```
                    0001  ; Disk Copy Utility from CPM Users Group
                    0002  ; Modified for CDOS by Trevor Marshall
                    0003  ;                        Elec Eng Dept
                    0004  ;                        Uni W.A.
                    0005  ; Jan 1980
                    0006  ;
                    0007  ; Rewritten for 2.36 CDOS (Double Density)
                    0008  ; Automatic disk label checking,
                    0009  ;   Sept 1980,  Trevor Marshall
                    0010  ;
                    0011  ;
      (0100)        0012          ORG     100H
0100  C34201    R   0013          JP      COPY
                    0014  ; BIOS FUNCTION CALLING TABLE
0103  CD2D01        0015  WBOOT:  CALL    BIOSGO
0106  CD2D01        0016  CONST:  CALL    BIOSGO
0109  CD2D01        0017  CONIN:  CALL    BIOSGO
010C  CD2D01        0018  CONOUT: CALL    BIOSGO
010F  CD2D01        0019  LIST:   CALL    BIOSGO
0112  CD2D01        0020  PUNCH:  CALL    BIOSGO
0115  CD2D01        0021  READER: CALL    BIOSGO
0118  CD2D01        0022  HOME:   CALL    BIOSGO
011B  CD2D01        0023  SELDSK: CALL    BIOSGO
011E  CD2D01        0024  SETTRK: CALL    BIOSGO
0121  CD2D01        0025  SETSEC: CALL    BIOSGO
0124  CD2D01        0026  SETDMA: CALL    BIOSGO
0127  CD2D01        0027  READ:   CALL    BIOSGO
012A  CD2D01        0028  WRITE:  CALL    BIOSGO
012D  E3            0029  BIOSGO: EX      (SP),HL      ; Get call addr in HL, save
                           HL on stack
012E  D5            0030          PUSH    DE           ; Save DE
012F  EB            0031          EX      DE,HL        ; Move call addr to DE
0130  2A0100        0032          LD      HL,(1)       ; Get BIOS entry address
0133  19            0033          ADD     HL,DE        ; Add call addr to entry ad
                         dr
0134  110601        0034          LD      DE,[WBOOT+3]; Get start of table
                    0035  ; Subtract DE from HL in 8080 code (for compatibility)
0137  7B            0036          LD      A,E
0138  2F            0037          CPL
0139  5F            0038          LD      E,A
013A  7A            0039          LD      A,D
013B  2F            0040          CPL
013C  57            0041          LD      D,A     ;Now have 1s compl of DE
013D  13            0042          INC     DE      ;2s compl
013E  19            0043          ADD     HL,DE   ;Done!
013F  D1            0044          POP     DE           ; Restore DE
0140  E3            0045          EX      (SP),HL      ; Restore HL, put jump addr
                           on stack
0141  C9            0046          RET                  ; Jump to BIOS routine
                    0047  ; Original coding by:
                    0048  ;         L.E. HUGHES     8080SDC     77/10/29
                    0049  ;
                    0050  ; Modified by   Trevor Marshall
                    0051  ;               E.E.Dept
                    0052  ;               Uni W.A.
