI
1B0B A2 E4                 LDX      #SYVBLH HI BYTE
1B0D AO 5F                 LDY      #SYVBLL
1B0F 20 5C E4              JSR      SETVBV
1B12 A9 07                 LDA      #7       SET VVBLKD
1B14 A2 E4                 LDX      #XTVBLH
1B16 AO 62                 LDY      #XTVBLL
1B18 20 5C E4              JSR      SETVBV
1B1B 6C FA BF              JMP      (CARTST)
1B1E 4E 4F 20    NCA       .BYTE    "NO CARTRIDGE",CR
1B21 43 41 52
1B24 54 52 49
1B27 44 47 45
1B2A 9B
                 HILO      NCA
```

```
        001E                NCAL    =       -256*NCAH+NCA
        1B2B 9B             BCMO    .BYTE   CR
                                    ;
                                    ;
        1B2C 3E 1B          BRUN    .WORD   BRMO
        1B2E 20 F1 21               JSR     GETLIN
        1B31 20 BA 24               JSR     GETNO
        1B34 20 79 22               JSR     PERX
        1B37 85 1A                  STA     RAMLO
        1B39 86 1B                  STX     RAMLO+1
        1B3B 6C 1A 00               JMP     (RAMLO)
        1B3E 52 55 4E       BRMO    .BYTE   "RUN FROM WHAT ADDRESS?",CR
        1B41 20 46 52
        1B44 4F 4D 20
        1B47 57 48 41
        1B4A 54 20 41
        1B4D 44 44 52
        1B50 45 53 53
        1B53 3F 9B
                                    ;
                                    ; DEFINE DEVICE
                                    ;
        1B55 4C 4F 47       DEDE    .BYTE   "LOGICAL DEVICE, PHYSICAL DEVICE?",CR
        1B58 49 43 41
        1B5B 4C 20 44
        1B5E 45 56 49
        1B61 43 45 2C
        1B64 20 50 48
        1B67 59 53 49
        1B6A 43 41 4C
        1B6D 20 44 45
        1B70 56 49 43
        1B73 45 3F 9B
        1B76 55 1B          DEFDEV  .WORD   DEDE
        1B78 20 F1 21               JSR     GETLIN
        1B7B 20 9E 22               JSR     GETFIL  GET LOGICAL DEVICE NAME
        1B7E AD AE 14               LDA     PTR
        1B81 C9 03                  CMP     #3      SHOULD BE DEVICE NAME ONLY
        1B83 D0 4C                  BNE     BS1     BAD SYNTAX
        1B85 20 9E 22               JSR     GETFIL  GET PHYSICAL DEVICE
        1B88 20 79 22               JSR     PERX
        1B8B AD 29 13               LDA     PAR     GET LOG DEV
        1B8E 20 21 1C               JSR     HATS    SEARCH HAT (HANDLER ADDRESS TABLE)
        1B91 10 0D                  BPL     DD1     IF FOUND
                                    ;DIDNT FIND IT SO FIND EMPTY SLOT
        1B93 A9 00                  LDA     #0
        1B95 20 21 1C               JSR     HATS
        1B98 30 22                  BMI     STF     TABLE IS FILLED
        1B9A AD 29 13               LDA     PAR
        1B9D 9D 1A 03               STA     HATABS,X  PUT NAME IN TABLE
        1BA0 A9 9B          DD1     LDA     #DRHL
        1BA2 9D 1B 03               STA     HATABS+1,X
        1BA5 A9 25                  LDA     #DRHH
        1BA7 9D 1C 03               STA     HATABS+2,X
        1BAA BE AD 14               STX     SAVX
                                    ;SET UP LDNT,PDST ENTRIES
        1BAD AD 29 13               LDA     PAR     GET DEVICE
        1BB0 20 2E 1C               JSR     FLDS
        1BB3 10 40                  BPL     SU1     IF FOUND IT
```

```
1BB5 A9 00          LDA     #0
1BB7 20 2E 1C       JSR     FLDS
1BBA 10 33          BPL     SU2
                    ;NO ROOM IN TABLE
1BBC A9 C6    STF   LDA     #TFL
1BBE A2 1B          LDX     #TFH
1BC0 20 61 23 DSX   JSR     DSPLIN
1BC3 4C 91 16       JMP     MENUSL
1BC6 54 41 42 TF    .BYTE   "TABLE FULL",CR
1BC9 4C 45 20
1BCC 46 55 4C
1BCF 4C 9B
                    HILO    TF
001B          TFH   =       TF/256
00C6          TFL   =       -256*TFH+TF
1BD1          BS2
1BD1          BS1
1BD1 A9 D7          LDA     #BSML
1BD3 A2 1B          LDX     #BSMH
1BD5 D0 E9          BNE     DSX
1BD7 43 41 4E BSM   .BYTE   "CANT DEFINE THIS DEVICE",CR
1DDA 54 20 44
1BDD 45 46 49
1BE0 4E 45 20
1BE3 54 48 49
1BE6 53 20 44
1BE9 45 56 49
1BEC 43 45 9B
                    HILO    BSM
001B          BSMH  =       BSM/256
00D7          BSML  =       -256*BSMH+BSM
1BEF AD 29 13 SU2   LDA     PAR
1DF2 9D 9D 25       STA     LDNT,X
                    ;SEE NOW IF THIS IS A DELETE
1DF5 AD 2B 13 SU1   LDA     PAR+2
1BF8 C9 9B          CMP     #CR
1BFA D0 0E          BNE     CRD     NO, GO CREATE DEVICE
                    ;DELETE OPERATION, ZERO LDNT AND HAT ENTRIES
1BFC A9 00          LDA     #0
1BFE 9D 9D 25       STA     LDNT,X
1C01 AE AD 14       LDX     SAVX
1C04 9D 1A 03       STA     HATABS,X
1C07 4C 91 16       JMP     MENUSL
                    ;CREATE DEVICE - HAT AND LDNT ARE SET UP, MAKE PDBT ENTRY
1C0A 8A       CRD   TXA
1C0B 0A             ASL     A
1C0C 0A             ASL     A
1C0D 0A             ASL     A
1C0E 0A             ASL     A
1C0F AA             TAX
1C10 A0 00          LDY     #0
1C12 B9 2C 13 CPBL  LDA     PAR+3,Y GET BYTE OF PHYSICAL DEV:FILSPEC
1C15 9D A8 25       STA     PDBT,X  STORE IN TABLE
1C18 E8             INX
1C19 C8             INY
1C1A C0 10          CPY     #16
1C1C D0 E4          BNE     CPBL
1C1E 4C 91 16       JMP     MENUSL
                    ; HATS FINDS DEVICE NAMED IN A, RETURNS ITS INDEX IN X
```

```
        1C21 A2 21          HATB    LDX     #MAXDEV
        1C23 DD 1A 03        HATBL   CMP     HATABB,X
        1C26 F0 05                   BEQ     HATX
        1C28 CA                      DEX
        1C29 CA                      DEX
        1C2A CA                      DEX
        1C2B 10 F6                   BPL     HATBL
        1C2D 60             HATX     RTB
                            ;FLDS IS THE SAME THING FOR LDNT
        1C2E A2 04          FLDS    LDX     #4
        1C30 DD 9D 25        FLL     CMP     LDNT,X
        1C33 F0 03                   BEQ     FLDX
        1C35 CA                      DEX
        1C36 10 F8                   BPL     FLL
        1C38 60             FLDX     RTB
                            ;
                            ; WBOOT - WRITE OUT NEW DOS.SYS FILE
                            ;
        1C39 88 1C          WBOOT   .WORD   GWMO
        1C3B 20 33 22                JSR     CHRGET
        1C3E C9 59                   CMP     #'Y
        1C40 D0 2A                   BNE     WBX     EXIT UNLESS Y
        1C42 A9 6F                   LDA     #WBMOL
        1C44 A2 1C                   LDX     #WBMOH
        1C46 20 61 23                JSR     DSPLIN
        1C49 A9 03                   LDA     #OPEN
        1C4B A2 10                   LDX     #$10
        1C4D 9D 42 03                STA     ICCOM,X
        1C50 A9 A7                   LDA     #DSL
        1C52 9D 44 03                STA     ICBAL,X
        1C55 A9 1C                   LDA     #DSH
        1C57 9D 45 03                STA     ICBAH,X
        1C5A A9 08                   LDA     #8
        1C5C 9D 4A 03                STA     ICAX1,X
        1C5F 20 C2 23                JSR     CIOCL
        1C62 A2 10                   LDX     #$10
        1C64 A9 0C                   LDA     #CLOSE
        1C66 9D 42 03                STA     ICCOM,X
        1C69 20 C2 23                JSR     CIOCL
        1C6C 4C 91 16       WBX     JMP     MENUSL
        1C6F 57 52 49       WRMO    .BYTE   "WRITING NEW DOS.SYS FILE",CR
        1C72 54 49 4E
        1C75 47 20 4E
        1C78 45 57 20
        1C7B 44 4F 53
        1C7E 2E 53 59
        1C81 53 20 46
        1C84 49 4C 45
        1C87 9B
                            HILO    WBMO
        001C                WBMOH   =       WBMO/256
        006F                WBMOL   =       -256*WBMOH+WBMO
        1C88 54 59 50       GWMO    .BYTE   "TYPE ",$22,"Y",$22," TO WRITE NEW DOS FILE",CR
        1C8B 45 20 22
        1C8E 59 22 20
        1C91 54 4F 20
        1C94 57 52 49
        1C97 54 45 20
        1C9A 4F 43 57
```

