

Debian User Forums

<https://forums.debian.net/>

[HOWTO] - Access CP/M Floppy's via libdsk & cpmtools

<https://forums.debian.net/viewtopic.php?t=112244>

[HOWTO] - Access CP/M Floppy's via libdsk & cpmtools

Page 1 of 1

by Ikraemer

Posted: 2014-03-04 23:38

[HOWTO] - Access CP/M Floppy's created with Teledisk using libdsk & cpmtools

This HOWTO: will explain how to Install libdsk Version 1.3.5 & cpmtools Version 2.18, along with the dependencies, that allow READ ONLY access to Teledisk (*.TD0) files with your Debian 7.x install.

BEFORE YOU TRY A COMPILE YOU MUST INSTALL SOME REQUIRED SOFTWARE:

Typically you need to install build-essential, and the headers for the kernel you are running.

Debian Wheezy (Ver 7.x) Packages needed for Software Compile:

1. Build-Essential
2. Linux debian 3.2.0-4-amd64 Headers (or the Headers for your Version)

`uname -r`

will tell you the kernel you are currently running.

`sudo apt-get install build-essential linux-headers-$(uname -r)`

will install the software needed to successfully compile your source code.

Install Additional Required Software to Compile libdsk and cpmtools:

Install libncurses5-dev:

`sudo apt-get install libncurses5-dev`

At this point you may have had several directory changes or updates, and your system likely needs updated with:

`sudo updatedb`

It's a good idea to execute this command at a regular interval, to keep the database updated for the locate command.

DOWNLOAD THE SOURCE:

Create a new subdirectory of your choice to contain the downloaded source code.

Download both source packages from:

<http://www.seasip.info/Unix/LibDsk/libdsk-1.3.5.tar.gz>

[http://www.moria.de/~michael/cpmtools/c18.tar.gz](http://www.moria.de/~michael/cpmtools/c...18.tar.gz)

Create a subdirectory to contain the source. I used /home/larry/Downloads/cpmtools
Copy both source files to /home/larry/Downloads/cpmtools, then extract the source folders

`tar -zxvf libdsk-1.3.5.tar.gz`

`tar -zxvf cpmtools-2.18.tar.gz`

The next step you need to do is to find the "SHELL" and "env" you are using, and how many shells are installed and available.

1. env -- Display the current environment, find what Shell is being used with the following command.....

`env | grep SHELL=`

To get a list of available shells on your system, look in the /etc/shells file. This will list

To get a list of available shells on your system, look in the /etc/shells file. This will list the available shells with their complete path names. You can see the list of available shells by running:

```
cat /etc/shells
```

Additional shells may be available in your distribution's software repositories.

Performing a search for the term "shell" in your package manager should provide you with a list of alternative command line shells.

If you want to experiment with a shell, just for now, you can launch a shell from within the existing shell. For example, if you want to run the tcsh shell, you can run:

```
tcsh
```

When you are finished exploring tcsh you can type exit to return to the default shell.

Once you have found a shell you want to continue using you can switch to the default shell by running the chsh command. For example, if you want to switch to using the tcsh shell, you can run

```
chsh -s /usr/bin/tcsh
```

Please note that you must provide the full directory path to the new shell.

2. locate *.rc -- Find the current logged in user's home shell Configuration file (*.rc)...

use:

```
locate *.rc
```

NOTE: For Debian 11 the command will be

```
find *.rc
```

REF:

http://en.wikipedia.org/wiki/Unix_shell

Bourne shell (sh)

Almquist shell (ash)

Bourne-Again shell (bash)

Debian Almquist shell (dash)

Korn shell (ksh)

Z shell (zsh)

C shell (csh)

TENEX C shell (tcsh)

other shells.....

There may be a .bash_profile file in /home/loginuser along with .bashrc file. You can put configurations in either file, and you can create either if it doesn't exist. But, why two different files? What is the difference?

According to the bash man page, .bash_profile is executed for login shells, while .bashrc is executed for interactive non-login shells.

If .bash_profile exists in /home/user with the following information already inserted:

```
PATH=$PATH:$HOME/bin
```

```
export PATH
```

just append your path modifications here instead of the .bashrc file. (ie. CentOS 6)

You need to list the *.rc file to verify the contents and set the PATHS. I'm ASSUMING a Bash shell.....Your may be different!

3. cat .bashrc -- List the configuration file, then append the proper search paths for the users shell with edit.

```
export LD_LIBRARY_PATH=????????????????????????????????????????
```

```
export LIBRARY_PATH=????????????????????????????????????????
```

```
export C_INCLUDE_PATH=????????????????????????????????????????
```

```
export CXX_INCLUDE_PATH=????????????????????????????????????????
```

```
export CPATH=/usr/lib:/usr/local/lib
```

Mine happens to now be:

```
export LD_LIBRARY_PATH=/usr/lib:/usr/local/lib
```

```
export LIBRARY_PATH=/usr/lib:/usr/local/lib
```

```
export C_INCLUDE_PATH=./usr/include:/usr/local/include
```

```
export CPATH=./usr/include:/usr/local/include
```

But, your's may vary accordingly. You won't be able to cut & paste mine, unless your system is built exactly like mine. It's up to you to locate exactly where all the libs and includes are located. That is what all the previous commands should have helped you do. Just because you have /usr/lib & /usr/local/lib included....doesn't mean your needed lib is in that path.

That is where your detective work comes to play. SEARCH and use grep to locate the libs.

Once you have the env set either reboot or reset the env. Once again, your system command for this can/may be different.

4. source .bashrc -- Reset the environment to what we need for Compiles, assuming the SHELL is bash. This may not be available on your system.....

REF:

http://linux.about.com/library/cmd/blcmdl1_ulimit.htm

```
source filename [arguments]
```

Read and execute commands from filename in the current shell environment and return the exit status of the last command executed from filename. v If filename does not contain a slash, file names in PATH are used to find the directory containing filename. vThe file searched for in PATH need not be executable. vWhen bash is not in posix mode, the current directory is searched if no file is found in PATH. vIf the sourcepath option to the shopt builtin command is turned off, the PATH is not searched. vIf any arguments are supplied, they become the positional parameters when filename is executed. vOtherwise the positional parameters are unchanged. vThe return status is the status of the last command exited within the script (0 if no commands are executed), and false if filename is not found or cannot be read.

VERIFY the env is correct:

5. env -- Display the new environment to verify the paths.....PARTIAL DISPLAY of SPECIFIC's I NEEDED...

```
LIBRARY_PATH=/usr/lib:/usr/local/lib
```

```
LD_LIBRARY_PATH=/usr/lib:/usr/local/lib
```

```
CPATH=./usr/include:/usr/local/include
```

```
C_INCLUDE_PATH=./usr/include:/usr/local/include
```

Append the new CP/M Floppy Disk Definitions for cpmtools to the /usr/local/share/diskdefs file.

Append these sample diskdefs to /usr/local/share/diskdefs & /etc/cpmtools/diskdefs

Additional Definitions for diskdefs:

```
diskdef amp1
  seclen 512      #= Sectors xx,512
  tracks 40       #= (Cylinders * Sides) = 40*1 = 40
  sectrk 10       #= Sectors 10,xxx
  blocksize 2048  #= (128*(BLM+1)) = 2048
  maxdir 64       #= (DRM+1) = 64
```

```
    skew 0          # = SKEW 0
    boottrk 2       # = OFS = 2
    os 2.2
end
```

```
# setfdprm /dev/fd0 DS DD ssize=512 cyl=40 sect=10 head=2
diskdef amp2
    seclen 512
    tracks 80
    sectrk 10
    blocksize 2048
    maxdir 128
    skew 0
    boottrk 2
    os 2.2
end
```

```
# setfdprm /dev/fd0 zerobased SS DD ssize=512 cyl=40 sect=10 head=1
# Kaypro II
diskdef kpii
    seclen 512
    tracks 40
    sectrk 10
    blocksize 1024
    maxdir 64
    skew 0
    boottrk 1
    os 2.2
end
```

```
# setfdprm /dev/fd0 zerobased DS DD ssize=512 cyl=40 sect=10 head=2
# Kayro IV
diskdef kpiv
    seclen 512
    tracks 80
    sectrk 10
    blocksize 2048
    maxdir 64
    skew 0
    boottrk 1
    os 2.2
end
```

```
diskdef zor1
    seclen 512
    tracks 80
    sectrk 10
    blocksize 2048
    maxdir 64
    skew 0
    boottrk 2
    os 2.2
end
```

```
diskdef zor2
    seclen 512
    tracks 160
```

```

sectrk 10
blocksize 4096
maxdir 128
skew 0
boottrk 2
os 2.2
end

```

UPDATE: 02/17/2023 CP/M Floppy Definitions

The latest CP/M Floppy definitions are now available on github.