                    0053  ;
                    0054  ; to work with any sized CDOS system
```

```
0055  ; and to prompt for drives
0056  ;
0057
0058  ;          MISC SYMBOLS
```

```
                    0059
      (000A)        0060  LF      EQU    0AH   ;LINE FEED
      (000D)        0061  CR      EQU    0DH   ;CARRIAGE RETURN
                    0062
      (0000)        0063  ITRK    EQU    0     ;INITIAL TRACK TO COPY
      (004C)        0064  LTRK    EQU    76    ;LAST TRACK TO COPY
                    0065
      (0005)        0066  BDOS:   EQU     5
                    0067  ;
0142  210000        0068  COPY:   LD      HL,0
0145  39            0069          ADD     HL,SP
0146  22E006        0070          LD      (OLDSP),HL
0149  312207        0071          LD      SP,STACK+64
                    0072  ;
                    0073  ;         ALLOW USER TO MOUNT DISK(S) BEFORE PROCEEDING
                    0074
014C  0E96          0075  AGAIN:  LD      C,96H   ;Turn drive motors off
014E  CD0500        0076          CALL    BDOS
                    0077  ;
0151  219704        0078          LD      HL,CRLF
0154  CDFE02        0079          CALL    WASC
0157  215504        0080          LD      HL,STR1 ;PRINT 'Source Disk Drive,etc'
015A  CDFE02        0081          CALL    WASC
015D  CD1503        0082          CALL    RACC
0160  FE03          0083          CP      3       ;CTL-C ABORT
0162  CAF702        0084          JP      Z,EXIT1
0165  CD0803        0085          CALL    WACC
0168  D641          0086          SUB     'A'
016A  DA4C01    R   0087          JP      C,AGAIN ;Invalid entry
016D  FE03          0088          CP      3
016F  D24C01    R   0089          JP      NC,AGAIN
0172  322E07        0090          LD      (SOURCE),A
0175  219704        0091          LD      HL,CRLF
0178  CDFE02        0092          CALL    WASC
                    0093  ; Now prompt for destination
017B  219A04        0094          LD      HL,STR1A
017E  CDFE02        0095          CALL    WASC
0181  CD1503        0096          CALL    RACC
0184  CD0803        0097          CALL    WACC
0187  D641          0098          SUB     'A'
0189  DA4C01    R   0099          JP      C,AGAIN
018C  FE03          0100          CP      3
018E  D24C01    R   0101          JP      NC,AGAIN
0191  322F07        0102          LD      (DEST),A
0194  212E07        0103          LD      HL,SOURCE
0197  4E            0104          LD      C,(HL)
0198  B9            0105          CP      A,C
0199  CA4C01    R   0106          JP      Z,AGAIN
019C  219704        0107          LD      HL,CRLF
019F  CDFE02        0108          CALL    WASC
                    0109
                    0110
                    0111  ; Now we must determine whether single or double
                    0112  ;  density disks are in use, and check that we
                    0113  ;  are not trying to copy between incompatible formats
                    0114  ;
                    0115  ; We will use the CDOS 1BH cal (Get disk allocation vec)
```

```
0116  ;   as the disk labels are read before this call returns
          .
0117  ; This call returns the number of clusters in DE
0118  ;
```

```
                    0119  ;
01A2  0E0D          0120          LD      C,0DH   ;Reset CDOS
01A4  CD0500        0121          CALL    BDOS
                    0122  ;
01A7  3A2E07        0123          LD      A,(SOURCE) ;Select Source drive
01AA  5F            0124          LD      E,A
01AB  0E0E          0125          LD      C,0EH
01AD  CD0500        0126          CALL    BDOS
                    0127  ;
01B0  0E1B          0128          LD      C,1BH   ;Get CLUSTER.SIZE size map
01B2  CD0500        0129          CALL    BDOS    ;for first disk
01B5  ED532207      0130          LD      (ACLUSTERS),DE
                    0131  ;
01B9  3A2F07        0132          LD      A,(DEST) ;Select Dest. drive
01BC  5F            0133          LD      E,A
01BD  0E0E          0134          LD      C,0EH
01BF  CD0500        0135          CALL    BDOS
                    0136  ;
01C2  0E1B          0137          LD      C,1BH   ;Get cluster size map
01C4  CD0500        0138          CALL    BDOS    ;for second disk
01C7  ED532407      0139          LD      (BCLUSTERS),DE ;save it
01CB  2A2207        0140          LD      HL,(ACLUSTERS) ;get it again
01CE  AF            0141          XOR     A
01CF  ED52          0142          SBC     HL,DE   ;to compare them
01D1  7D            0143          LD      A,L
01D2  B5            0144          OR      A,L     ;see if HL is zero
01D3  C20802      R 0145          JP      NZ,SEND.DIFF.ERROR ;no
                    0146  ; Now display the disks in use and branch to copy S/R
01D6  7B            0147          LD      A,E     ;LSB of # clusters
01D7  FEFE          0148          CP      A,0FEH  ;D/D S/S?