```
        1CA0 53 20 46
        1CA3 49 4C 45
        1CA6 9B
        1CA7 44 3A 44     DS      .BYTE   "D:DOS.SYS",CR
        1CAA 4F 53 2E
        1CAD 53 59 53
        1CB0 9B
                          HILO    DS
        001C              DSH     =       DS/256
        00A7              DSL     =       -256*DSH+DS

                          ;
                          ; LDFIL - LOAD FILE FUNCTION
                          ;
        1CB1 F3 1C        LDFIL   .WORD   LFMO
        1CB3 A9 91                LDA     #MNSLL
        1CB5 8D E0 02            STA     RUNAD   SET RUN ADDRESS TO MENU SELECT
        1CB8 A9 16                LDA     #MNSLH  SO IF NONE IN FILE WE GO THERE
        1CBA 8D E1 02            STA     RUNAD+1
        1CBD 20 84 22            JSR     GETIC1
        1CC0 20 79 22            JSR     PERX
        1CC3 20 19 13            JSR     LOADV
        1CC6 E0 00                CPX     #0      PROCESS LOAD SUBR RESPONSE
        1CC8 D0 03                BNE     *+5     BR IF ERRORS
        1CCA 6C E0 02            JMP     (RUNAD) IF OK, GO TO RUN ADDRESS
        1CCD E0 03                CPX     #3
        1CCF F0 07                BEQ     NLF     IF BAD LOAD FILE
        1CD1 98                  TYA             OTHERWISE WE GOT A CIO ERROR
        1CD2 4C CA 23            JMP     CIOER   GO SAY WHAT IT IS
        1CD5 4C 91 16   LDFX     JMP     MENUSL
        1CD8 A9 E5      NLF      LDA     #BLFL
        1CDA A2 1C               LDX     #BLFH
        1CDC 20 61 23            JSR     DSPLIN  BAD LOAD FILE MSG
        1CDF 20 F8 23            JSR     CLOSX   CLOSE THE FILE
        1CE2 4C 91 16            JMP     MENUSL  EXIT
        1CE5 42 41 44   BLF      .BYTE   "BAD LOAD FILE",CR
        1CE8 20 4C 4F
        1CEB 41 44 20
        1CEE 46 49 4C
        1CF1 45 9B
                         HILO    BLF
        001C             BLFH    =       BLF/256
        00E5             BLFL    =       -256*BLFH+BLF
        1CF3 4C 4F 41   LFMO     .BYTE   "LOAD FROM WHAT FILE?",CR
        1CF6 44 20 46
        1CF9 52 4F 4D
        1CFC 20 57 48
        1CFF 41 54 20
        1D02 46 49 4C
        1D05 45 3F 9B
                          ;
                          ; LKFILE - DO CIO LOCK
                          ;
        1D08 1D 1D      LKFIL    .WORD   LKMO
        1D0A 20 84 22            JSR     GETIC1
        1D0D 20 79 22            JSR     PERX
        1D10 A9 23               LDA     #LOCK
        1D12 A2 10               LDX     #$10
```

```
1D17 20 C2 23              JSR     CIOCL
1D1A 4C 91 16              JMP     MENUBL
1D1D 57 48 41    LKMO    .BYTE   "WHAT FILE TO LOCK?",CR
1D20 54 20 46
1D23 49 4C 45
1D26 20 54 4F
1D29 20 4C 4F
1D2C 43 4B 3F
1D2F 9B

                 ; ULFIL - DO CIO UNLOCK
1D30 45 1D    ULFIL   .WORD   ULMO
1D32 20 84 22              JSR     GETIC1
1D35 20 79 22              JSR     PERX
1D38 A9 24                 LDA     #UNLOCK
1D3A A2 10                 LDX     #$10
1D3C 9D 42 03              STA     ICCOM,X
1D3F 20 C2 23              JSR     CIOCL
1D42 4C 91 16              JMP     MENUBL
1D45 57 48 41    ULMO    .BYTE   "WHAT FILE TO UNLOCK?",CR
1D48 54 20 46
1D4B 49 4C 45
1D4E 20 54 4F
1D51 20 55 4E
1D54 4C 4F 43
1D57 4B 3F 9B
```

```
1D5A 44 55 50        DDMO    .BYTE    "DUP DISK-SOURCE,DEST DRIVES?",CR
1D5D 20 44 49
1D60 53 48 2D
1D63 53 4F 55
1D66 52 43 45
1D69 2C 44 45
1D6C 53 54 20
1D6F 44 52 49
1D72 56 45 53
1D75 3F 9B
1D77 54 59 50        OK      .BYTE    "TYPE ",$22,"Y",$22," IF OK TO USE PROGRAM AREA",CR
1D7A 45 20 22
1D7D 59 22 20
1D80 49 46 20
1D83 4F 4B 20
1D86 54 4F 20
1D89 55 53 45
1D8C 20 50 52
1D8F 4F 47 52
1D92 41 4D 20
1D95 41 52 45
1D98 41 9B
                     HILO    OK
001D                 OKH     =        OK/256
0077                 OKL     =        -256*OKH+OK
                     ;****************

                     ;
                     ;
                     ;        DUPLICATE DISK ROUTINE
                     ;
                     ;*************
                     ;
                     ;        RVTOC READS VOLUME TABLE OF CONTENTS SECTOR
                     ;
1D9A A9 01           RVTOC   LDA      #1
1D9C 8D 0B 03                STA      DSHI     READ VTOC SECTOR
1D9F A9 68                   LDA      #$68
1DA1 8D 0A 03                STA      DSLO
1DA4 A9 14                   LDA      #DB1H
1DA6 8D 05 03                STA      DBUFHI
1DA9 A9 21                   LDA      #DB1L
1DAB 8D 04 03                STA      DBUFLO   POINT DCB AT PAR
1DAE 20 C5 1F                JSR      RSEC1
1DB1 A9 00                   LDA      #0
1DB3 8D AE 14                STA      PTR
1DB6 AD 33 13                LDA      PAR+$A
1DB9 8D AB 14                STA      CBRC     BYTE OF ALLOC MAP
1DBC A9 08                   LDA      #8
1DBE 8D AF 14                STA      IPTR     COUNT BITS IN BYTE
1DC1 A9 00                   LDA      #0
1DC3 8D 0B 03                STA      DSHI     POINT T SECTOR ONE
1DC6 A9 01                   LDA      #1
1DC8 8D 0A 03                STA      DSLO
1DCB 60                      RTS
```

```
1DCE 20 F1 21          JSR     GETLIN
1DD1 20 53 25          JSR     GETDN
1DD4 8D A3 14          STA     UNNO    UNIT NO FOR READ
1DD7 20 53 25          JSR     GETDN
1DDA 8D AC 14          STA     CDES
1DDD 20 79 22          JSR     PERX
1DE0 AD A3 14          LDA     UNNO
1DE3 CD AC 14          CMP     CDES    IF BOTH UNITS THE SAME
1DE6 F0 45             BEQ     SDD     SINGLE DRIVE DUP
1DE8 A2 1E             LDX     #IBDH
1DEA A9 0E             LDA     #IBDL
1DEC 20 61 23          JSR     DSPLIN  PROMPT TO INSERT BOTH DISKS
1DEF 20 33 22          JSR     CHRGET
1DF2 20 9A 1D          JSR     RVTOC
1DF5 20 B7 1F   NSEC   JSR     AAM     ADV ALLOC MAP ONE BIT
1DF8 30 0C             BMI     CEND
1DFA 20 BB 1F          JSR     RSEC
1DFD AD AC 14          LDA     CDES    UNIT TO WRITE TO
1E00 8D 01 03          STA     DUNIT
1E03 20 D0 1F          JSR     DKWRT
1E06 20 A4 1F   CEND   JSR     ASP
1E09 D0 EA             BNE     NSEC    IF MORE SECTORS
1E0B 4C 91 16          JMP     MENUSL  DONE
1E0E 49 4E 53   IBD    .BYTE   "INSERT BOTH DISKS, TYPE RETURN",CR
1E11 45 52 54
1E14 20 42 4F
1E17 54 48 20
1E1A 44 49 53
1E1D 4B 53 2C
1E20 20 54 59
1E23 50 45 20
1E26 52 45 54
1E29 55 52 4E
1E2C 9B
                       HILO    IBD
001E            IBDH    =       IBD/256
000E            IBDL    =       -256*IBDH+IBD
                       ;SINGLE DRIVE DISK DUP
                       ;
                       ;FIND AVAIL MEMORY. FILL FROM SOURCE DISK, SWAP, EMPTY, SWAP, REPEAT.
                       ;
1E2D AD E7 02   SDD    LDA     MEMLO   END OF DUP
1E30 8D B4 14          STA     STVEC
1E33 AD E8 02          LDA     MEMLO+1
1E36 8D B5 14          STA     STVEC+1
1E39 AD E5 02          LDA     MEMTOP
1E3C 38               SEC
1E3D E9 80             SBC     #$80    END-128 TO SEE IF ROOM
1E3F 8D B2 14          STA     T1      FOR ANOTHER SECTOR
1E42 AD E6 02          LDA     MEMTOP+1
1E45 E9 00             SBC     #0
1E47 8D B3 14          STA     T1+1
                       ;SEE IF ROOM FOR AT LEAST ONE SECTOR!
1E4A AD B3 14          LDA     T1+1
1E4D CD B5 14          CMP     STVEC+1
1E50 30 0A             BMI     NORM
1E52 D0 12             BNE     ENUF
1E54 AD B2 14          LDA     T1
1E57 CD B4 14          CMP     STVEC
```