<https://github.com/ldkraemer/CPM-Floppy-Definitions>

UPDATE: 08/13/2014 cpmtools Version 2.19 has been released with additional disk definitions

I've attached my definition file because I've added (and tested) several additional definitions.

CORRECTED SKEW=2 in Radio Shack Model 4 CP/M Definitions.

diskdefs.zip

diskdefs.zip

(6.67 KiB) Downloaded 1515 times

Copy these sample libdiskrc definitions to an External file at /home/user/.libdiskrc

.libdiskrc contains:

[amp1]

description = Ampro - SSDD 48 tpi 5.25"

cylinders = 40

heads = 1

sectors = 10

secbase = 1

seccsize = 512

datarate = DD

rwgap = 12

fmtgap = 23

fm = N

multitrack = N

skipdeleted = Y

[amp2]

description = Ampro - DSDD 48 tpi 5.25"

sides=alt

cylinders = 40

heads = 2

sectors = 10

secbase = 17

seccsize = 512

datarate = DD

rwgap = 12

fmtgap = 23

fm = N

multitrack = N

skipdeleted = Y

[kpii]

description = Kaypro II/2 - SSDD 48 tpi 5.25"

cylinders = 40

heads = 1

```
sectors = 10
secbase = 0
secsize = 512
datarate = DD
rwgap = 12
fmtgap = 23
fm = N
multitrack = N
skipdeleted = Y
```

```
[kpiv]
description = Kaypro 2X/4/10 - DSDD 48 tpi 5.25"
cylinders = 80
heads = 2
sectors = 10
secbase = 0
secsize = 512
datarate = DD
rwgap = 12
fmtgap = 23
fm = N
multitrack = N
skipdeleted = Y
```

```
[zor1]
description = Zorba - DSDD 48 tpi 5.25"
sides=alt
cylinders = 80
heads = 2
sectors = 10
secbase = 1
secsize = 512
datarate = DD
rwgap = 12
fmtgap = 23
fm = N
multitrack = N
skipdeleted = Y
```

```
[zor2]
description = Zorba - DSDD 48 tpi 5.25"
sides=alt
cylinders = 160
heads = 2
sectors = 10
secbase = 1
secsize = 512
datarate = DD
rwgap = 12
fmtgap = 23
fm = N
multitrack = N
skipdeleted = Y
sudo cp diskdefs /usr/local/share/diskdefs
sudo cp diskdefs /etc/cpmtools/diskdefs
cp .libdiskrc /home/user/.libdiskrc
```

TYPICAL COMPILE STEPS: (from within your source directory)

```
./configure
make clean
make
sudo make install
```

"make clean" won't remove anything on the first compile, but will clean up on a successive compile.

Build libdsk:

```
cd ~/Downloads/cpmtools
cd libdsk-1.3.5
./configure
make
sudo make install
sudo make install-man
```

Build cpmtools:

```
cd ..
cd cpmtools-2.18
./configure --with-libdsk=/usr/local
make
sudo make install
```

VERIFY cpmls executes properly:

```
cd ..
cpmls
```

If you don't get the following cli usage message:

Usage: cpmls [-f format] [-T libdsk-type] [-d|-D|-F|-A|[-l|][-c|][-i]] image [file ...]

try this command:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH: "/usr/local/lib"
```

Then do:

```
cpmls
```

Open a Teledisk *.TD0 file containing compressed CP/M Files, and view the CP/M Directory with the following commands:

```
man cpmls
```

```
cpmls -f kpii -T tele,kpii -d kii-b282.td0
```

```
cpmls -f kpiv -T tele,kpiv -l KAY2SYS.td0
```

```
cpmls -f amp1 -T tele,amp1 -i ampro.td0
```

```
cpmls -f zor1 -T tele,zor1 -l compat32.td0
```

```
cpmls -f zor1 -T tele,zor1 -d compat32.td0
```

```
man cpmcp
```

```
cpmcp -f a1 bb2mon.img 0:*.mac /home/larry/Downloads/cpmtools/xfer1
```

```
larry@debian:~/Downloads/cpmtools$ cpmls -f zor1 -T tele,zor1 -l compat32.td0
0:
-rwxrwxrwx 19072 Dec 31 1969 compat.com
-rwxrwxrwx 22784 Dec 31 1969 format.com
-rwxrwxrwx 4352 Dec 31 1969 install.com
-rwxrwxrwx 11008 Dec 31 1969 ms.com
-rwxrwxrwx 9472 Dec 31 1969 mycopy.com
-rw-rw-rw- 1408 Dec 31 1969 z2000q.hex
-rw-rw-rw- 4480 Dec 31 1969 z2000q.prl
-rw-rw-rw- 1280 Dec 31 1969 zorbad.hex
-rw-rw-rw- 4224 Dec 31 1969 zorbad.prl
-rw-rw-rw- 1280 Dec 31 1969 zorbaq.hex
```

```
-rw-rw-rw- 4352 Dec 31 1969 zorbaq.pri
larry@debian:~/Downloads/cpmtools$
```

```
larry@debian:~/Downloads/cpmtools$ cpmls -f zor1 -T tele,zor1 -d
compat32.td0
COMPAT COM : FORMAT COM : MS COM : INSTALL COM
MYCOPY COM : ZORBAD PRL : ZORBAD HEX : ZORBAQ PRL
ZORBAQ HEX : Z2000Q PRL : Z2000Q HEX
larry@debian:~/Downloads/cpmtools$
```

THANKS to John Elliott (libdisk) & Michael Haardt (cpmtools) for developing each software package.

PM me with a valid email address if you would like a CP/M *.TD0 file to test.

Larry

Re: [HOWTO] - Access CP/M Floppy's via libdisk & cpmtools

by **Ikraemer**

Posted: **2014-03-11 00:02**

Now that you need to modify or add CP/M definitions to 22DISK, LIBDSK, and/or CPMTOOLS, this document will assist you with locating the proper parameters for the three definition files for:

```
22DISK - CPMDISKS.DEF
LIBDSK - .libdiskrc
CPMTOOLS - diskdefs
~/path/to/your/22Disk/CPMDISKS.DEF
/home/user/.libdiskrc
/usr/local/share/diskdefs (also located at /etc/cpmtools/diskdefs)
```

Some Helpful Hints on Developing a Disk Definition from Scratch

If possible, run programs on the CP/M system to help get the necessary information for building a new definition. The following files may be transferred by RS-232C if a Terminal Program is available, or by PIPMODEM.COM (PIPMODEM.DOC & PIPMODEM.ASM)

1. CPMADR.COM - Finds addresses used by the CP/M system.
2. DPB.COM - Finds Disk Parameter Block information.
DPB.BAS - Same, but runs under MBASIC. (Located in 22DISK.TXT)
3. PROBE.COM - Probes CP/M System and Drive for information.
Note: The previous four programs may not locate and display accurate information for some CP/M systems.
4. SKEW.COM - Finds disk SKEW information.
5. Execute the following "STAT" commands on system in question.
Note: "Control P" will allow a hardcopy of the information.

STAT DEV: - Displays current DEVICE information.

STAT VAL: - Displays current STATUS information.

STAT USR: – Displays current USR information.
 STAT A*.* \$S – Displays detailed file information.
 STAT DSK: – Gives some detailed disk information for logged drive.
 STAT A: DSK: – Gives some detailed disk information for A:.
 STAT B: DSK: – Gives some detailed disk information for B:.

If you have a functional CP/M system, you can use DDT to find the Disk Parameter Block and the information is stored there for each drive.

1. Log the drive you wish to locate the DPB data for. I will use A; for this example
2. Run DDT and enter the code below:

```

DDT
-f0100,0200,00
-A0100
0100 mvi c,1f
0102 call 5
0105 rst 7
0106 . #period to exit assembler
-d0100
0100 0e1fcd0500FF
-g0100
*105
-x #display register HL
  
```

The contents of register HL is the address of Drive A: DPB. If you display that address (mine was F5E3) you will see F5E3: 2400040F0154007F00C00020000200.

RPT = 0024 = The number of 128 byte records per track.
 BSH = 04 = The block shift count.
 BLM = 0F = The block mask.
 EXM = 01 = The extent mask.
 DSM = 0054 = Disk storage maximum (the largest block number).
 DRM = 007F = Directory maximum (the largest directory entry).
 DAB = 00C0 = Directory Allocation Block AL0: = 00 & AL1: = C0
 CKS = 00020 = Directory check size.
 OFF = 0002 = Track offset (number of reserved tracks).

SPT: Number of sectors per track. May differ from RPT if physical sectors are other than 128 bytes in size.

SSZ: Sector size code:
 0 = 128 bytes per sector
 1 = 256 bytes per sector
 2 = 512 bytes per sector
 3 = 1024 bytes per sector

NTK: Number of tracks (35, 40, 77, or 80). The number of tracks can be determined by formatting a Floppy, or with Anadisk. Anadisk will also tell you the sector size and SKEW.

Quite frequently, the only information available when developing a definition for 22DISK is a

diskette with a few files and nothing more.

Here's a guide on what to do if you'd like to try working up a definition yourself:

1. Get a copy of Sydex's ANADISK. This utility will furnish far more information than will any other program. A typical disk definition follows:

```
Ampro Little Board Z80 running CP/M 2.21
BEGIN AMP1 Ampro - SSDD 48 tpi 5.25"
DENSITY MFM, LOW
CYLINDERS 40 SIDES 1 SECTORS 10,512 SKEW 2
SIDE1 0 1,2,3,4,5,6,7,8,9,10
BSH 4 BLM 15 EXM 1 DSM 94 DRM 63 AL0 080H AL1 0 OFS 2
END
```

New 22DISK definitions require the following information:

```
DENSITY xx,xx
CYLINDERS xx
SIDES xx
SECTORS x,xxxx
SKEW x
SIDE1 0 x,x1,x2,x3, , ,xn
SIDE2 x x,x1,x2,x3, , ,xn
ORDER x
BSH x
BLM x
EXM x
DSM x
DRM x
AL0 x
AL1 x
OFS x (or SOFS x)
COMPLEMENT (unlikely)
```

New LIBDSK definitions require the following information:

[title]
description = DESC The description of the format as shown by (for example) dskform-help.
sidedness =TREATMENT How a double-sided disk is handled. This can either be alt (sides alternate - used by most PC-hosted operating systems),
outback (use side 0 tracks 0-79, then side 1 tracks 79-0 - used by 144FEAT CP/M disks), or outout (use side 0 tracks 0-79, then side 1 tracks 0-79 - used by some Acorn formats). If the disk is single-sided, this parameter can be omitted.
cylinders = COUNT Sets the number of cylinders (usually 40 or 80).
heads = COUNT Sets the number of heads (usually 1 or 2 for single- or double- sided).
sectors = COUNT Sets the number of sectors per track.
secbase = NUMBER Sets the first sector number on a track. Usually 1; some Acorn formats use 0.
seclsize = COUNT Sets the size of a sector in bytes. This should be a power of 2.
datarate = VALUE Sets the rate at which the disk should be accessed. This is one of: HD, DD, SD or ED.
rwgap = VALUE Sets the read/write gap.

rmtgap = VALUE Sets the format gap.
 fm = Y or N Sets the recording mode – Y for FM, N for MFM.
 multitrack = Y or N Sets multitrack mode.
 skipdeleted = Y or N Sets whether to skip deleted data.

The LIBDSK Data rate will be one of:

RATE_HD, /* Rate for High-density disc – 1.2Mb in 5.25" 96 tpi drive, or 1.44Mb in 3.5" 96 tpi drive */
 RATE_DD, /* Rate for Double-density disc – 360k in 5.25" 48 tpi drive, or 720K in 3.5" 48 tpi drive */
 RATE_SD, /* Rate for Double-density disc – 180k in 5.25" 48 tpi drive, or 360k in 3.5" 48 tpi drive */
 RATE_ED /* Data rate for 2.8Mb 3.5" in 3.5" 96 tpi drive */

New CPMTOOLS definitions require the following information:

```

diskdef title
seclen xxx #= Sectors xx,1024
tracks xx #= (Cylinders * Sides) = 80*2 = 160
sectrk xx #= Sectors 5,xxx
blocksize xxxx #= (128*(BLM+1)) = 2048
maxdir xxx #= (DRM+1) = 256
skew x #= may be 1 thru 6, or so
boottrk x #= OFS = 2
os x.x #= 2.2, or 2, or 3
end
  
```

So, if you know the 22DISK parameters, you can easily fill in the details for LIBDSK & CPMTOOLS. As an example, here is an AMPRO LITTLE BOARD DSDD 96 tpi 5.25" Definition for all three software packages.

```

BEGIN AMP4 Ampro - DSDD 96 tpi 5.25"
DENSITY MFM,LOW
CYLINDERS 80 SIDES 2 SECTORS 5,1024 SKEW 2
SIDE1 0 17,18,19,20,21
SIDE2 1 17,18,19,20,21
ORDER SIDES
BSH 4 BLM 15 EXM 0 DSM 394 DRM 255 AL0 0F0H AL1 0 OFS 2
END
  
```

```

[amp4]
description = Ampro - DSDD 96 tpi 5.25"
sides=alt #=/* XXX Provisional depending on 22DISK Definition */
cylinders = 80 #= Cylinders = 80
heads = 2 #= Sides = 2
sectors = 5 #= Sectors 5,xxx
secbase = 17 #= First Sectors from (SIDE1 0 17)
secsize = 1024 #= Sectors xx,1024
datarate = DD #= Data Rate = SD, DD, HD, ED
rwgap = 12 #=/* XXX Provisional */
fmtgap = 23 #=/* XXX Provisional */
fm = N #= FM or MFM (Likely MFM)
multitrack = N #= Most likely N
skipdeleted = Y #= Most likely Y
  
```

```

diskdef AMP4
seclen 1024 #= Sectors xx,1024
tracks 160 #= (Cylinders * Sides) = 80*2 = 160
sectrk 5 #= Sectors 5,xxx
blocksize 2048 #= (128*(BLM+1)) = 2048
maxdir 256 #= (DRM+1) = 256
skew 0 #= may be 1 thru 6, or so
boottrk 2 #= OFS = 2
os 2.2 #= 2.2, or 2, or 3
end

```

2. Run ANADISK on the diskette in SECTOR EDIT mode with the display set for "HEX". Note that the interleave (or SKEW) is given, as well as the number, size and addresses of

sectors. You now have data for: DENSITY, CYLINDERS, SIDES, SKEW SIDE1 and SIDE2.

3. Locate the directory on the diskette. Almost invariably, it starts on the first sector of a track.

If for some reason it can't be found on the first sector of a track, try using the SEARCH feature

to find a "FILE" name that exists on the disk. The number of tracks or sectors up the point

where the directory begins will give the figure for OFS or SOFS. The directory stands out

because it shows file names every 32 bytes, or every other line on the ANADISK display.

Directory was located at Cylinder 0, Side 0, Sector 2:

```

0000 00 32 32 44 49 53 4b 20 20 44 4f 43 00 00 00 80 .22DISK DOC....
0010 02 00 03 00 04 00 05 00 06 00 07 00 08 00 09 00 .....
0020 00 32 32 44 49 53 4b 20 20 44 4f 43 01 00 00 80 .22DISK DOC....
0030 0a 00 0b 00 0c 00 0d 00 0e 00 0f 00 10 00 11 00 .....
0040 00 32 32 44 49 53 4b 20 20 44 4f 43 02 00 00 80 .22DISK DOC....
0050 12 00 13 00 14 00 15 00 16 00 17 00 18 00 19 00 .....
0060 00 32 32 44 49 53 4b 20 20 44 4f 43 03 00 00 80 .22DISK DOC....
0070 1a 00 1b 00 1c 00 1d 00 1e 00 1f 00 20 00 21 00 ..... !.
0080 00 32 32 44 49 53 4b 20 20 44 4f 43 04 00 00 08 .22DISK DOC....
0090 22 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ".....
00a0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00b0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00c0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00d0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00e0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00f0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0100 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0110 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0120 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0130 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0140 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0150 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0160 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0170 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0180 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0190 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01a0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01b0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01c0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....

```

```

0100 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01d0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01e0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01f0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....

```

Fig 1

Directory continues on Cylinder 0, Side 0, Sector 3:

```

0000 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0010 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0020 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0030 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0040 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0050 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0060 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0070 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0080 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0090 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00a0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00b0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00c0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00d0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00e0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00f0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0100 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0110 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0120 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0130 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0140 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0150 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0160 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0170 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0180 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0190 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01a0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01b0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01c0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01d0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01e0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01f0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....

```

Fig 2

4. Examine the first directory entry. The second 16 bytes (the second line of the entry) gives a list of allocation unit numbers (AUN) belonging to the file. The first one of these will tell how many allocation units have been reserved to hold the directory. Thus, if the first allocation byte is 02, two allocation units have been allocated to the directory. Another thing to note is whether the allocation unit number appears to be an 8 bit or 16 bit quantity. For example, if an entry has the following form:

```

- - - - - 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

```
00 50 49 50 20 20 20 20 20 43 4f 4d 00 00 00 10 .PIP COM....
04 00 05 00 06 00 00 00 00 00 00 00 00 00 00 00 .....
```

the allocation unit designators are 4, 5 and 6 and are 16 bit quantities. On the other hand, the following item:

```
00 50 49 50 20 20 20 20 20 43 4f 4d 00 00 00 10 .PIP COM....
04 05 06 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

shows the same allocation units, but as 8 bit quantities. If 16 bit quantities are involved, more than 255 allocation units are defined for the diskette, if an 8 bit quantity, less than 256 units.

This serves as a check for later computations. In any case, the first allocation number of the first directory entry gives the number of bits to be set in AL0 and AL1.

Allocation Unit Number (AUN)

```
02 AL0 11000000B AL1 0
04 AL0 11110000B AL1 0
```

The following information shows a detailed breakdown of all the information in the the first and second directory entries.

```
00 - Valid file (E5 for Erased File)
32 32 44 49 53 4b 20 20 44 4f 43 - File name 22DISK DOC (11 bytes)
00 - Extent (used for large files)
00 - S1 (Digital Research Reserved)
00 - S2 (Digital Research Reserved)
80 - Number of records in this extent
02 00 03 00 04 00 05 00 06 00 07 00 08 00 09 00 - allocation units
```

Note: 02 is the first AUN.

```
00 - (Continuation of above file)
32 32 44 49 53 4b 20 20 44 4f 43 - File name 22DISK DOC (11 bytes)
01 - Extent (used for large files)
00 - S1 (Digital Research Reserved)
00 - S2 (Digital Research Reserved)
80 - Number of records used
0a 00 0b 00 0c 00 0d 00 0e 00 0f 00 10 00 11 00 - allocation units
```

Note: Unused allocation units will be 00.

5. Starting with the first directory sector, advance the sector (and track, if necessary) until you find the data belonging to the first file. Unused directory entries are almost always filled out with a pattern of E5's. The number of sectors dedicated to the directory will give an indication of both the number of directory entries and the allocation unit size.

First Program located at Cylinder 0, Side 1, Sector 2:

NOTE: (This was the 9th sector displayed, so 8 are reserved for directory)

```

0000 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 20 20 20 20 .....
0010 20 20 53 79 64 65 78 0d 0a 0d 0a 0d 0a 0d 0a 0d Sydex.....
0020 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d .....
0030 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 20 .....
0040 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0050 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0060 20 20 20 20 20 32 32 44 49 53 4b 0d 0a 0d 0a 0d 22DISK.....
0070 0a 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 .
0080 20 20 20 20 20 41 20 43 50 2f 4d 2d 74 6f 2d 44 A CP/M-to-D
0090 4f 53 20 44 69 73 6b 65 74 74 65 20 49 6e 74 65 OS Diskette Inte
00a0 72 63 68 61 6e 67 65 20 55 74 69 6c 69 74 79 0d rchange Utility.
00b0 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d .....
00c0 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d .....
00d0 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 20 .....
00e0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00f0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0100 20 20 20 20 20 20 20 20 20 20 20 20 20 20 53 79 64 Syd
0110 65 78 0d 0a 20 20 20 20 20 20 20 20 20 20 20 20 20 ex..
0120 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0130 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0140 20 20 31 35 33 20 4e 6f 72 74 68 20 4d 75 72 70 153 North Murp
0150 68 79 20 41 76 65 2e 0d 0a 20 20 20 20 20 20 20 20 hy Ave...
0160 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0170 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0180 20 20 20 20 20 20 20 20 53 75 6e 6e 79 76 61 6c 65 Sunnyvale
0190 2c 20 43 41 20 20 39 34 30 38 36 0d 0a 20 20 20 , CA 94086..
01a0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
01b0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
01c0 20 20 20 20 20 20 20 20 20 20 20 20 28 34 30 38 29 (408)
01d0 20 37 33 39 2d 34 38 36 36 0d 0a 0d 0a 0d 0a 0d 739-4866.....
01e0 0a 0c 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a .....
01f0 0d 0a 0d 0a 0d 0a 0d 0a 20 20 20 20 20 20 20 20 .....

```

Fig 3

Suppose that a our hypothetical diskette shows 8 sectors of 512 bytes allocated to the directory.

Since a directory entry is 32 bytes long, the number of directory entries possible for this format is:

16 entries/sector X 8 sectors = 128 entries total
 or
 ((Sector Size in Bytes/32) X Sectors to DIR) -1 = DRM

Thus, DRM is one less than this or 127. Since we know that these 8 sectors represent 2 allocation units or 4,096 bytes, the Allocation Unit Size (AUS) is:

4,096 / 2 = 2,048 bytes
 or
 ((Sector Size in Bytes) X Sectors to DIR) / AUN = AUS

Given this, we can use the following table to compute the values for BSH and BLM:

Unit Size	BSH	BLM	
1,024	3	7	+-
2,048	4	15	-----+ T
4,096	5	31	+-
8,192	6	63	
16,384	7	127	

7. We still need a value for EXM and DSM. DSM is simply the total number of allocation units

on the diskette, not counting the system OFS area. Let's assume that our diskette has an offset

of 2 tracks and is single-sided with 40 cylinders with 8 sectors per track. The value of DSM

is then:

$((40 \text{ cylinders} \times 1 \text{ side}) - 2 \text{ tracks}) \times 8 \text{ sectors} \times 512 \text{ bytes} = 155,648 \text{ bytes} / 2,048 = 76 \text{ Allocation units, total}$

To get EXM, we then use the following table:

Unit Size	DSM < 256	DSM > 255
1,024	0	na
2,048	1	0
4,096	3	1
8,192	7	3
16,384	15	7

8. That's almost the complete picture, but for the SIDE1 and SIDE2 sector ordering. This is the

hardest to determine of any of the parameters. Find a readable text file on the diskette and

piece the text back together by observing how words break across sector boundaries.

After a

bit, a pattern in the ordering will become apparent, such as:

1, 3, 5, 7, 2, 4, 6, 8

9. The ORDER value can be determined on double-sided diskettes by observing whether file

data appears to move from one track to the next on the same side (CYLINDERS), or appears to

go from one side to the other and then advance to the next cylinder (SIDES). It should be

noted here that there are several different CP/M Format layouts possible. For ORDER, there

are five different options in 22DISK: SIDES, CYLINDERS, EAGLE, COLUMBIA, or EVEN-ODD

Here are the 22DISK choices for those Parameters:

ORDER SIDES specifies that first one side then the other is written/read before the cylinder number is advanced.

ORDER CYLINDERS specifies that all tracks on the first side are recorded then all tracks on second side. Tracks on side 0 are written from 0 up and side 1 down to 0.

ORDER EAGLE same as CYLINDERS except both sides start at zero or outermost track.

ORDER COLUMBIA same as CYLINDERS except tracks are sequential from side 0 to side 1 ie. 0 – 79 for a 40 track disk.

ORDER EVEN-ODD means side 0 contains even-numbered tracks and side 1 contains odd-numbered tracks.

Here are the LIBDISK choices for those Parameters:

```
SIDES_ALT, /* Track n is cylinder C C0H0,C0H1,C1H0,C1H1,.....*/
SIDES_OUTBACK, /* Tracks go (head 0) 0,1,2,3,...37,38,39, then
(head 1) 39,38,37,...,2,1,0 */
SIDES_OUTOUT, /* Tracks go (head 0) 0,1,2,3,...37,38,39, then
(head 1) 0,1,2,3,...37,38,39 */
```

10. COMPLEMENT is rare. Generally, if a diskette appears to be formatted with values of 1A instead of E5, the data on the diskette is complemented.

11. Last resort for information will require a memory dump from F200 hex through FFFF to provide all the DBP information as hex bytes. It will be necessary to dig through 3584 (E00) bytes of data to locate the DPB information. Most systems follow the layout shown below.

(Hex values starting in memory at Fxxx)
RPT: (word) total # of logical 128-byte sectors/track
BSH: (byte) data allocation block shift factor,
determined by data block allocation size
1k=3 2k=4 4k=5 8k=6 16k=7
BLM: (byte) data allocation block mask ($2^{BSH}-1$)
1k=7 2k=15 4k=31 8k=63 16k=127
EXM: (byte) extent mask, determined by the data block
allocation size and the number of disk blocks
1k=0 2k=1 4k=3 8k=7 16k=15 if DSM < 256
1k=n 2k=0 4k=1 8k=3 16k=7 if DSM > 256
DSM: (word) disk size in blocks-1; determines the total
storage capacity of the disk size
DRM: (word) directory size-1; determines the total number
directory entries that can be stored on this
drive. DRM: = ((# dir entries) / 4) - 1
AL0: (byte) alloc0 (1100 0000) 2 blocks/dir
(1000 0000) 1 block/dir
AL1: (byte) alloc1 (0000 0000)

Note that AL0: and AL1: are taken together as a single sixteen-bit mask for the number of blocks of directory allocation.

(1100 0000 0000 0000) 2 blocks/dir
(1000 0000 0000 0000) 1 block/dir

CKS: (word) directory check vector size

fixed media: $CKS = 0$
removable media: $CKS = (DRM + 1)/4$

OFF/SOFF: (word) # of reserved system tracks or sectors (offset) at the beginning of the (logical) disk.

Usually the directory begins at the first sector of the track defined by this number.

By formatting a disk the number of tracks can be determined, which will give the first byte to look for. The SPT will be followed by 00 and the value for BSH and BLM will be the third and fourth bytes.

BSH and BLM will be one of the following Hex values:

BSH 3 or 4 or 5 or 6 or 7
BLM 7 or 0F or 1F or 3F or 7F

The above information should be fairly easily to locate.

Larry

Re: [HOWTO] - Access CP/M Floppy's via libdisk & cpmtools

by Ikraemer

Posted: **2014-03-11 01:18**

For those folks still using their Radio Shack Model 4 or 4P, I've attached the Montezuma Micro CP/M Version 2.31 Disk Definitions I created (and submitted) for 22DISK. I've also updated the Definitions with those for LIBDSK & CPMTOOLS (03/11/2014).

The ones marked as #NOT TESTED were not verified, as I don't have a 96 tpi 5.25" Floppy Drive.

```
#22DISK CP/M Disk Definitions
BEGIN TRSG TRS-80 Model 4,4P Montezuma System 170K - SSDD 48 tpi 5.25"
DENSITY MFM, LOW
CYLINDERS 40 SIDES 1 SECTORS 18,256
SIDE1 0 1,3,5,7,9,11,13,15,17,2,4,6,8,10,12,14,16,18
BSH 4 BLM 15 EXM 1 DSM 84 DRM 127 AL0 0C0H AL1 0 OFS 2
END
```

```
#CPMTOOLS CP/M Disk Definitions
#diskdef trsg
# seclen 256
# tracks 40
# sectrk 18
# blocksize 2048
# maxdir 128
# ...
```

```
# datarate DD
# fm NO
# skew 2
# boottrk 2
# os 2.2
#end
```

#LIBDSK CP/M Disk Definitions

```
#[trsg]
#description = TRS-80 Model 4,4P Montezuma System 170K - SSDD 48 tpi 5.25"
#cylinders = 40
#heads = 1
#sectors = 18
#secbase = 1
#secsize = 256
#datarate = DD
#rwgap = 12
#fmtgap = 23
#fm = N
#multitrack = N
#skipdeleted = Y
```

```
BEGIN TRSH TRS-80 Model 4,4P Montezuma Data 200K - SSDD 48 tpi 5.25"
DENSITY MFM, LOW
CYLINDERS 40 SIDES 1 SECTORS 10,512
SIDE1 0 1,3,5,7,9,2,4,6,8,10
BSH 4 BLM 15 EXM 1 DSM 99 DRM 127 AL0 0C0H AL1 0 OFS 0
END
```

```
#diskdef trsh
# seclen 512
# tracks 40
# sectrk 10
# blocksize 2048
# maxdir 128
# datarate DD
# fm NO
# skew 2
# boottrk 0
# os 2.2
#end
```

```
#[trsh]
#description = TRS-80 Model 4,4P Montezuma Data 200K - SSDD 48 tpi 5.25"
#sides=alt
#cylinders = 40
#heads = 1
#sectors = 10
#secbase = 1
#secsize = 512
#datarate = DD
#rwgap = 12
#fmtgap = 23
#fm = N
#multitrack = N
#skipdeleted = Y
```

```
BEGIN TRSI TRS-80 Model 4,4P Montezuma System 350K - DSDD 48 tpi 5.25"
DENSITY MFM, LOW
CYLINDERS 40 SIDES 2 SECTORS 18,256
SIDE1 0 1,3,5,7,9,11,13,15,17,2,4,6,8,10,12,14,16,18
SIDE2 1 1,3,5,7,9,11,13,15,17,2,4,6,8,10,12,14,16,18
ORDER EAGLE
BSH 4 BLM 15 EXM 1 DSM 174 DRM 127 AL0 0C0H AL1 0 OFS 2
END
```

```
#diskdef trsi
# seclen 256
# tracks 80
# sides outout
# sectrk 18
# blocksize 2048
# maxdir 128
# datarate DD
# fm NO
# skew 2
# boottrk 2
# os 2.2
#end
```

```
#[trsi]
#description = TRS-80 Model 4,4P Montezuma System 350K - DSDD 48 tpi 5.25"
#sides=alt
#cylinders = 40
#heads = 2
#sectors = 18
#secbase = 1
#secsize = 256
#datarate = DD
#rwwgap = 12
#fmtgap = 23
#fm = N
#multitrack = N
#skipdeleted = Y
```

```
BEGIN TRSJ TRS-80 Model 4,4P Montezuma Data 400K - DSDD 48 tpi 5.25"
DENSITY MFM, LOW
CYLINDERS 40 SIDES 2 SECTORS 10,512
SIDE1 0 1,3,5,7,9,2,4,6,8,10
SIDE2 1 1,3,5,7,9,2,4,6,8,10
ORDER EAGLE
BSH 4 BLM 15 EXM 1 DSM 199 DRM 127 AL0 0C0H AL1 0 OFS 0
END
```

```
#diskdef trsj
# seclen 512
# tracks 80
# sides outout
# sectrk 10
# blocksize 2048
# maxdir 128
# datarate DD
# fm NO
# skew 2
```

```

# skew 2
# boottrk 0
# os 2.2
#end

#[trs]
#description = TRS-80 Model 4,4P Montezuma Data 400K - DSDD 48 tpi 5.25"
#sides=outout
#cylinders = 40
#heads = 2
#sectors = 10
#secbase = 1
#secsize = 512
#datarate = DD
#rwwgap = 12
#fmtgap = 23
#fm = N
#multitrack = N
#skipdeleted = Y