01D9  CA2B03        0149          JP      Z,SEND.DD.MSG
01DC  FE60          0150          CP      A,060H  ;D/D D/S
01DE  CA2103        0151          JP      Z,SEND.DS.MSG
01E1  FEF3          0152          CP      A,0F3H  ;S/D S/S
01E3  2808          0153          JR      Z,SEND.SD.MSG
01E5  FEF7          0154          CP      A,0F7H  ;S/D D/S
01E7  CA2103        0155          JP      Z,SEND.DS.MSG   ;Dont use this often,
                    0156                          ;I wont bother to tidy it up
01EA  C31302      R 0157          JP      SEND.NON.STANDARD.MSG
                    0158  ;
01ED  114806        0159  SEND.SD.MSG: LD DE,SD.MSG
01F0  0E09          0160          LD      C,9
01F2  CD0500        0161          CALL    BDOS ;Print it
01F5  3E1B          0162          LD      A,1BH   ;Max sectors/trk+1
01F7  322907        0163          LD      (SECTORS.PER.TRACK),A
01FA  218000        0164          LD      HL,80H  ;The sector size for S/D
01FD  222A07        0165          LD      (DMA.INCR),HL
0200  21C506        0166          LD      HL,SDMAP-1 ;Point to sector map
0203  222C07        0167          LD      (SECTOR.MAP.ADDRESS),HL
0206  1810          0168          JR      MAIN
                    0169  ;
0208  118706        0170  SEND.DIFF.ERROR: LD DE,DIFF.ERROR.MSG
020B  0E09          0171  L5:     LD      C,9
020D  CD0500        0172          CALL    BDOS
0210  C34C01        0173          JP      AGAIN
0213  11A306        0174  SEND.NON.STANDARD.MSG: LD DE,NON.STANDARD.ERROR.MSG
0216  18F3          0175          JR      L5
```

```
                    0176  ;
                    0177  ;          BEGIN SINGLE DENSITY LOOP
                    0178
0218  117706       0179  MAIN:   LD      DE,MSG1 ;Finish disk msg
```

```
021B  0E09        0180          LD      C,9
021D  CD0500      0181          CALL    BDOS
                  0182  ;
0220  211A05      0183          LD      HL,STR4       ;PRINT HEADER
0223  CDFE02      0184          CALL    WASC
0226  216705      0185          LD      HL,STR5
0229  CDFE02      0186          CALL    WASC
                  0187  ;
022C  3E00        0188          LD      A,ITRK        ;INITIAL TRACK NUMBER
022E  323007      0189          LD      (TRKNO),A
                  0190  COPY2:
                  0191  ; Check for CTL-C abort
0231  CD0601      0192          CALL    CONST
0234  FE00        0193          CP      0
0236  CA4302    R 0194          JP      Z,FT1
0239  CD0901      0195          CALL    CONIN
023C  E65F        0196          AND     5FH
023E  FE03        0197          CP      3       ;CTL-C
0240  CAEE02      0198          JP      Z,COPYX
0243  3A2E07      0199  FT1:    LD      A,(SOURCE) ;Select Source Disk
0246  4F          0200          LD      C,A
0247  CD1B01      0201          CALL    SELDSK
024A  3A3007      0202          LD      A,(TRKNO)     ;SET TRACK NUMBER
024D  4F          0203          LD      C,A
024E  CD1E01      0204          CALL    SETTRK
0251  213A07      0205          LD      HL,TBUF       ;SET INITIAL DMA ADDRESS
0254  223207      0206          LD      (DMAPTR),HL
0257  3E01        0207          LD      A,1           ;INITIAL SECTOR NUMBER
0259  323107      0208          LD      (SECNO),A
025C  3A3107      0209  COPY3:  LD      A,(SECNO)     ;SET SECTOR NUMBER
025F  2A2C07      0210          LD      HL,(SECTOR.MAP.ADDRESS)
0262  1600        0211          LD      D,0
0264  5F          0212          LD      E,A
0265  19          0213          ADD     HL,DE
0266  4E          0214          LD      C,(HL)
0267  CD2101      0215          CALL    SETSEC
026A  2A3207      0216          LD      HL,(DMAPTR)   ;SET DMA ADDRESS
026D  44          0217          LD      B,H
026E  4D          0218          LD      C,L