```
1E5C A9 34          NORM    LDA     #NRML
1E5E A2 1F                  LDX     #NRMH
1E60 20 61 23              JSR     DSPLIN
1E63 4C 91 16              JMP     MENUSL
1E66 20 F2 1F      ENUF    JSR     CKMEM   SEE IF OK TO USE USER AREA
1E69 A9 00                 LDA     #0
1E6B 8D B1 14             STA     OPT     SET UP FOR READ HERE FIRST PASS
1E6E A9 44                LDA     #ISDL
1E70 A2 1F                LDX     #ISDH
1E72 20 61 23            JSR     DSPLIN
1E75 20 33 22            JSR     CHROET
1E78 20 9A 1D            JSR     RVTOC   READ VTOC AFTER 1ST INSERT SOURCE MSG
1E7B AD 0A 03            LDA     DSLO    COPY INITIAL WRITE POINTERS
1E7E 8D A6 14           STA     SWDP    TO INITIAL READ POINTERS
1E81 AD 0B 03           LDA     DSHI
1E84 8D A7 14          STA     SWDP+1
1E87 AD AE 14          LDA     PTR
1E8A 8D A8 14         STA     SWDP+2
1E8D AD AF 14         LDA     IPTR
1E90 8D A9 14        STA     SWDP+3
1E93 AD A8 14        LDA     CSRC
1E96 8D AA 14       STA     SWDP+4
1E99 4C AB 1E              JMP     LRS1    SKIP FIRST READ PROMPT
                          ;READ FROM SOURCE DISK TIL BUF FULL OR END OF DATA.
1E9C A9 00         DORD    LDA     #0      FLAG WE ARE READING
1E9E 8D B1 14             STA     OPT
1EA1 A9 44                LDA     #ISDL   INSERT SRC DISK
1EA3 A2 1F                LDX     #ISDH
1EA5 20 61 23     XBLK    JSR     DSPLIN
1EA8 20 33 22            JSR     CHROET
                          ;SWAP POINTERS TO WHERE WE ARE
1EAB 20 00 1F     LRS1    JSR     DOSWDP  SWAP SECTOR AND BITMAP POINTERS
                          ;LOOP READING/WRITING SECTORS TO BUFFER AREA
1EAE 20 87 1F     LRS     JSR     AAM     ADVANCE ALLOCATION MAP
1EB1 30 1F                BMI     ASPT    IF FREE, ADV SECTOR POINTER AND TRY AGIN
1EB3 2C B1 14            BIT     OPT     SEE WHAT MODE
1EB6 30 06                BMI     DOW     BR IF WRITE
1EB8 20 C5 1F            JSR     RSEC1   DO READ
1EBB 4C C1 1E            JMP     IOD
1EBE 20 D0 1F     DOW     JSR     DKWRT   DO WRITE
1EC1 AD 04 03     IOD     LDA     DBUFLO  ADVANCE BUFFER POINTER
1EC4 18                   CLC
1EC5 69 80                ADC     #$80
1EC7 8D 04 03            STA     DBUFLO
1ECA AD 05 03           LDA     DBUFHI
1ECD 69 00                ADC     #0
1ECF 8D 05 03            STA     DBUFHI
1ED2 20 A4 1F     ASPT    JSR     ASP     GO ADVANCE SECTOR POINTER
1ED5 F0 21                BEQ     STDD1   ALL SECTORS DONE, SWAP TO DEST DISK
1ED7 AD B3 14           LDA     T1+1    SEE IF ROOM FOR ANOTHER
1EDA CD 05 03           CMP     DBUFHI
1EDD 30 0A                BMI     STDD    IF NO ROOM, GO SWAP
1EDF D0 CD                BNE     LRS     ROOM SO CONTINUE
1EE1 AD B2 14           LDA     T1
1EE4 CD 04 03           CMP     DBUFLO
1EE7 10 C5                BPL     LRS     ROOM.
                          ;SWAP DISKS AND CONTINUE
1EE9 AD B1 14     STDD    LDA     OPT
1EEC 30 AE                BMI     DORD    IF WAS WRITE, GO READ
```

```
1EF1 A9 63                LDA    #IDDL    INSERT DEST DISK
1EF3 A2 1F                LDX    #IDDH
1EF5 4C A5 1E             JMP    XBLK     GO DO WRITE
1EF8 AD B1 14     STDD1   LDA    OPT      END OF DATA
1EFB 10 F1                BPL    STDD2    IF READ GO WRITE
1EFD 4C 91 16             JMP    MENUSL   IF WRITE WE ARE DONE
                        ; DOSWDP - EXCHANGE CURRENT AND SAVED BITMAP&SECTOR POINTERS
                        ; ALSO INIT BUFFER POINTER
1F00 A0 04       DOSWDP   LDY    #4
1F02 B9 2A 1F    SWLOP    LDA    SWATL,Y
1F05 85 1A                STA    RAMLO
1F07 B9 2F 1F             LDA    SWATH,Y
1F0A 85 1B                STA    RAMLO+1  GET ADDRESS FROM TABLE TO RAMLO
1F0C A2 00                LDX    #0
1F0E A1 1A                LDA    (RAMLO,X)        GET WHATS THERE
1F10 48                   PHA
1F11 B9 A6 14             LDA    SWDP,Y
1F14 81 1A                STA    (RAMLO,X)
1F16 68                   PLA
1F17 99 A6 14             STA    SWDP,Y
1F1A 88                   DEY
1F1B 10 E5                BPL    SWLOP
1F1D AD B4 14             LDA    STVEC
1F20 8D 04 03             STA    DBUFLO
1F23 AD B5 14             LDA    STVEC+1
1F26 8D 05 03             STA    DBUFHI
1F29 60                   RTS
                        ;
                        ;     WHAT A MESS
                                 HILO    DBLO
0003             DBLOH    =       DBLO/256
000A             DBLOL    =       -256*DBLOH+DBLO
                                 HILO    DBHI
0003             DBHIH    =       DBHI/256
000B             DBHIL    =       -256*DBHIH+DBHI
                                 HILO    PTR
0014             PTRH     =       PTR/256
00AE             PTRL     =       -256*PTRH+PTR
                                 HILO    IPTR
0014             IPTRH    =       IPTR/256
00AF             IPTRL    =       -256*IPTRH+IPTR
                                 HILO    CSRC
0014             CSRCH    =       CSRC/256
00AB             CSRCL    =       -256*CSRCH+CSRC
                        ;
1F2A 0A 0B AE    SWATL    .BYTE   DBLOL,DBHIL,PTRL,IPTRL,CSRCL,
1F2D AE AB
1F2F 03 03 14    SWATH    .BYTE   DBLOH,DBHIH,PTRH,IPTRH,CSRCH
1F32 14 14
                        ;
1F34 4E 4F 54    NRM      .BYTE   "NOT ENOUGH ROOM",CR
1F37 20 45 4E
1F3A 4E 55 47
1F3D 48 20 52
1F40 4F 4F 4D
1F43 9B
1F44 49 4E 53    ISD      .BYTE   "INSERT SOURCE DISK,TYPE RETURN",CR
1F47 45 52 54
```

```
1F4D 55 52 43
1F50 45 20 44
1F53 49 53 4B
1F56 2C 54 59
1F59 50 45 20
1F5C 52 45 54
1F5F 55 52 4E
1F62 9B
1F63 49 4E 53        IDD     .BYTE   "INSERT DESTINATION DISK, TYPE RETURN",CR
1F66 45 52 54
1F69 20 44 49
1F6C 53 54 49
1F6F 4E 41 54
1F72 49 4F 4E
1F75 20 44 49
1F78 53 4B 2C
1F7B 54 59 50
1F7E 45 20 52
1F81 45 54 55
1F84 52 4E 9B
                     HILO    NRM
001F                 NRMH    =       NRM/256
0034                 NRML    =       -256*NRMH+NRM
                     HILO    ISD
001F                 ISDH    =       ISD/256
0044                 ISDL    =       -256*ISDH+ISD
                     HILO    IDD
001F                 IDDH    =       IDD/256
0063                 IDDL    =       -256*IDDH+IDD
                     ; AAM - ADVANCE ALLOCATION MAP ONE BIT.
                     ; RETURN MINUS IF FREE.
1F87 0E AB 14        AAM     ASL     CBRC    NEXT BIT OF ALLOC MAP
1F8A CE AF 14                DEC     IPTR
1F8D D0 11                   BNE     CBIT    IF DONE WITH THIS BYTE
1F8F EE AE 14                INC     PTR     GET NEXT ONE
1F92 AE AE 14                LDX     PTR
1F95 BD 28 14                LDA     DB1+$A,X
1F98 8D AB 14                STA     CBRC
1F9B A9 08                   LDA     #8
1F9D 8D AF 14                STA     IPTR
1FA0 AD AB 14        CBIT    LDA     CBRC    CHECK THE BIT
1FA3 60                      RTS
                     ; ASP - ADVANCE SECTOR POINTER IN DCB.
                     ; RETURN EQ IF AT END.
1FA4 AD 0A 03        ASP     LDA     DSLO    SEE IF END
1FA7 C9 CF                   CMP     #207
1FA9 D0 07                   BNE     NXS
1FAB AD 0B 03                LDA     DSHI
1FAE C9 02                   CMP     #2
1FB0 F0 08                   BEQ     ASPX    ALL DONE
1FB2 EE 0A 03        NXS     INC     DSLO
1FB5 D0 03                   BNE     ASPX
1FB7 EE 0B 03                INC     DSHI
1FBA 60              ASPX    RTS
                     ;
                     ; RSEC - READ A SECTOR WHOSE NUMBER IS IN DCB
                     ;
1FBB A9 A1           RSEC    LDA     #DBUFL
1FBD 8D 04 03                STA     DBUFLO
```