#NOT TESTED
BEGIN TRSK TRS-80 Model 4,4P Montezuma System 350K - SSDD 96 tpi 5.25"
DENSITY MFM, LOW
CYLINDERS 80 SIDES 1 SECTORS 18,256
SIDE1 0 1,3,5,7,9,11,13,15,17,2,4,6,8,10,12,14,16,18
BSH 4 BLM 15 EXM 1 DSM 174 DRM 127 AL0 0C0H AL1 0 OFS 2
END

#diskdef trsk
# seclen 256
# tracks 80
# sectrk 18
# blocksize 2048
# maxdir 128
# datarate DD
# fm NO
# skew 2
# boottrk 2
# os 2.2
#end

#[trsk]
#description = TRS-80 Model 4,4P Montezuma System 350K - SSDD 96 tpi 5.25"
#cylinders = 80
#heads = 1
#sectors = 18
#secbase = 1
#secsize = 256
#datarate = DD
#rwwgap = 12
#fmtgap = 23
#fm = N
#multitrack = N
#skipdeleted = Y

#NOT TESTED
BEGIN TRSL TRS-80 Model 4,4P Montezuma Data 400K - SSDD 96 tpi 5.25"

```

```
DENSITY MFM, LOW
CYLINDERS 80 SIDES 1 SECTORS 10,512
SIDE1 0 1,3,5,7,9,2,4,6,8,10
BSH 4 BLM 15 EXM 1 DSM 199 DRM 127 AL0 0C0H AL1 0 OFS 0
END
```

```
#diskdef trsl
# seclen 512
# tracks 80
# sectrk 10
# blocksize 2048
# maxdir 128
# datarate DD
# fm NO
# skew 2
# boottrk 0
# os 2.2
#end
```

```
#[trsl]
#description = TRS-80 Model 4,4P Montezuma Data 400K - SSDD 96 tpi 5.25"
#cylinders = 80
#heads = 1
#sectors = 10
#secbase = 1
#secsize = 512
#datarate = DD
#rwwgap = 12
#fmtgap = 23
#fm = N
#multitrack = N
#skipdeleted = Y
```

```
#NOT TESTED
BEGIN TRSM TRS-80 Model 4,4P Montezuma System 710K - DSDD 96 tpi 5.25"
DENSITY MFM, LOW
CYLINDERS 80 SIDES 2 SECTORS 18,256
SIDE1 0 1,3,5,7,9,11,13,15,17,2,4,6,8,10,12,14,16,18
SIDE2 1 1,3,5,7,9,11,13,15,17,2,4,6,8,10,12,14,16,18
ORDER SIDES
BSH 4 BLM 15 EXM 0 DSM 354 DRM 127 AL0 0C0H AL1 0 OFS 2
END
```

```
#diskdef trsm
# seclen 256
# tracks 160
# sides alt
# sectrk 18
# blocksize 2048
# maxdir 128
# datarate DD
# fm NO
# skew 2
# boottrk 2
# os 2.2
#end
```

```
#[trsm]
#description = TRS-80 Model 4,4P Montezuma System 710K - DSDD 96 tpi 5.25"
#sides=alt
#cylinders = 80
#heads = 2
#sectors = 18
#secbase = 1
#secsize = 256
#datarate = DD
#rwwgap = 12
#fmtgap = 23
#fm = N
#multitrack = N
#skipdeleted = Y
```

```
#NOT TESTED
BEGIN TRSN TRS-80 Model 4,4P Montezuma Data 800K - DSDD 96 tpi 5.25"
DENSITY MFM, LOW
CYLINDERS 80 SIDES 2 SECTORS 10,512
SIDE1 0 1,3,5,7,9,2,4,6,8,10
SIDE2 1 1,3,5,7,9,2,4,6,8,10
ORDER SIDES
BSH 4 BLM 15 EXM 0 DSM 399 DRM 127 AL0 0C0H AL1 0 OFS 0
END
```

```
#diskdef trsn
# seclen 512
# tracks 160
# sides alt
# sectrk 10
# blocksize 2048
# maxdir 128
# datarate DD
# fm NO
# skew 2
# boottrk 0
# os 2.2
#end
```

```
#[trsn]
#description = TRS-80 Model 4,4P Montezuma Data 800K - DSDD 96 tpi 5.25"
#sides=alt
#cylinders = 80
#heads = 2
#sectors = 10
#secbase = 1
#secsize = 512
#datarate = DD
#rwwgap = 12
#fmtgap = 23
#fm = N
#multitrack = N
#skipdeleted = Y
```

```
BEGIN TRSO TRS-80 Model 4,4P Montezuma Extend System 190K - SSDD 48 tpi 5.25"
DENSITY MFM, LOW
CYLINDERS 40 SIDES 1 SECTORS 10 512
```

```

CYLINDERS 40 SIDES 1 SECTORS 10,512
SIDE1 0 1,3,5,7,9,2,4,6,8,10
BSH 4 BLM 15 EXM 1 DSM 94 DRM 127 AL0 0C0H AL1 0 OFS 2
END

```

```

#diskdef trso
# seclen 512
# tracks 40
# sectrk 10
# blocksize 2048
# maxdir 128
# datarate DD
# fm NO
# skew 2
# boottrk 2
# os 2.2
#end

```

```

#[trso]
#description = TRS-80 Model 4,4P Montezuma Extend System 190K - SSDD 48 tpi
5.25"
#cylinders = 40
#heads = 1
#sectors = 10
#secbase = 1
#secsize = 512
#datarate = DD
#rwwgap = 12
#fmtgap = 23
#fm = N
#multitrack = N
#skipdeleted = Y

```

```

BEGIN TRSP TRS-80 Model 4,4P Montezuma Extend System 390K - SSDD 48 tpi 5.25"
DENSITY MFM, LOW
CYLINDERS 40 SIDES 2 SECTORS 10,512
SIDE1 0 1,3,5,7,9,2,4,6,8,10
SIDE2 1 1,3,5,7,9,2,4,6,8,10
ORDER SIDES
BSH 4 BLM 15 EXM 1 DSM 194 DRM 127 AL0 0C0H AL1 0 OFS 2
END

```

```

#diskdef trsp
# seclen 512
# tracks 80
# sides alt
# sectrk 10
# blocksize 2048
# maxdir 128
# datarate DD
# fm NO
# skew 2
# boottrk 2
# os 2.2
#end

```

```

#[trsp]

```

```
#description = TRS-80 Model 4,4P Montezuma Extend System 390K - SSDD 48 tpi
5.25"
#sides=alt
#cylinders = 40
#heads = 2
#sectors = 10
#secbase = 1
#secsize = 512
#datarate = DD
#rwwgap = 12
#fmtgap = 23
#fm = N
#multitrack = N
#skipdeleted = Y

#NOT TESTED
BEGIN TRSQ TRS-80 Model 4,4P Montezuma Extend System 390K - SSDD 96 tpi 5.25"
DENSITY MFM, LOW
CYLINDERS 80 SIDES 1 SECTORS 10,512
SIDE1 0 1,3,5,7,9,2,4,6,8,10
BSH 4 BLM 15 EXM 0 DSM 194 DRM 127 ALO 0C0H AL1 0 OFS 2
END

#diskdef trsq
# seclen 512
# tracks 80
# sectrk 10
# blocksize 2048
# maxdir 128
# datarate DD
# fm NO
# skew 2
# boottrk 2
# os 2.2
#end

#[trsq]
#description = TRS-80 Model 4,4P Montezuma Extend System 390K - SSDD 96 tpi
5.25"
#cylinders = 80
#heads = 1
#sectors = 10
#secbase = 1
#secsize = 512
#datarate = DD
#rwwgap = 12
#fmtgap = 23
#fm = N
#multitrack = N
#skipdeleted = Y

#NOT TESTED
BEGIN TRSR TRS-80 Model 4,4P Montezuma Extend System 790K - DSDD 96 tpi 5.25"
DENSITY MFM, LOW
CYLINDERS 80 SIDES 2 SECTORS 10,512
SIDE1 0 1,3,5,7,9,2,4,6,8,10
SIDE2 1 1,3,5,7,9,2,4,6,8,10
```

ORDER SIDES