026F  CD2401      0219          CALL    SETDMA
0272  CD2701      0220          CALL    READ          ;READ SECTOR
0275  2A3207      0221          LD      HL,(DMAPTR)   ;ADD INCR TO DMA PTR
0278  D5          0222          PUSH    DE
0279  ED5B2A07    0223          LD      DE,(DMA.INCR)
027D  19          0224          ADD     HL,DE
027E  D1          0225          POP     DE
027F  223207      0226          LD      (DMAPTR),HL
0282  213107      0227          LD      HL,SECNO      ;ADD 1 TO SECTOR NUMBER
0285  34          0228          INC     (HL)
0286  7E          0229          LD      A,(HL)
0287  212907      0230          LD      HL,SECTORS.PER.TRACK
028A  BE          0231          CP      A,(HL) ;LOOP THRU ENTIRE TRACK
028B  DA5C02    R 0232          JP      C,COPY3
                  0233
                  0234  ;         WRITE TBUF TO CURRENT TRACK ON DISK C
                  0235
028E  3A2F07      0236          LD      A,(DEST) ;Select dest dsk
```

```
0291  4F          0237          LD      C,A
0292  CD1B01      0238          CALL    SELDSK
0295  3A3007      0239          LD      A,(TRKNO)     ;SET TRACK NUMBER
0298  4F          0240          LD      C,A
```

```
0299  CD1E01      0241          CALL    SETTRK
029C  213A07      0242          LD      HL,TBUF       ;SET DMA ADDRESS
029F  223207      0243          LD      (DMAPTR),HL
02A2  3E01        0244          LD      A,1           ;SET INITIAL SECTOR NUMBER
02A4  323107      0245          LD      (SECNO),A
02A7  3A3107      0246  COPY4:  LD      A,(SECNO)     ;SET SECTOR NUMBER
02AA  2A2C07      0247          LD      HL,(SECTOR.MAP.ADDRESS)
02AD  1600        0248          LD      D,0
02AF  5F          0249          LD      E,A
02B0  19          0250          ADD     HL,DE
02B1  4E          0251          LD      C,(HL)
02B2  CD2101      0252          CALL    SETSEC
02B5  2A3207      0253          LD      HL,(DMAPTR)   ;SET DMA ADDRESS
02B8  44          0254          LD      B,H
02B9  4D          0255          LD      C,L
02BA  CD2401      0256          CALL    SETDMA
02BD  CD2A01      0257          CALL    WRITE         ;WRITE SECTOR
02C0  2A3207      0258          LD      HL,(DMAPTR)   ;ADD INCR TO DMAPTR
02C3  D5          0259          PUSH    DE
02C4  ED5B2A07    0260          LD      DE,(DMA.INCR)
02C8  19          0261          ADD     HL,DE
02C9  D1          0262          POP     DE
02CA  223207      0263          LD      (DMAPTR),HL
02CD  213107      0264          LD      HL,SECNO      ;ADD 1 TO SECTOR NUMBER
02D0  34          0265          INC     (HL)
02D1  7E          0266          LD      A,(HL)
02D2  212907      0267          LD      HL,SECTORS.PER.TRACK
02D5  BE          0268          CP      A,(HL) ;LOOP THRU ENTIRE TRACK
02D6  DAA702    R 0269          JP      C,COPY4
                  0270
                  0271  ;         ADVANCE TO NEXT TRACK
                  0272
02D9  3E2A        0273          LD      A,'*'
02DB  CD0803      0274          CALL    WACC
02DE  213007      0275          LD      HL,TRKNO
02E1  34          0276          INC     (HL)
02E2  7E          0277          LD      A,(HL)
02E3  FE4D        0278          CP      LTRK+1        ;LOOP THRU ENTIRE DISK
02E5  DA3102      0279          JP      C,COPY2
                  0280
                  0281  ;         ALL DONE SINGLE DENSITY
                  0282
02E8  21EC04      0283  COPY5:  LD      HL,STR2       ;PRINT 'COPY COMPLETE'
02EB  C3F102    R 0284          JP      COPY6
02EE  21FF04      0285  COPYX:  LD      HL,STR3       ;PRINT 'COPY ABORTED'
02F1  CDFE02      0286  COPY6:  CALL    WASC
02F4  C34201      0287          JP      COPY
02F7  2AE006      0288  EXIT1:  LD      HL,(OLDSP)    ;EXIT TO CP/M