```
IFC2 8D 05 03              STA     DBUFHI
IFC5 AD A3 14    RSEC1     LDA     UNNO
IFC8 8D 01 03              STA     DUNIT
IFCB A9 52                 LDA     #$52    GET SECTOR
IFCD 4C D2 1F              JMP     CLDKH

                   ;   DKWRT - WRITE A SECTOR, NUMBER IN DCB
                   ;
IFD0 A9 57       DKWRT     LDA     #$57
IFD2 8D 02 03    CLDKH     STA     DCOMND
IFD5 A9 02                 LDA     #2      SET RETRY COUNT
IFD7 8D A4 14              STA     RCNT
IFDA 20 53 E4    CLD1      JSR     DKHND
IFDD AD 03 03              LDA     DSTATS
IFE0 C9 01                 CMP     #1
IFE2 F0 08                 BEQ     DRTS    OK STATUS
IFE4 CE A4 14              DEC     RCNT
IFE7 10 F1                 DPL     CLD1
IFE9 4C CA 23              JMP     CIOER   CIO ERROR, GO SAY WHICH
IFEC 60          DRTS      RTS

                   ;   CKMEM - ASK IF OK TO USE USER AREA
                   ;
IFED 68          DDXT      PLA             POP RETURN ADDRESS
IFEE 68                    PLA
IFEF 4C 91 16              JMP     MENUSL
IFF2 A9 08       CKMEM     LDA     WARMST  IF MEMORY WAS INTACT
IFF4 F0 12                 BEQ     CPTR1   QUERY TO BOMB IT
IFF6 A9 77                 LDA     #OKL
IFF8 A2 1D                 LDX     #OKH
IFFA 20 61 23              JSR     DSPLIN
IFFD 20 33 22              JSR     CHRGET
2000 C9 59                 CMP     #'Y
2002 D0 E9                 BNE     DDXT
2004 A9 00                 LDA     #0
2006 85 08                 STA     WARMST
2008 60          CPTR1     RTS
```

```
                                !        DUPFIL - COPY FILE FROM ONE DISK TO ANOTHER
     2009 4E 41 4D      DPFM     .BYTE   "NAME OF FILE TO MOVE?",CR
     200C 45 20 4F
     200F 46 20 46
     2012 49 4C 45
     2015 20 54 4F
     2018 20 4D 4F
     201B 56 45 3F
     201E 9B
     201F 09 20      DUPFIL    .WORD   DPFM
     2021 20 84 22              JSR     GETIC1  GET A DEV:FILSPEC
     2024 20 79 22              JSR     PERX    CHECK FOR ERRORS
     2027 20 F2 1F              JSR     CKMEM
     202A A2 10                 LDX     #$10
     202C A9 03                 LDA     #OPEN
     202E 9D 42 03              STA     ICCOM,X
     2031 A9 04                 LDA     #4
     2033 9D 4A 03              STA     ICAX1,X
     2036 20 C2 23              JSR     CIOCL
     2039 AD E5 02              LDA     MEMTOP  SET BUFFER LENGTH
     203C 38                    SEC             TO ALL AVAILABLE MEMORY
     203D ED E7 02              SBC     MEMLO
     2040 9D 48 03              STA     ICBLL,X
     2043 AD E6 02              LDA     MEMTOP+1
     2046 ED E8 02              SBC     MEMLO+1
     2049 9D 49 03              STA     ICBLH,X
     204C AD E7 02              LDA     MEMLO
     204F 9D 44 03              STA     ICBAL,X
     2052 AD E8 02              LDA     MEMLO+1
     2055 9D 45 03              STA     ICBAH,X
     2058 A9 06                 LDA     #GETCHR
     205A 9D 42 03              STA     ICCOM,X
     205D 20 56 E4              JSR     CIO
     2060 C0 88                 CPY     #$88
     2062 F0 1B                 BEQ     OKS     IF EOF, WE READ WHOLE FILE SO OK
     2064 C0 01                 CPY     #1
     2066 F0 03                 BEQ     *+5
     2068 4C C9 23              JMP     CIOER1  NOT OK OR EOF SO GIVE ERROR MESSAGE
     206B A9 34                 LDA     #NRML   OK, SAY NO ROOM
     206D A2 1F                 LDX     #NRMH
     206F 20 61 23              JSR     DSPLIN
     2072 A9 0C      DFCX       LDA     #CLOSE
     2074 A2 10                 LDX     #$10
     2076 9D 42 03              STA     ICCOM,X
     2079 20 56 E4              JSR     CIO
     207C 4C 91 16              JMP     MENUSL
     207F A9 0C      OKS        LDA     #CLOSE
     2081 A2 10                 LDX     #$10
     2083 9D 42 03              STA     ICCOM,X
     2086 BD 48 03              LDA     ICBLL,X
     2089 8D B2 14              STA     T1
     208C BD 49 03              LDA     ICBLH,X
     208F 8D B3 14              STA     T1+1
     2092 20 56 E4              JSR     CIO
     2095 A2 1F                 LDX     #IDDH
     2097 A9 63                 LDA     #IDDL
     2099 20 61 23              JSR     DSPLIN  SAY TO SWAP DISKS
```

```
209F A2 10              LDX     #$10
20A1 A9 29              LDA     #PARL
20A3 9D 44 03           STA     ICBAL,X  POINT TO FILENAMEAGAIN
20A6 A9 13              LDA     #PARH
20A8 9D 45 03           STA     ICBAH,X
20AB A9 03              LDA     #OPEN
20AD 9D 42 03           STA     ICCOM,X
20B0 A9 08              LDA     #8
20B2 9D 4A 03           STA     ICAX1,X
20B5 20 C2 23           JSR     CIOCL    OPEN FOR OUTPUT
20B8 AD E7 02           LDA     MEMLO
20BB 9D 44 03           STA     ICBAL,X
20BE AD E8 02           LDA     MEMLO+1  SET UP ADR AND LENGTH
20C1 9D 45 03           STA     ICBAH,X
20C4 AD B2 14           LDA     T1
20C7 9D 48 03           STA     ICBLL,X
20CA AD B3 14           LDA     T1+1
20CD 9D 49 03           STA     ICBLH,X
20D0 A9 0A              LDA     #PUTCHR
20D2 9D 42 03           STA     ICCOM,X
20D5 20 C2 23           JSR     CIOCL    DO THE WRITE
20D8 4C 72 20           JMP     DFCX     CLOSE AND EXIT

                        ;**************
                        ;
                        ;   SAVE FILE ROUTINE
                        ;
                        ;**************
20DB A0 21      SAVFIL  .WORD   SFMO
20DD 20 84 22           JSR     GETIC1
20E0 AD B1 14           LDA     OPT
20E3 48                 PHA
20E4 AE AE 14           LDX     PTR      PUT EOL ON FILENAME
20E7 A9 9B              LDA     #CR
20E9 9D 28 13           STA     PAR-1,X
20EC 20 BA 24           JSR     GETNO    GET HEX PARAMETER
20EF 8D BB 14           STA     LDST
20F2 8E BC 14           STX     LDST+1
20F5 20 BA 24           JSR     GETNO    END ADDRESS
20F8 8D BD 14           STA     LDND
20FB 8E BE 14           STX     LDND+1
20FE 2C A2 14           BIT     PER      SEE IF PARAM ERRORS
2101 10 03              BPL     *+5
2103 4C 91 16           JMP     MENUBL   IF ERRORS
2106 A9 00              LDA     #0
2108 8D B1 14           STA     OPT
210B 68                 PLA              OPTION CHAR FROM FILENAME
210C C9 41              CMP     #'A      IF APPEND
210E D0 03              BNE     *+5
2110 CE B1 14           DEC     OPT      SET OT=$FF
                        ; OPEN THE FILE
2113 A2 10              LDX     #$10
2115 A9 03              LDA     #OPEN
2117 9D 42 03           STA     ICCOM,X
211A 2C B1 14           BIT     OPT      IF APPEND
211D 30 04              BMI     *+6
211F A9 08              LDA     #8
2121 D0 02              BNE     *+4
2123 A9 09              LDA     #9
2125 9D 4A 03           STA     ICAX1,X
```

```
              2128 20 C2 23              JSR     CIOCL
                          ; WRITE SAVE FILE HEADER
              212B A2 10                 LDX     #$10
              212D A9 0A                 LDA     #PUTCHR
              212F 9D 42 03              STA     ICCOM,X
              2132 2C B1 14              BIT     OPT       IF APPEND,
              2135 30 17                 BMI     WDR       DONT WRITE ANOTHER HEADER
              2137 A9 BF                 LDA     #SAVHL
              2139 9D 44 03              STA     ICBAL,X
              213C A9 14                 LDA     #SAVHH
              213E 9D 45 03              STA     ICBAH,X
              2141 A9 02                 LDA     #2
              2143 9D 48 03              STA     ICBLL,X
              2146 A9 00                 LDA     #0
              2148 9D 49 03              STA     ICBLH,X
              214B 20 C2 23              JSR     CIOCL
                          ; WRITE DATA RECORD
              214E A2 10          WDR    LDX     #$10
              2150 A9 04                 LDA     #4
              2152 9D 48 03              STA     ICBLL,X
              2155 A9 00                 LDA     #0
              2157 9D 49 03              STA     ICBLH,X
              215A A9 BB                 LDA     #DTHL     DATA RECORD HEADER
              215C 9D 44 03              STA     ICBAL,X
              215F A9 14                 LDA     #DTHH
              2161 9D 45 03              STA     ICBAH,X
              2164 20 C2 23              JSR     CIOCL     NO. WRITE DATA REC HDR
              2167 A2 10                 LDX     #$10
              2169 38                    SEC
              216A AD BD 14              LDA     LDND
              216D ED BB 14              SBC     LDST
              2170 9D 48 03              STA     ICBLL,X
              2173 AD BE 14              LDA     LDND+1
              2176 ED BC 14              SBC     LDST+1
              2179 9D 49 03              STA     ICBLH,X
              217C FE 48 03              INC     ICBLL,X
              217F D0 03                 BNE     *+5
              2181 FE 49 03              INC     ICBLH,X
              2184 AD BB 14              LDA     LDST
              2187 9D 44 03              STA     ICBAL,X
              218A AD BC 14              LDA     LDST+1
              218D 9D 45 03              STA     ICBAH,X
              2190 20 C2 23              JSR     CIOCL     WRITE THE DATA
              2193 A2 10                 LDX     #$10
              2195 A9 0C                 LDA     #CLOSE
              2197 9D 42 03              STA     ICCOM,X
              219A 20 C2 23              JSR     CIOCL
              219D 4C 91 16              JMP     MENUSL    DONE.
              21A0 53 41 56       SFMO   .BYTE   "SAVE--GIVE FILE, START, END",CR
              21A3 45 2D 2D
              21A6 47 49 56
              21A9 45 20 46
              21AC 49 4C 45
              21AF 2C 20 53
              21B2 54 41 52
              21B5 54 2C 20
              21B8 45 4E 44
```