```
BSH 4 BLM 15 EXM 0 DSM 394 DRM 127 AL0 0C0H AL1 0 OFS 2
END
```

```
#diskdef trsr
# seclen 512
# tracks 160
# sides alt
# sectrk 10
# blocksize 2048
# maxdir 128
# datarate DD
# fm NO
# skew 2
# boottrk 2
# os 2.2
#end
```

```
#[trsr]
#description = TRS-80 Model 4,4P Montezuma Extend System 790K - DSDD 96 tpi
5.25"
#sides=alt
#cylinders = 80
#heads = 2
#sectors = 10
#secbase = 1
#secsize = 512
#datarate = DD
#rwwgap = 12
#fmtgap = 23
#fm = N
#multitrack = N
#skipdeleted = Y
```

```
BEGIN TRSS TRS-80 Model 4,4P Montezuma Super Data 220K - SSDD 48 tpi 5.25"
DENSITY MFM, LOW
CYLINDERS 40 SIDES 1 SECTORS 6,1024
SIDE1 0 1,3,5,2,4,6
BSH 4 BLM 15 EXM 1 DSM 109 DRM 127 AL0 0C0H AL1 0 OFS 0
END
```

```
#diskdef trss
# seclen 1024
# tracks 40
# sectrk 6
# blocksize 2048
# maxdir 128
# datarate DD
# fm NO
# skew 2
# boottrk 0
# os 2.2
#end
```

```
#[trss]
#description = TRS-80 Model 4,4P Montezuma Super Data 220K - SSDD 48 tpi 5.25"
#cylinders = 40
```

```

# cylinders = 40
#heads = 1
#sectors = 6
#secbase = 1
#secsize = 1024
#datarate = DD
#rwgap = 12
#fmtgap = 23
#fm = N
#multitrack = N
#skipdeleted = Y

```

```

BEGIN TRST TRS-80 Model 4,4P Montezuma Super Data 440K - DSDD 48 tpi 5.25"
DENSITY MFM, LOW
CYLINDERS 40 SIDES 2 SECTORS 6,1024
SIDE1 0 1,3,5,2,4,6
SIDE2 1 1,3,5,2,4,6
ORDER EAGLE
BSH 4 BLM 15 EXM 1 DSM 219 DRM 127 AL0 0C0H AL1 0 OFS 0
END

```

```

#diskdef trst
# seclen 1024
# tracks 80
# sides outout
# sectrk 6
# blocksize 2048
# maxdir 128
# datarate DD
# fm NO
# skew 2
# boottrk 0
# os 2.2
#end

```

```

#[trst]
#description = TRS-80 Model 4,4P Montezuma Super Data 440K - DSDD 48 tpi 5.25"
#sides=outout
#cylinders = 40
#heads = 2
#sectors = 6
#secbase = 1
#secsize = 1024
#datarate = DD
#rwgap = 12
#fmtgap = 23
#fm = N
#multitrack = N
#skipdeleted = Y

```

```

#NOT TESTED
BEGIN TRSU TRS-80 Model 4,4P Montezuma Super Data 440K - SSDD 96 tpi 5.25"
DENSITY MFM, LOW
CYLINDERS 80 SIDES 1 SECTORS 6,1024
SIDE1 0 1,3,5,2,4,6
BSH 4 BLM 15 EXM 1 DSM 219 DRM 127 AL0 0C0H AL1 0 OFS 0
END

```

```
#diskdef trsu
# seclen 1024
# tracks 80
# sectrk 6
# blocksize 2048
# maxdir 128
# datarate DD
# fm NO
# skew 2
# boottrk 0
# os 2.2
#end
```

```
#[trsu]
#description = TRS-80 Model 4,4P Montezuma Super Data 440K - SSDD 96 tpi 5.25"
#cylinders = 80
#heads = 1
#sectors = 6
#secbase = 1
#secsize = 1024
#datarate = DD
#rwwgap = 12
#fmtgap = 23
#fm = N
#multitrack = N
#skipdeleted = Y
```

```
#NOT TESTED
BEGIN TRSV TRS-80 Model 4,4P Montezuma Super Data 880K - DSDD 96 tpi 5.25"
DENSITY MFM, LOW
CYLINDERS 80 SIDES 2 SECTORS 6,1024
SIDE1 0 1,3,5,2,4,6
SIDE2 1 1,3,5,2,4,6
ORDER SIDES
BSH 4 BLM 15 EXM 0 DSM 439 DRM 127 AL0 0C0H AL1 0 OFS 0
END
```

```
#diskdef trsv
# seclen 1024
# tracks 160
# sides alt
# sectrk 6
# blocksize 2048
# maxdir 128
# datarate DD
# fm NO
# skew 2
# boottrk 0
# os 2.2
#end
```

```
#[trsv]
#description = TRS-80 Model 4,4P Montezuma Super Data 880K - DSDD 96 tpi 5.25"
#sides=alt
#cylinders = 80
#heads = 2
```

```

#heads = 1
#sectors = 6
#secbase = 1
#secsize = 1024
#datarate = DD
#rwwgap = 12
#fmtgap = 23
#fm = N
#multitrack = N
#skipdeleted = Y

```

```

BEGIN TRSW TRS-80 Model 4,4P Montezuma System 400K - SSDD 96 tpi 3.5"
DENSITY MFM, LOW
CYLINDERS 80 SIDES 1 SECTORS 10,512
SIDE1 0 1,3,5,7,9,2,4,6,8,10
BSH 4 BLM 15 EXM 1 DSM 199 DRM 127 AL0 0C0H AL1 0 OFS 0
END

```

```

#diskdef trsw
# seclen 512
# tracks 80
# sectrk 10
# blocksize 2048
# maxdir 128
# datarate DD
# fm NO
# skew 2
# boottrk 0
# os 2.2
#end

```

```

#[trsw]
#description = TRS-80 Model 4,4P Montezuma System 400K - SSDD 96 tpi 3.5"
#cylinders = 80
#heads = 1
#sectors = 10
#secbase = 1
#secsize = 512
#datarate = DD
#rwwgap = 12
#fmtgap = 23
#fm = N
#multitrack = N
#skipdeleted = Y

```

Larry

Re: [HOWTO] - Access CP/M Floppy's via libdsk & cpmtools

by **Ikraemer**

Posted: **2014-03-11 12:46**

LIBDSK - SUPPORTS

Disk image types supported:

remote : Remote LibDsk instance

.....libdsk server at the far end of a serial connection

.....Remote LibDsk server, most likely at the other end of a serial line.

rcpmfs : Reverse CP/MFS driver
Host directory presented as a CP/M filesystem
Reverse CP/M filesystem. A directory is made to appear as a CP/M disk. This is an experimental
system and should be approached with caution.

floppy : Linux floppy driver
Host system's floppy drive (under Linux, DOS, or Windows).

dsk : CPCEMU .DSK driver
Disc image in the DSK format used by CPCEMU. The format of a .DSK file is described in the CPCEMU documentation.

edsk : Extended .DSK driver
Disc image in the extended CPCEMU DSK format.

apridisk : APRIDISK file driver
Disc image in the format used by the ApriDisk utility. The format is described in apridisk.html.

qm : CopyQM file driver
Disc images created by Sydex's CopyQM. This is a read-only driver.

tele : TeleDisk file driver
Disc images created by Sydex's TeleDisk. This is a read-only driver.

ydisk : YAZE YDSK driver
Disc image format used by the yaze and yaze-ag CP/M emulators.

raw : Raw file driver
Raw disc image – as produced by "dd if=/dev/fd0 of=image"
On systems other than Linux, DOS or Windows, this is also used to access the host system's floppy drive.

myz80 : MYZ80 hard drive driver
MYZ80 hard drive image, which is nearly the same as "raw" but has a 256 byte header.

simh : SIMH disc image driver

nanowasp : NanoWasp image file driver
Disc image in the 400k Microbee format used by the NanoWasp emulator. This is similar to "raw", but the
tracks are stored in a different order. LibDsk also applies a sector skew so that the sectors are
read/written in the logical order. Strictly speaking, it should not do this (when libdsk is used
with cpmtools, cpmtools is the one that does the skewing) but cpmtools cannot handle the skewing
scheme used by the Microbee format.

logical : Raw file logical sector order
Raw disc image in logical filesystem order. Previous versions of LibDsk could generate such
images (for example, by using the now-deprecated-logical option to dsktrans) but couldn't then write
them back or use them in emulators.

cfi : CFI file driver
Compressed floppy image, as produced by FDCOPY.COM under DOS. Its format is described in cfi.html.

LibDsk is a library that attempts to create uniform functions for accessing:

- * Floppy drives, under Linux, Windows and MSDOS;
- * Raw "dd" disc images (including floppy drives under other Unixes);
- * .DSK files (as used by CPCEMU, JOYCE, XZX, etc.)
- * MYZ80 hard drive images (as used by MYZ80 and JOYCE).
- * Various other disc image formats which this margin is too small to enumerate.

LIBDSK has drivers for:

Raw files (including /dev/fdn).

Raw files, rearranged into logical filesystem order.

.DSK files, as used in CPCEMU, JOYCE and other Sinclair/Amstrad emulators.

MYZ80 hard drive image files.

NanoWasp floppy image files.

.CFI (Compressed Floppy Image) files, as created by FDCOPY.COM under DOS.

The Linux floppy drive (supports CPC System and Data formats, which the standard "Raw file" driver does not).

The Windows 3.x/95/98/ME/NT/2000 floppy drive.

The DOS floppy drive (via the PC BIOS).

CopyQM files (read-only)

TeleDisk files (read-only)

APRIDISK image files

The experimental LDBS format

rcpmfs – a backend that makes a Unix/Windows directory appear to be a CP/M disc image.

Another instance of LibDsk over a serial line.

LIBDSK also has the following default CP/M Floppy definitions included.