02FA  F9          0289          LD      SP,HL
02FB  C30000      0290          JP      0
                  0291
                  0292  ;         WASC - WRITE ASCII STRING TO CONSOLE
                  0293
02FE  7E          0294  WASC:   LD      A,(HL)
02FF  B7          0295          OR      A
0300  C8          0296          RET     Z
0301  CD0803      0297          CALL    WACC
```

```
0304  23          0298          INC     HL
0305  C3FE02    R 0299          JP      WASC
                  0300
                  0301  ;         WACC - WRITE ASCII CHARACTER TO CONSOLE
```

```
                        0302
0308  E5                0303  WACC:   PUSH    HL
0309  D5                0304          PUSH    DE
030A  C5                0305          PUSH    BC
030B  F5                0306          PUSH    AF
030C  4F                0307          LD      C,A
030D  CD0C01            0308          CALL    CONOUT
0310  F1                0309          POP     AF
0311  C1                0310          POP     BC
0312  D1                0311          POP     DE
0313  E1                0312          POP     HL
0314  C9                0313          RET
                        0314
                        0315  ;         RACC - READ ASCII CHARACTER FROM CONSOLE
                        0316
0315  E5                0317  RACC:   PUSH    HL
0316  D5                0318          PUSH    DE
0317  C5                0319          PUSH    BC
0318  CD0901            0320          CALL    CONIN
031B  E65F              0321          AND     5FH     ;Make LC=UC
031D  C1                0322          POP     BC
031E  D1                0323          POP     DE
031F  E1                0324          POP     HL
0320  C9                0325          RET
                        0326
0321  3E02              0327  SEND.DS.MSG: LD A,2
0323  322807            0328          LD      (SIZE.FLAG),A
0326  115906            0329          LD      DE,DS.MSG
0329  1808              0330          JR      V1
032B  3E01              0331  SEND.DD.MSG: LD A,1
032D  322807            0332          LD      (SIZE.FLAG),A
0330  113706            0333          LD      DE,DD.MSG
0333  0E09              0334  V1:     LD      C,9
0335  CD0500            0335          CALL    BDOS
                        0336  ;
                        0337  ;Calculate the 2's complement of the block capacity of d
                              isk
0338  2A2207            0338          LD      HL,(ACLUSTERS) ;Get # on disk
033B  29                0339          ADD     HL,HL   ; x2 (16 blocks per cluster)
033C  29                0340          ADD     HL,HL   ; x4
033D  29                0341          ADD     HL,HL   ; x8
033E  29                0342          ADD     HL,HL   ; x16
033F  EB                0343          EX      DE,HL   ;into DE
0340  37                0344          SCF             ;add 1 to block capacity
0341  210000            0345          LD      HL,0
0344  ED52              0346          SBC     HL,DE   ;have 2's complement
0346  222607            0347          LD      (BLOCK.CAPACITY),HL
                        0348  ; We cannot use the same approach for D/D disks
                        0349  ;   as for S/D, so we will use a LOGICAL BLOCK
                        0350  ;   approach. Each block is 128 Bytes, I have
                        0351  ;   ~32K free RAM, so will read ~28K at a time
                        0352  ;   (160h, 352d blocks).
                        0353  ;
0349  117706            0354  DOUBLE: LD      DE,MSG1 ;Finish disk msg
034C  0E09              0355          LD      C,9
034E  CD0500            0356          CALL    BDOS
                        0357  ;
```

```
0351  3A2807      0358          LD      A,(SIZE.FLAG) ;Is it D/D or D/S ?