```
                                     ;          S U B R O U T I N E S
                                     ;
                                     ;
                                     ;
                                     ;  INIT10 - CLOSE ANY OPEN IOCBS
            21BC A2 70               INIT10  LDX     #$70
            21BE BD 40 03            II01    LDA     ICHID,X
            21C1 C9 FF                       CMP     #$FF    CHECK FOR CLOSED IOCB
            21C3 F0 08                       REQ     NXIOCB  ALREADY CLOSED, CHECK NEXT IOCB
            21C5 A9 0C                       LDA     #CLOSE
            21C7 9D 42 03                    STA     ICCOM,X
            21CA 20 56 E4                    JSR     CIO
            21CD 8A               NXIOCB  TXA
            21CE 38                       SEC
            21CF E9 10                    SBC     #$10
            21D1 AA                       TAX
            21D2 10 EA                    BPL     II01
            21D4 A2 00                    LDX     #0      REOPEN IOCB0 TO E:
            21D6 A9 03                    LDA     #OPEN
            21D8 8D 42 03                 STA     ICCOM
            21DB A9 EE                    LDA     #ECL
            21DD 8D 44 03                 STA     ICBAL
            21E0 A9 21                    LDA     #ECH
            21E2 8D 45 03                 STA     ICBAH
            21E5 A9 0C                    LDA     #$C
            21E7 9D 4A 03                 STA     ICAX1,X
            21EA 20 56 E4                 JSR     CIO
            21ED 60                       RTS
            21EE 45 3A 9B        EC      .BYTE   "E: ",CR
                                         HILO    EC
            0021                 ECH     =       EC/256
            00EE                 ECL     =       -256*ECH+EC
                                     ;
            21F1 A9 9B           GETLIN  LDA     #CR
            21F3 A2 4F                   LDX     #79
            21F5 9D 51 13                STA     LINE,X
            21F8 CA                      DEX
            21F9 10 FA                   BPL     *-4
            21FB A9 00                   LDA     #0
            21FD 8D AE 14                STA     PTR
            2200 8D AF 14                STA     IPTR
            2203 8D A2 14                STA     PER
            2206 20 0D 22                JSR     CIOGET
            2209 20 8E 23                JSR     SCROL
            220C 60                      RTS
                                     ;
                                     ;
                                     ;  CIOGET - GET LINE OF INPUT FROM SCREEN EDITOR
                                     ;
            220D A9 05           CIOGET  LDA     #GETRCD+TEXT
            220F 8D 42 03                STA     ICCOM   SCREEN EDIT IOCB
            2212 A9 51                   LDA     #LBUFL
            2214 8D 44 03                STA     ICBAL
            2217 A9 13                   LDA     #LBUFH
```

```
221C A9 50          LDA     #80
221E 8D 48 03       STA     ICBLL
2221 A9 00          LDA     #0
2223 8D 49 03       STA     ICBLH
2226 A2 00          LDX     #0
2228 20 56 E4       JSR     CIO     READ RECORD FROM SCREEN EDITOR
222B C0 80          CPY     #$80    CHECK FOR BREAK ABORT STATUS
222D D0 03          BNE     *+5
222F CE A2 14       DEC     PER     PARAM ERROR FLAG IS SET IF 80
2232 60             RTS
                            ;
                            ;
                            ; CHRGET - GET 1 CHAR FROM EDITOR IN A.
                            ;
2233 A9 00   CHRGET LDA     #0
2235 8D A2 14       STA     PER
2238 20 0D 22 CHR01 JSR     CIOGET  GET A LINE FROM E:
223B AD 48 03       LDA     ICBLL   SAVE CHAR COUNT
223E 8D A4 14       STA     RCNT
2241 20 8E 23       JSR     SCROL
2244 AD A2 14       LDA     PER
2247 10 06          BPL     CHR02   IF BREAK, CLOSE AND EXIT
2249 20 F8 23       JSR     CLOSX
224C 4C 91 16       JMP     MENUSL
224F AD A4 14 CHR02 LDA     RCNT    EXPECT 1 OR 2 CHARACTERS
2252 C9 03          CMP     #3
2254 30 0A          BMI     CHR03   IF OK
2256 A9 64          LDA     #OLL
2258 A2 22          LDX     #OLH
225A 20 61 23       JSR     DSPLIN
225D 4C 38 22       JMP     CHR01   TRY AGIN
2260 AD 51 13 CHR03 LDA     LINE    GET 1ST CHAR              (A)
2263 60             RTS
2264 50 4C 45 OL    .BYTE   "PLEASE TYPE 1 LETTER",CR
2267 41 53 45
226A 20 54 59
226D 50 45 20
2270 31 20 4C
2273 45 54 54
2276 45 52 9B
                            HILO    OL
0022                OLH     =       OL/256
0064                OLL     =       -256*OLH+OL
                            ;
                            ; PERX - EXIT IF PARAMETER ERRORS
                            ;
2279 2C A2 14 PERX  BIT     PER
227C 30 01          BMI     PERX1
227E 60             RTS
227F 68      PERX1  PLA
2280 68             PLA
2281 4C 91 16       JMP     MENUSL
                            ;
                            ; CRTIC1 - READ LINE, GET FILENAME, POINT TO IT IN IOCB1
                            ;
```

```
                              ;
                              ;
 2290 A9 08          GETNAME LDA      #8
 2292 8D B0 14               STA      CTR
 2295 AC AE 14               LDY      PTR
 2298 AE AF 14               LDX      IPTR
 229B 4C F4 22               JMP      CFTE
                              ;
                              ;
                     ;GET FILESPEC FROM INPUT LINE
 229E AC AE 14       GETFIL  LDY      PTR
 22A1 AE AF 14               LDX      IPTR
 22A4 A9 0A                  LDA      #10
 22A6 8D B0 14               STA      CTR
                     ;AVOID GETTING JUNK ON VERY SHORT PARAMS
 22A9 BD 51 13               LDA      LINE,X
 22AC C9 2C                  CMP      #',
 22AE F0 3B                  BEQ      ADDC
 22B0 C9 9B                  CMP      #CR
 22B2 F0 34                  BEQ      ADDC
 22B4 BD 52 13               LDA      LINE+1,X
 22B7 C9 2C                  CMP      #',
 22B9 F0 22                  BEQ      GT1
 22BB C9 9B                  CMP      #CR
 22BD F0 1E                  BEQ      GT1
 22BF A9 3A                  LDA      #':      LOOK FOR : IN FILESPEC
 22C1 DD 53 13               CMP      LINE+2,X     SEE IF HAVE COMPLETE FILESPEC ALREADY
 22C4 F0 2E                  BEQ      CFTE
 22C6 DD 52 13               CMP      LINE+1,X
 22C9 D0 12                  BNE      GT1
 22CB CE B0 14               DEC      CTR
 22CE BD 51 13               LDA      LINE,X
 22D1 C9 41                  CMP      #'A
 22D3 10 1F                  BPL      CFTE     HAVE X:FILE, COMPLETE FILESPEC
                     ;IF FALLS THRU, IS UNIT:FILE; ADD D
 22D5 A9 44          GT2     LDA      #'D
 22D7 99 29 13               STA      PAR,Y
 22DA C8                     INY
 22DB 10 17                  BPL      CFTE
 22DD CE B0 14       GT1     DEC      CTR
 22E0 DD 51 13               CMP      LINE,X   AN UNLIKELY CASE (:FILE)
 22E3 F0 F0                  BEQ      GT2      TREAT :FILE AS U:FILE
 22E5 CE B0 14               DEC      CTR
 22E8 A9 44          ADDC    LDA      #'D
 22EA 99 29 13               STA      PAR,Y
 22ED C8                     INY
 22EE A9 3A                  LDA      #':
 22F0 99 29 13               STA      PAR,Y
 22F3 C8                     INY
 22F4 A9 00          CFTE    LDA      #0
 22F6 8D B1 14               STA      OPT
 22F9 BD 51 13       CFTE1   LDA      LINE,X
 22FC 99 29 13               STA      PAR,Y
 22FF E8                     INX
 2300 C8                     INY
 2301 C9 9B                  CMP      #CR      LOOK FOR TERMINATOR
 2303 F0 2C                  BEQ      EOC
 2305 C9 2C                  CMP      #',
 2307 F0 28                  BEQ      EOC
```