0354  FE01        0359          CP      1       ;D/D only?
0356  200E        0360          JR      NZ,JJ1  ;No, D/S D/D
0358  21B705      0361          LD      HL,STR6 ;PRINT D/D HEADER
```

```
035B  CDFE02      0362          CALL    WASC
035E  21C805      0363          LD      HL,STR7
0361  CDFE02      0364          CALL    WASC
0364  180C        0365          JR      JJ2
                  0366  ;
0366  21DC05      0367  JJ1:    LD      HL,STR8 ;PRINT D/S HEADER
0369  CDFE02      0368          CALL    WASC
036C  210B06      0369          LD      HL,STR9
036F  CDFE02      0370          CALL    WASC
                  0371  ;
0372  210000      0372  JJ2:    LD      HL,0    ;INITIAL BLOCK NUMBER
0375  223407      0373          LD      (BLKNO),HL
                  0374  ;
                  0375  ; Begin the main (fill buffer) loop
0378  21BA06      0376  DCOPY2: LD      HL,TBUF-80H
037B  223207      0377          LD      (DMAPTR),HL ;Initialize DMA ptr
037E  ED5B3407    0378          LD      DE,(BLKNO) ;Block # to DE
0382  ED533607    0379          LD      (FIRSTBLK),DE ;Save first blocK #
                  0380  ;
                  0381  ; Check for CTL-C abort
0386  CD0601      0382          CALL    CONST
0389  FE00        0383          CP      0
038B  CA9803    R 0384          JP      Z,DFT1
038E  CD0901      0385          CALL    CONIN
0391  E65F        0386          AND     5FH
0393  FE03        0387          CP      3       ;CTL-C
0395  CAEE02      0388          JP      Z,COPYX ;and abort
                  0389  ;
                  0390  ; Get enough blocks to fill buffer, one at a time
0398  2A3207      0391  DFT1:   LD      HL,(DMAPTR)  ;ADD INCR TO DMA PTR
039B  118000      0392          LD      DE,80H  ;Block size is 128 bytes
039E  19          0393          ADD     HL,DE
039F  223207      0394          LD      (DMAPTR),HL
                  0395  ; Are we at end of Buffer,
03A2  110080      0396          LD      DE,-8000H
03A5  19          0397          ADD     HL,DE   ;C means .LT. 8000H
03A6  3845        0398          JR      C,DCOPY6 ;If so write the buffer
                  0399  ;Set DMA for DOS
03A8  ED5B3207    0400          LD      DE,(DMAPTR)
03AC  0E1A        0401          LD      C,1AH
03AE  CD0500      0402          CALL    BDOS
                  0403  ; Is this block beyond the end of disk?
03B1  ED5B2607    0404          LD      DE,(BLOCK.CAPACITY)
03B5  2A3407      0405          LD      HL,(BLKNO)
03B8  19          0406          ADD     HL,DE   ;see if DE & HL are .EQ.
03B9  7D          0407          LD      A,L
03BA  B4          0408          OR      A,H
03BB  CAE802      0409          JP      Z,COPY5   ;Z = done copy
                  0410  ; Read the block
03BE  ED5B3407    0411          LD      DE,(BLKNO) ;Block # to DE
03C2  0E83        0412          LD      C,83H
03C4  212E07      0413          LD      HL,SOURCE
03C7  46          0414          LD      B,(HL) ;disk # to B
03C8  04          0415          INC     B       ;For BDOS
03C9  CBF8        0416          SET     7,B     ;Set interleaved read
03CB  CD0500      0417          CALL    BDOS
                  0418  ; Handle error status
```

```
03CE  FE01        0419          CP      A,1     ;I/O error
03D0  CAEE02      0420          JP      Z,COPYX
03D3  FE02        0421          CP      A,2     ;Illegal request
03D5  CAEE02      0422          JP      Z,COPYX
```

```
03D8  FE03        0423          CP      A,3     ;Illegal Block
03DA  CAEE02      0424          JP      EQ,COPYX
                  0425  ; Now incr the block #
03DD  2A3407      0426          LD      HL,(BLKNO)
03E0  23          0427          INC     HL
03E1  223407      0428          LD      (BLKNO),HL
                  0429  ; Is this beyond the end of disk?
03E4  ED5B2607    0430          LD      DE,(BLOCK.CAPACITY)
03E8  19          0431          ADD     HL,DE   ;see if DE & HL are .EQ.
03E9  7D          0432          LD      A,L
03EA  B4          0433          OR      A,H
03EB  20AB        0434          JR      NZ,DFT1   ;Z = done read
                  0435  ; Must decr last BLKNO so write does not overrun
03ED  2A3407      0436  DCOPY6: LD      HL,(BLKNO)
03F0  2B          0437          DEC     HL
03F1  223407      0438          LD      (BLKNO),HL
                  0439  ;
                  0440  ;Write the buffer if full
                  0441  DCOPY3:
                  0442  ; Begin the main (empty buffer) loop
03F4  3E2A        0443  SENDSTAR: LD    A,'*'
03F6  CD0803      0444          CALL    WACC
                  0445  ;
03F9  21BA06      0446          LD      HL,TBUF-80H
03FC  223207      0447          LD      (DMAPTR),HL ;Initialize DMA ptr
                  0448  ; Save last BLK # and get first #
03FF  2A3407      0449          LD      HL,(BLKNO)
0402  223807      0450          LD      (LASTBLK),HL
                  0451  ;
0405  ED5B3607    0452          LD      DE,(FIRSTBLK) ;First Block # to DE
0409  ED533407    0453          LD      (BLKNO),DE ;Save blocK #
                  0454  ;
                  0455  ; Put enough blocks to empty buffer, one at a time
040D  2A3207      0456  DFT2:   LD      HL,(DMAPTR)  ;ADD INCR TO DMA PTR
0410  118000      0457          LD      DE,80H  ;Block size is 128 bytes
0413  19          0458          ADD     HL,DE
0414  223207      0459          LD      (DMAPTR),HL
                  0460  ; Is this beyond the end of disk?
                  0461  ;       LD      DE,(BLOCK.CAPACITY)
                  0462  ;       LD      HL,(BLKNO)      ;get the block #
                  0463  ;       ADD     HL,DE   ;see if DE & HL are .EQ.
                  0464  ;       LD      A,L
                  0465  ;       OR      A,H
                  0466  ;       JP      Z,COPY5   ;Z = done copy
                  0467  ; Have we written too many blocks
0417  2A3807      0468          LD      HL,(LASTBLK)
041A  ED5B3407    0469          LD      DE,(BLKNO)
041E  AF          0470          XOR     A
041F  ED52        0471          SBC     HL,DE
0421  DA7803      0472          JP      C,DCOPY2 ;C if BLKNO > LASTBLK
                  0473  ; On exit BLKNO will be LASTBLK + 1
                  0474  ; Which is OK for next read loop
                  0475  ;Set DMA for DOS
0424  ED5B3207    0476          LD      DE,(DMAPTR)
0428  0E1A        0477          LD      C,1AH
042A  CD0500      0478          CALL    BDOS
                  0479  ;
```

```
                    0480  ; Write the block
042D  ED5B3407      0481          LD      DE,(BLKNO) ;Block # to DE
0431  0E84          0482          LD      C,84H
0433  212F07        0483          LD      HL,DEST
```

```
0436  46          0484          LD      B,(HL) ;disk # to B
0437  04          0485          INC     B       ;for BDOS
0438  CBF8        0486          SET     7,B     ;Set interleaved read
043A  CD0500      0487          CALL    BDOS
                  0488  ; Handle error status
043D  FE01        0489          CP      A,1     ;I/O error
043F  CAEE02      0490          JP      Z,COPYX
0442  FE02        0491          CP      A,2     ;Illegal request
0444  CAEE02      0492          JP      Z,COPYX
0447  FE03        0493          CP      A,3     ;Illegal Block,done
0449  CAEE02      0494          JP      EQ,COPYX
                  0495  ; Now incr the block #
044C  2A3407      0496          LD      HL,(BLKNO)
044F  23          0497          INC     HL
0450  223407      0498          LD      (BLKNO),HL
                  0499  ;
0453  18B8        0500          JR      DFT2    ;Loop for more blocks
                  0501  ;
                  0502  ;
                  0503  ;         OUTPUT STRINGS