```
230B F0 2B              BEQ     POPT
230D C9 2E              CMP     #'.      LOOK FOR START OF .EXT
230F D0 05              BNE     CFTE2
2311 A9 04              LDA     #4       FOUND, 4 MORE CHARS MAX
2313 8D B0 14           STA     CTR
2316 CE B0 14    CFTE2  DEC     CTR
2319 10 DE              BPL     CFTE1
                 ;GETS HERE IF TOO MANY CHARS IN FILENAME
231B A9 48              LDA     #NTLL
231D A2 23              LDX     #NTLH
231F 20 61 23           JSR     DSPLIN   NAME TOO LONG
2322 CE A2 14           DEC     PER      SET PARAMETER ERROR FLAG
2325 BD 51 13    STE    LDA     LINE,X   SKIP TO END
2328 E8                 INX
2329 C9 2C              CMP     #',
232B F0 04              BEQ     EOC
232D C9 9B              CMP     #CR
232F D0 F4              BNE     STE
2331 8E AF 14    EOC    STX     IPTR
2334 8C AE 14           STY     PTR
2337 60                 RTS
2338 BD 51 13    POPT   LDA     LINE,X
233B 8D B1 14           STA     OPT
233E E8                 INX
233F BD 51 13           LDA     LINE,X
2342 99 28 13           STA     PAR-1,Y  CHANGE STORED TERMINATOR TO , OR CR I HOPE
2345 E8                 INX
2346 10 E9              BPL     EOC
2348 4E 41 4D    NTL    .BYTE   "NAME TOO LONG",CR
234B 45 20 54
234E 4F 4F 20
2351 4C 4F 4E
2354 47 9B
                 HILO    NTL
0023             NTLH    =       NTL/256
0048             NTLL    =       -256*NTLH+NTL
                 ;
                 ; DSPMSG - DISPLAY N BYTES
                 ;       BUFFER POINTER AND LENGTH ARE ALREADY IN IOCB0
                 ;
2356 A9 0A       DSPMSG LDA     #PUTCHR
2358 8D 42 03           STA     ICCOM
235B A2 00              LDX     #0
235D 20 83 23           JSR     CIO1
2360 60                 RTS
                 ;
                 ; DSPLIN - DISPLAY ONE LINE OF TEXT
                 ;       A=LO,X=HI ADDRESS
2361 20 68 23    DSPLIN JSR     DSPL1
2364 20 8E 23           JSR     SCROL
2367 60                 RTS
                 ;
                 ;
2368 8D 44 03    DSPL1  STA     ICBAL    ADDRESS OF LINE TO IOCB
236B 8E 45 03           STX     ICBAH
236E A9 09              LDA     #PUTRCD+TEXT
2370 8D 42 03           STA     ICCOM
2373 A9 00              LDA     #0       SET BUFF LEN
2375 8D 49 03           STA     ICBLH
```

```
237A 8D 48 03           STA     ICBLL
237D A2 00              LDX     #0
237F 20 83 23           JSR     CIO1
2382 60                 RTS
                        ;
2383 20 56 E4   CIO1    JSR     CIO      CALL CIO AND GO TO MENUBL
2386 C0 80              CPY     ##80     IF BREAK KEY ABORT
2388 D0 03              BNE     *+5
238A 4C 91 16           JMP     MENUBL
238D 60                 RTS
                        ;
                        ; SCROL - DO SCROLLING OF AREA BELOW MENU
                        ;
238E A9 00      SCROL   LDA     #0
2390 AA                 TAX
2391 9D 49 03           STA     ICBLH, X
2394 A9 0A              LDA     #10
2396 9D 48 03           STA     ICBLL, X
2399 A9 23              LDA     #ZAPH
239B 9D 45 03           STA     ICBAH, X
239E A9 A7              LDA     #ZAPL
23A0 9D 44 03           STA     ICBAL, X
23A3 20 56 23           JSR     DSPMSG
23A6 60                 RTS
23A7 1C 1C 1C   ZAP     .BYTE   CUP, CUP, CUP, CUP, CUP
23AA 1C 1C
23AC 9C 1D 1D           .BYTE   DLL, CDN, CDN, CDN, CDN
23AF 1D 1D
                        HILO    ZAP
0023            ZAPH    =       ZAP/256
00A7            ZAPL    =       -256*ZAPH+ZAP
                        ;
                        ; PIOCB - POINT IOCB AT PAR(PTR)
                        ;
23B1 A9 29      PIOCB   LDA     #PARL
23B3 18                 CLC
23B4 6D AE 14           ADC     PTR
23B7 9D 44 03           STA     ICBAL, X
23BA A9 13              LDA     #PARH
23BC 69 00              ADC     #0
23BE 9D 45 03           STA     ICBAH, X
23C1 60                 RTS
                        ;
                        ; CIOCL - CALL CIO AND PROCESS ANY ERRORS
                        ;
23C2 20 56 E4   CIOCL   JSR     CIO      CALL CIO
23C5 98                 TYA
23C6 30 01              BMI     *+3
23C8 60                 RTS              OK, RETURN
23C9 98         CIOER1  TYA              ERROR STATUS
23CA 38         CIOER   SEC
23CB E9 64              SBC     #100     ERROR NUMS ALWAYS ARE 1XX DEC
23CD A2 2F              LDX     #'0-1    CONVERT TENS
23CF E8         CTNS    INX
23D0 38                 SEC
23D1 E9 0A              SBC     #10
23D3 10 FA              BPL     CTNS     THE EASY (SLOW) WAY
23D5 18                 CLC
23D6 69 3A              ADC     #10+'0   CONVERT
23D8 8D F4 23           STA     FUN
```

```
23DB 8E F5 23              STX     ETN
23DE A2 23                 LDX     #CIEH
23E0 A9 EB                 LDA     #CIEL
23E2 20 61 23      CIEX    JSR     DSPLIN
23E5 20 F8 23              JSR     CLOSX    CLOSE IOCBS 10,20
23E8 4C 91 16              JMP     MENUSL
23EB 45 52 52      CIE     .BYTE   "ERROR-   1"
23EE 4F 52 2D
23F1 20 20 20
23F4 31
23F5 00            ETN     .BYTE   0
23F6 00            EUN     .BYTE   0
23F7 9B                    .BYTE   CR
                           HILO    CIE
0023               CIEH    =       CIE/256
00EB               CIEL    =       -256*CIEH+CIE
                           ;
                           ; CLOSX - CLOSE IOCBS 10,20
                           ;
23F8 A9 0C         CLOSX   LDA     #CLOSE
23FA 8D 52 03              STA     ICCOM+$10     CLOSE IOCBS 10,20
23FD 8D 62 03              STA     ICCOM+$20
2400 A2 10                 LDX     #$10
2402 20 56 E4              JSR     CIO
2405 A2 20                 LDX     #$20
2407 20 56 E4              JSR     CIO
240A 60                    RTS
                           .IF     SYSRES
                           ;
                           ; LOAD - THIS SUBROUTINE
                           ;       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.
240B A2 10         LOAD    LDX     #$10
240D A9 03                 LDA     #OPEN
240F 9D 42 03              STA     ICCOM,X
2412 A9 04                 LDA     #4            OPEN TYPE=INPUT
2414 9D 4A 03              STA     ICAX1,X
2417 20 56 E4              JSR     CIO           TRY TO OPEN FILE
241A 10 04                 BPL     RDLF          CONT IF OK
241C A9 01                 LDA     #1            OPEN ERRORS
241E D0 4F                 BNE     CLFX          CLOSE AND EXIT
2420 A2 10         RDLF    LDX     #$10
2422 A9 A1                 LDA     #DBUFL
2424 9D 44 03              STA     ICBAL,X
2427 A9 13                 LDA     #DBUFH
2429 9D 45 03              STA     ICBAH,X
242C A9 02                 LDA     #2
242E 9D 48 03              STA     ICBLL,X
2431 A9 00                 LDA     #0
2433 9D 49 03              STA     ICBLH,X
2436 A9 06                 LDA     #GETCHR
2438 9D 42 03              STA     ICCOM,X
```

```
243E 30 45          BMI     ERST      IF ERRS
2440 A9 84          LDA     #$84
2442 CD A1 13       CMP     DBUF      CHECK FOR VALID LOAD FILE
2445 D0 3A          BNE     LNLF
2447 A9 09          LDA     #$09
2449 CD A2 13       CMP     DBUF+1
244C D0 33          BNE     LNLF
244E A2 10    RDDRC LDX     #$10
2450 A9 A1          LDA     #DBUFL
2452 9D 44 03       STA     ICBAL,X
2455 A9 13          LDA     #DBUFH
2457 9D 45 03       STA     ICBAH,X
245A A9 04          LDA     #4
245C 9D 48 03       STA     ICBLL,X
245F A9 00          LDA     #0
2461 9D 49 03       STA     ICBLH,X
2464 20 56 E4       JSR     CIO       NO ERROR CHECK SO CAN CATCH EOF
2467 10 20          BPL     STOX      IF NO ERROR
2469 C0 88          CPY     #$88      SEE IF EOF
246B D0 18          BNE     ERST      IF SOME ERROR STATUS
              ;EOF SO DONE, EXIT
246D A9 00          LDA     #0        OK STATUS
246F 48       CLFX  PHA
2470 98             TYA
2471 48             PHA
2472 A2 10          LDX     #$10
2474 A9 0C          LDA     #CLOSE
2476 9D 42 03       STA     ICCOM,X
2479 20 56 E4       JSR     CIO
247C 68             PLA
247D A8             TAY
247E 68             PLA
247F AA             TAX
2480 60             RTS
2481 A9 03    LNLF  LDA     #3        HAD LOAD FILE
2483 D0 EA          BNE     CLFX
2485 A9 02    ERST  LDA     #2        READ ERRS
2487 D0 E6          BNE     CLFX
2489 A2 10    STOX  LDX     #$10
248B AD A1 13       LDA     DBUF      MOVE PARAMS TO IOCB
248E 9D 44 03       STA     ICBAL,X
2491 AD A2 13       LDA     DBUF+1
2494 9D 45 03       STA     ICBAH,X
2497 AD A3 13       LDA     DBUF+2
249A 38             SEC
249B ED A1 13       SBC     DBUF
249E 9D 48 03       STA     ICBLL,X
24A1 AD A4 13       LDA     DBUF+3
24A4 ED A2 13       SBC     DBUF+1
24A7 9D 49 03       STA     ICBLH,X
24AA FE 48 03       INC     ICBLL,X
24AD D0 03          BNE     *+5
24AF FE 49 03       INC     ICBLH,X
24B2 20 56 E4       JSR     CIO       READ DATA DIRECITLY TO MEMORY
24B5 30 CE          BMI     ERST      IF ERRS
24B7 4C 4E 24       JMP     RDDRC     GO READ NEXT DATA REC.
              .ENDIF
```

COLLEEN DISK UTILITIES

```
                                    ;         RETURN A=LO, X=HI. PER SET MINUS IF ERROR.
                                    ;         INC IPTR PAST PARAM.
                                    ;
24BA A9 04            GETNO    LDA    #4         MAX NO DIGITS
24BC 8D B0 14                  STA    CTR
24BF A9 00                     LDA    #0
24C1 8D B2 14                  STA    T1
24C4 8D B3 14                  STA    T1+1       INIT TEMP TO BUILD NUMBER IN
24C7 AE AF 14         GHB      LDX    IPTR
24CA BD 51 13                  LDA    LINE,X     GET CHAR
24CD EE AF 14                  INC    IPTR
24D0 C9 9B                     CMP    #CR        SEE IF TRMINATOR
24D2 F0 2B                     BEQ    GND
24D4 C9 2C                     CMP    #',
24D6 F0 27                     BEQ    GND
24D8 20 3A 25                  JSR    HEXCON     CONVERT ASCII TO NIBBLE
24DB 30 29                     BMI    ERRX       IF ERROR
24DD A0 03                     LDY    #3         SHIFT T1,T1+1 BY 4
24DF 18               SHT1     CLC
24E0 2E B3 14                  ROL    T1+1
24E3 2E B2 14                  ROL    T1
24E6 88                        DEY
24E7 10 F6                     BPL    SHT1
24E9 0D B3 14                  ORA    T1+1       OR IN NEW NIBBLE
24EC 8D B3 14                  STA    T1+1
24EF CE B0 14                  DEC    CTR        COUNT DIGIT
24F2 10 D3                     BPL    GHB        LOOP UNLESS TOO MANY DIGITS
24F4 A9 0C                     LDA    #TMDL
24F6 A2 25                     LDX    #TMDH
24F8 20 61 23         ERRX1    JSR    DSPLIN
24FB CE A2 14                  DEC    PER
24FE 60                        RTS
24FF AD B3 14         GND      LDA    T1+1
2502 AE B2 14                  LDX    T1
2505 60                        RTS
2506 A9 1C            ERRX     LDA    #IHPL      INVALID HEX PARAM
2508 A2 25                     LDX    #IHPH
250A D0 EC                     BNE    ERRX1
250C 54 4F 4F         TMD      .BYTE  "TOO MANY DIGITS",CR
250F 20 4D 41
2512 4E 59 20
2515 44 49 47
2518 49 54 53
251B 9B
                      HILO     TMD
0025                  TMDH     =      TMD/256
000C                  TMDL     =      -256*TMDH+TMD
251C 49 4E 56         IHP      .BYTE  "INVALID HEXADECIMAL PARAMETER",CR
251F 41 4C 49
2522 44 20 48
2525 45 58 41
2528 44 45 43
252B 49 4D 41
252E 4C 20 50
2531 41 52 41
2534 4D 45 54
2537 45 52 9B
                      HILO     IHP
0025                  IHPH     =      IHP/256
```

```
                        ;
                        ; HEXCON - CONVERT ASCII CHAR IN A TO HEX NIBBLE IN A. RETURN
                        ;          MINUS CONDITION. A=FF IF ERROR.
                        ;
253A 38         HEXCON  SEC
253B E9 30              SBC     #'0
253D 30 0F              BMI     ERRX2   ASCII BELOW '0'
253F C9 0A              CMP     #10
2541 30 0D              BMI     OKX     0-9 CONVERTED SO EXIT
2543 38                 SEC
2544 E9 07              SBC     #'A-'0-10
2546 C9 0A              CMP     #10     CONVERTED VALUE MUST BE 10 OR MORE
2548 30 04              BMI     ERRX2   BETWEEN '9' AND 'A'
254A C9 10              CMP     #$10
254C 30 02              BMI     OKX     A-F CONVERTED
254E A9 FF      ERRX2   LDA     #$FF
2550 C9 00      OKX     CMP     #0      SET STATUS BY VALUE IN A
2552 60                 RTS
```

```
                                     ;      GETDN - GET A DEVICE NUMBER FROM LINE(IPTR)
                                     ;          RETURN IT IN A
                                     ;
2553 2C A2 14        GETDN    BIT     PER        SEE IF PARAM ERROR ALREADY
2556 30 27                    BMI     GDR        IF SO DONT BOTHER
2558 AE AF 14                 LDX     IPTR
255B BD 51 13        GETD     LDA     LINE,X
255E E8                       INX
255F C9 44                    CMP     #'D        IF DN
2561 F0 FB                    BEQ     GETD       GO GET DIGIT
2563 38                       SEC
2564 E9 30                    SBC     #'0        CONVERT DIGIT
2566 F0 1B                    BEQ     BDS        CANT BE ZERO
2568 30 16                    BMI     BDS        IF NOT DIGIT
256A C9 05                    CMP     #5
256C 10 12                    BPL     BDS        TOO LARGE
256E 48                       PHA
256F BD 51 13        GD1      LDA     LINE,X
2572 E8                       INX
2573 C9 2C                    CMP     #',
2575 F0 04                    BEQ     GDX        IF TERMINATOR
2577 C9 9B                    CMP     #CR
2579 D0 F4                    BNE     GD1        KEEP LOOKING
257B 8E AF 14        GDX      STX     IPTR       ADVANCE POINTER
257E 68                       PLA
257F 60              GDR      RTS
2580 CE A2 14        BDS      DEC     PER
2583 A9 8B                    LDA     #NDSL      NEED DEVICE SPEC MSG
2585 A2 25                    LDX     #NDSH
2587 20 61 23                 JSR     DSPLIN
258A 60                       RTS
258B 4E 45 45        NDS      .BYTE   "NEED D1 THRU D4",CR
258E 44 20 44
2591 31 20 54
2594 48 52 55
2597 20 44 34
259A 9B
                              HILO    NDS
0025                 NDSH     =       NDS/256
008B                 NDSL     =       -256*NDSH+NDS
                     ;
                     ; DRH - DEVICE REDIRECTION HANDLER
                     ;
                     ;      WHEN LOGICAL DEVICES ARE CREATED, THE HANDLER ADDRESS
                     ;      POINTS HERE. THE DRH LOOKS UP THE PHYSICAL DEVICE SPEC
                     ;      AT OPEN TIME AND CALLS THE APPROPRIATE HANDLER. THE IOCB
                     ;      HANDLER ID IS THEN SET TO THAT HANDLER SO CIO CALLS AFTER
                     ;      THE OPEN GO DIRECTLY TO THAT HANDLER.
                     ;
259B                 DRH      =*
                              HILO    DRH
0025                 DRHH     =       DRH/256
009B                 DRHL     =       -256*DRHH+DRH
                     ;
                     ;      HANDLER VECTORS (ONLY OPEN IS USED)
                     ;
259B F7 25                    .WORD   EOPEN-1
259D 00 00 00        LUNI     .BYTE   0,0,0,0,0
```