                  0504
0455  536F7572    0505  STR1:   DEFB    'Source disk drive (A to D) ?',CR,LF
0473  54797065    0506          DB      'Type <CR> if you make an error --- ',0
0497  0D0A00      0507  CRLF:   DB      cr,lf,0
049A  456E7375    0508  STR1A:  db      'Ensure you mount the disks before'
04BB  74797069    0509          db      'typing the : ',cr,lf
04CA  44657374    0510            DB    'Destination drive (A to D) ? --- ',0
04EC  0D0A436F    0511  STR2:   DEFB    CR,LF,'Copy completed',CR,LF,0
04FF  2A2A2A2A    0512  STR3:   DEFB    '***** Copy aborted *****',CR,LF,0
051A  0D0A0A      0513  STR4:   DEFB    0DH,0AH,0AH
051D  20202020    0514          DB      '          1         2         3
                           4'
0546  20202020    0515          DEFB    '         5         6         7',CR,LF,0
0567  30313233    0516  STR5:   DEFB    '01234567890123456789012345678901234567
                           90'
0590  31323334    0517          DEFB    '12345678901234567890123456789012345
                           6',CR,LF,0
05B7  0D0A0A      0518  STR6:   DB      0DH,0AH,0AH
05BA  20202020    0519          DB      '          1',CR,LF,0
05C8  30313233    0520  STR7:   DB      '01234567890123456',CR,LF,0
05DC  0D0A0A      0521  STR8:   DB      0DH,0AH,0AH
05DF  20202020    0522          DB      '          1         2         3
                           4',CR,LF,0
060B  30313233    0523  STR9:   DB      '01234567890123456789012345678901234567
                           90',CR,LF,0
                  0524  ;
0637  0D0A446F    0525  DD.MSG: DB      0DH,0AH,'Double Density$'
0648  0D0A5369    0526  SD.MSG: DB      0DH,0AH,'Single Density$'
0659  0D0A446F    0527  DS.MSG: DB      0DH,0AH,'Double sided Double Density$'
0677  20646973    0528  MSG1:   DB      ' disks mounted.$'
0687  0D0A2A2A    0529  DIFF.ERR.MSG: DB 0DH,0AH,'**** DISK LABEL ERROR ***$'
06A3  0D0A2A2A    0530  NON.STANDARD.ERROR.MSG: DB 0DH,0AH,'**** DISKS ARE NON S
                           TANDARD ****$'
                  0531  ;
                  0532  ;         SECTOR MAP
                  0533
06C6  01070D13    0534  SDMAP:  DB      1,7,0DH,13H,19H,5,0BH,11H,17H,3,9,0FH,15
```

```
                                H,2
06D4  080E141A    0535          DB      8,0EH,14H,1AH,6,0CH,12H,18H,4,0AH,10H,16
                                H
                  0536  ;DDMAP: DB      1,0CH,7,2,0DH,8,3,0EH,9,4,0FH,0AH,5,10H,
```

```
                        0BH,6
                  0537  ;
06E0  (0002)      0538  OLDSP:  DEFS    2
06E2  (0040)      0539  STACK:  DEFS    64
0722  (0002)      0540  ACLUSTERS: DS   2       ;Clusters on DSK 1
0724  (0002)      0541  BCLUSTERS: DS   2       ;Clusters on DSK 2
0726  (0002)      0542  BLOCK.CAPACITY: DS 2    ;Blocks on the disk
0728  (0001)      0543  SIZE.FLAG: DS   1       ;D/D OR D/S ?
0729  (0001)      0544  SECTORS.PER.TRACK: DS 1
072A  (0002)      0545  DMA.INCR: DS    2
072C  (0002)      0546  SECTOR.MAP.ADDRESS: DS 2
072E  (0001)      0547  SOURCE: DS      1       ;SOURCE DRIVE #
072F  (0001)      0548  DEST:   DS      1       ;DEST DRIVE #
                  0549
0730  (0001)      0550  TRKNO:  DEFS    1            ;TRACK NUMBER
0731  (0001)      0551  SECNO:  DEFS    1            ;SECTOR NUMBER
0732  (0002)      0552  DMAPTR: DEFS    2            ;DMA POINTER
0734  (0002)      0553  BLKNO:  DS      2
0736  (0002)      0554  FIRSTBLK: DS    2
0738  (0002)      0555  LASTBLK: DS     2
                  0556  ;
                  0557  ; THe buffer for D/D is assumed 45 K long
073A  (0D00)      0558  TBUF:   DEFS    26*128       ;TRACK BUFFER
                  0559
143A  (0100)      0560          END     100H

Errors             0
Range Count       14
Parity Count       0
```