```
                              *=*+5          UNUSED
        25A7 60               RTS            INIT ROUTINE
                         ;
        25A8             PDST  *=*+80         TABLE OF PHYSICAL DEV:FILSPECS
                              HILO    PDST
        0025             PDSTH =       PDST/256
        00A8             PDSTL =       -256*PDSTH+PDST
                         ;
        25F8 A0 00       EDPEN LDY     #0
        25FA B1 24             LDA     (ICBALZ),Y      GET LOGICAL DEVICE NAME
        25FC 20 2E 1C          JSR     FLDS    SEARCH LOG DEV NAME TABLE
        25FF 10 03            BPL      FLD     IF FOUND IT OK
                         ;INTERNAL ERROR IF WE GOT HERE.
        2601 A0 AC       IERR  LDY     #$AC    NEW ERROR
        2603 60          HERRX RTS
        2604 8A          FLD   TXA             CREATE POINTER TO PHYSICAL DEVSPEC
        2605 A6 2E             LDX     ICIDNO  GET IOCB #
        2607 0A               ASL      A
        2608 0A               ASL      A
        2609 0A               ASL.     A
        260A 0A               ASL      A
        260B 69 A8             ADC     #PDSTL
        260D 85 24             STA     ICBALZ
        260F 9D 44 03          STA     ICBAL,X FOR FMS
        2612 A9 00             LDA     #0
        2614 69 25             ADC     #PDSTH
        2616 85 25             STA     ICBAHZ
        2618 9D 45 03          STA     ICBAH,X FOR FMS
        261B 20 9E E6          JSR     DEVSRC  GO FIND PHYS DEVICE HID/DNO
        261E B0 E3             BCS     HERRX   IF ERRORS
        2620 A5 21             LDA     ICDNOZ
        2622 9D 41 03          STA     ICDNO,X FOR FMS
        2625 20 3D E6          JSR     COMENT  GET PHYS DEV HDLR OPEN ENTRY
        2628 B0 D9             BCS     HERRX
        262A 20 89 E6          JSR     GOHAND  GO DO REAL OPEN
        262D 60               RTS      THATS ALL
                              .END
```

SYMBOL TABLE

AAM	1F87	ADDC	22E8	AF	131D	AFH	0013
AFL	001D	ASP	1FA4	ASPT	1ED2	ASPX	1FBA
HDS	2580	BLF	1CE5	BLFH	001C	BLFL	00E5
BRKKEY	0011	BRMQ	1B3E	BRUN	1B2C	BS1	1BD1
DB2	1BD1	BSM	1BD7	BSMH	001B	BSML	00D7
CARTST	BFFA	CBIT	1FA0	CDES	14AC	CDN	001D
CDSK	1AF2	CEND	1E06	CFTE	22F4	CFTE1	22F9
CFTE2	2316	CHRO1	2238	CHRO2	224F	CHRO3	2260
CHRGET	2233	CIE	23ED	CIEH	0023	CIEL	00EB
CIEX	23E2	CIO	E456	CIO1	2383	CIOCL	23C2
CIOER	23CA	CIOER1	23C9	CIOGET	220D	CKMEM	1FF2
CKRS	1A27	CLD1	1FDA	CLDXH	1FD2	CLF	001E
CLFX	246F	CLOC	1A33	CLOOP	19FA	CLDS1	1BDB
CLOSE	000C	CLOSX	23FB	CLSCR	007D	COMENT	E63D
CPMQ	190D	CPSL	1C12	CPTR1	2008	CPYFIL	1936
CR	009B	CRD	1C0A	CRT	001F	CSRC	14AB
CSRCH	0014	CSRCL	00AB	CTNS	23CF	CIR	14B0
CUP	001C	DB1	1421	DB1H	0014	DB1L	0021
DB3	139E	DB3H	0013	DB3L	009E	DBLH	0001
DBLL	0000	DBUF	13A1	DBUFH	0013	DBUFHI	0305
DBUFL	00A1	DBUFLO	0304	DCB	0300	DCOMND	0302
DD1	1BA0	DDMQ	1D5A	DDSK	1AEF	DDXT	1FED
DEDE	1B55	DEFDEV	1B76	DELETE	0021	DELFIL	17BA
DELX	18D5	DEMQ	18FC	DEVSRC	E69E	DF1	17E3
DFCX	2072	DIRLST	172D	DKIND	E453	DKWRT	1FD0
DIL	009C	DLMQ	179B	DLSTO	1788	DLST1	178B
DMEND	1607	DMENU	14C7	DMENUH	0014	DMENUL	00C7
DOCPY	19DA	DORD	1E9C	DOS	1300	DOSOS	1625
DOSOSH	0016	DOSOSL	0025	DOSVEC	000A	DOSHDP	1F00
DOW	1EBE	DPFM	2009	DRH	259B	DRIH	0025
DRIL	009B	DRTS	1FEC	DS	1CA7	DSH	001C
DSHI	030B	DSHIH	0003	DSHIL	0008	DSKUTL	1650
DSL	00A7	DSLO	030A	DSLOH	0003	DSLOL	000A
DSPL1	2368	DSPLIN	2361	DSPMSG	2356	DSTATS	0303
DSX	1BC0	DTH	14BB	DTHH	0014	DTHL	00BB
DU1	1650	DU3	1A51	DU4	1A44	DUJPT	1607
DUJPTH	0016	DUJPTL	0007	DULEN	0140	DULENH	0001
DULENL	0040	DUNIT	0301	DUNUM	000F	DUPDSK	1DCC
DUPFIL	201F	DWQ	17F7	EC	21EE	ECH	0021
EC1	00EE	EDH	0014	EDL	00C1	EDN	14C1
ENUF	1E66	EDC	2331	EOPEN	25F8	ERRX	2506
ERRX1	24F8	ERRX2	254E	ERST	2485	ETN	23F5
EUN	23F6	FDP	1AF1	FDPH	001A	FDPL	00F1
FLD	2604	FLDS	1CPE	FLDX	1C3B	FLL	1C30
FMINIT	0788	FMS	0700	FMIDSK	1A87	FMX	1ABD
FORMAT	00FE	QC1	19DC	GD1	256F	GIR	257F
GDX	2578	GETCHR	0006	GETD	2558	GETDN	2553
GETFIL	229E	GETIC1	2284	GETLIN	21F1	GETNAM	2290
GETNO	24BA	GETRCD	0004	GHB	24C7	GLF	1759
GND	24FF	GOHAND	E689	GTJ	22DD	GTP	22D5
HATABS	031A	HATS	1C21	HATSL	1C23	HATX	1C2D
HERRX	2603	HEXCON	253A	IBD	1E0E	IRDH	001E
IRDL	000E	ICAX1	034A	ICAX2	034B	ICBAH	0345
ICBAHZ	0025	ICBAL	0344	ICBALZ	0024	ICBLH	0349
ICBLL	0348	ICCOM	0342	ICDNO	0341	ICDNOZ	0021
ICI	0340	ICHIDZ	0020	ICIDNO	002E	ICSTA	0343
IDD	1F63	IDDH	001F	IDDL	0063	IHRD	182A

IOD	1EC1	IPTR	14AF	IPTRH	0014	IPTRL	00AF
IRQEN	D20E	ISCAR	1B06	ISD	1F44	ISDH	001F
ISDL	0044	JMPTBL	0018	KDN	14C4	KDNH	0014
KDNL	00C4	LBUFH	0013	LBUFL	0051	LDFIL	1CB1
LDF-X	1CD5	LDND	14BD	LDNT	259D	LDST	14BB
LFMO	1CF3	LINE	1351	LKFIL	1D08	LKMO	1D1D
LNLF	2481	LOAD	240B	LOADV	1319	LOCK	0023
LRS	1EAE	LRS1	1FAB	MAXDEV	0021	MDN1	187C
MDN2	188C	MDN3	1894	MEMLO	02E7	MEMTOP	02E5
MENUSL	1691	MENUSZ	14A1	MNA	14B6	MNL	14B8
MNSL	1691	MNSLH	0016	MNSLL	0091	NCA	1B1E
NCAH	001B	NCAL	001E	NDF	17D3	NDFH	0017
NDFL	00D3	NDS	25BB	NDSH	0025	NDSL	008B
NIF	1CD8	NORM	1E5C	NRM	1F34	NRMH	001F
NRML	0034	NSEC	1DF5	NSI	16E2	NSIH	0016
NSIL	00E2	NTL	2348	NTLH	0023	NTLL	0048
NXIOCB	21CD	NXS	1FB2	ODMS	19AD	OF	191D
OFH	0019	DEL	001D	OK	1D77	OKH	001D
OKL	0077	OKS	207F	OKX	2550	OL	2264
OLH	0022	OLL	0064	OPDES	1987	OPDES1	19C3
OPDES3	19C5	OPEN	0003	OPSRC	1971	OPSRC1	1982
IOPT	14B1	PAR	1329	PARH	0013	PARL	0029
PDES	1998	PDES1	19A2	PDST	25A8	PDSTH	0025
PDSTL	00A8	PER	14A2	PERX	2279	PERX1	227F
PIOCB	23D1	PMES	1718	PMESH	0017	PMESL	0018
POKMSK	0010	POPT	2338	PSMEN	16FB	PSRC	1959
PTR	14AE	PTRH	0014	PTRL	00AE	PUTCHR	000A
PUTRCD	0008	QWMO	1CB8	RAMLO	001A	RANGE	16DB
RCNT	14A4	RDDRC	244E	RDFN	1857	RDLF	2420
RDM	14BA	RENAME	0020	RENFIL	1A54	RNMO	1A6C
RSEC	1FBB	RSEC1	1FC5	RUNAD	02E0	RVTOC	1D9A
SAVFIL	20DB	SAVH	14BF	SAVHH	0014	SAVHL	00BF
SAVX	14AD	SCMO	1B2B	SCROL	239E	SDP	1E2D
SETVBV	E45C	SFMO	21A0	SHFLOK	02BE	SHMEN	1671
SHT1	24DF	SIT	16EF	SITH	0016	SITL	00EF
SSTAT	14A5	STCAR	1AF5	STDD	1EE9	STDD1	1EF8
STDD2	1EEE	STE	2325	STF	1BBC	STOK	2489
STVEC	14B4	SU1	1BF5	SU2	1BEF	SWATH	1F2F
SWATL	1F2A	SWDP	14A6	SWLOP	1F02	SYSRES	0001
SYSVBV	E45F	SYVBL	E45F	SYVBLH	00E4	SYVBLL	005F
TI	14B2	TEXT	0001	TF	1BC6	TFH	001B
TFL	00C6	TMD	250C	TMDH	0025	TMDL	000C
TYG	18E6	TYGH	0018	TYGL	00E6	ULFIL	1D30
ULMO	1D45	UNLOCK	0024	UNNO	14A3	USRDOS	1700
VFM	1AD7	VFMH	001A	VFML	00D7	WARMST	0008
WBMO	1C6F	WBMOH	001C	WBMOL	006F	WROOT	1C39
WBX	1C6C	WDR	214E	WHD	1AC0	WHDH	001A
WHDL	00C0	XBLK	1EA5	XITVBV	E462	XTVBL	E462
XTVBLH	00E4	XTVBLL	0062	ZAP	23A7	ZAPH	0023
ZAPL	00A7						