```
pcw180      : PCW / IBM 180k
pcsys       : CPC System
pcpdata     : CPC Data
pcw720      : PCW / IBM 720k
pcw1440     : PCW16 / IBM 1440k
ibm160      : IBM 160k (CP/M-86 / DOSPLUS)
ibm320      : IBM 320k (CP/M-86 / DOSPLUS)
ibm360      : IBM 360k (CP/M-86 / DOSPLUS)
ibm720      : IBM 720k (144FEAT)
ibm1200     : IBM 1.2M (144FEAT)
ibm1440     : IBM 1.4M (144FEAT)
acorn160    : Acorn 160k
acorn320    : Acorn 320k
acorn640    : Acorn 640k
acorn800    : Acorn 800k
acorn1600   : Acorn 1600k
pcw800      : PCW 800k
pcw200      : PCW 200k
bbc100      : BBC 100k
bbc200      : BBC 200k
mbee400     : Microbee 400k
mgt800      : MGT 800k
trdos640    : TR-DOS 640k
ampro200    : Ampro 40 track single-sided
ampro400d   : Ampro 40 track double-sided
ampro400s   : Ampro 80 track single-sided
ampro800    : Ampro 80 track double-sided
myz80       : MYZ80 8Mb
pcpm320     : IBM 320k (CP/M-86 / DOSPLUS)
```

Additional CP/M Disk Definitions can be added in /home/user/.libdskrc

DSKTRANS — Copy from one floppy or image file to another

SYNOPSIS:

```
dsktrans [-itype TYPE] [-otype TYPE] [-iside SIDE] [-oside SIDE] [-icomp COMP] [-ocomp COMP]
[-idstep] [-odstep] [-retry COUNT] [-format FMT] [-first CYLINDER] [-last CYLINDER]
[-comment TEXT] [-comment @FILE] [-md3] [-logical] [-apricot] [-pcdos] [-noformat]
INPUT-IMAGE OUTPUT-IMAGE
```

dsktrans copies floppy discs or images, optionally converting image types. This is a high-level copy which assumes that the disc has a straightforward geometry; all the tracks are assumed to have the same layout of sectors, and interleave is not preserved.

To convert the imagedisk .imd images back to .img (raw - you can rename .raw to .img) use:

```
dsktrans -otype raw -format sin2 ElwroG1.dsk ElwroG1.raw
dsktrans -otype raw -format amp1 AmprOrig.dsk AmprOrig.raw
cpmls -d -T raw,elwro -f elwro ElwroG1.img
cpmls -d -T raw,pcw720 -f elwro ElwroG1.img
cpmls -d -T edsk,elwro -f elwro ElwroG1.dsk
libdisk can also be used standalone to convert from a Cylinder/Head/Sector disk image
(as used by, for example, VirtualBox)
to a dump in file system order:
```

```
dsktrans -itype raw -otype logical -format ibm1440 infile.vfd outfile.cpmfs
and back:
dsktrans -itype logical -otype raw -format ibm1440 infile.cpmfs outfile.vfd
```

```
dsktrans -otype floppy 720CPM86.IMG /dev/fd0
```

```
dsktrans /dev/fd0 filename.ufi -otype raw
cpmcp -f amstrad filename.ufi 0:filename .
```

I have two Amstrad files that are CPM8256.DSK & CPM9512.DSK, and I wanted to convert them to a .RAW file (Sector Dump). I used libdisk's dsktrans command to do this.

```
dsktrans -itype qm -otype raw input output
or even
```

```
dsktrans -itype qm -otype floppy input /dev/fd0
$ dsktrans -itype dsk -otype raw CPM8256.dsk CPM8256.RAW
$ dsktrans -itype edsk -otype raw CPM9512.dsk CPM9512.RAW
```

Then I used cpmtools to get a directory listing of both files. (I could have used dsktrans to write directly to Floppy.)

```
$ cpmls -f ams0 -D CPM8256.RAW
```

Name	Bytes	Recs	Attr	update	create
BASIC .COM	28K	224			
DIR .COM	15K	114			
DISCKIT .COM	7K	56			
ED .COM	10K	73			
ERASE .COM	4K	29			
J15ACPM3.EMS	40K	320	R		
KEYS .WP	1K	7			
LANGUAGE.COM	1K	8			

```

PALETTE .COM      1K      8
PAPER   .COM      2K     16
PIP     .COM      9K     68
PROFILE .SUB      1K      2
RENAME  .COM      3K     23
RPED    .BAS      7K     55
RPED    .SUB      1K      1
SET     .COM     11K     81
SET24X80.COM      1K      8
SETDEF  .COM      4K     32
SETKEYS .COM      2K     16
SETLST  .COM      2K     16
SETSI0  .COM      2K     16
SHOW    .COM      9K     66
SUBMIT  .COM      6K     42
TYPE    .COM      3K     24

```

```

    24 Files occupying    170K,        3K Free.
$ cpmfs -f ams0 -D CPM9512.RAW

```

Name	Bytes	Recs	Attr	update	create
ASSIGN .SYS	1K	1			
BASIC .COM	28K	221			
CPMKEYS .COM	1K	8			
DAISY .COM	4K	25			
DATE .COM	3K	23			
DDFXHR8 .PRL	15K	118			
DDFXLR8 .PRL	12K	96			
DDHP7470.PRL	11K	86			
DDSCREEN.PRL	5K	40			
DEMO .BAS	2K	11			
DEVICE .COM	8K	58			
DIR .COM	15K	114			
DISCKIT .COM	10K	74			
DUMP .COM	1K	8			
ED .COM	10K	73			
ERASE .COM	4K	29			
EX1 .BAS	2K	13			
EX2 .BAS	3K	23			
EX3 .BAS	5K	34			
EXGSX .BAS	2K	13			
EXJTSAM .BAS	2K	16			
EXJTSAM2.BAS	4K	28			
EXJTSAMI.BAS	1K	1			
GENCOM .COM	15K	116			
GENGRAF .COM	2K	12			
GET .COM	7K	51			
GSX .SYS	2K	11			
GSXPREP .BAS	1K	2			
HELP .COM	7K	56			
HELP .HLP	95K	757			
HEXCOM .COM	2K	9			
HIST .UTL	2K	10			
INITDIR .COM	32K	250			
KEYS .DRL	1K	7			
KEYS .WP	1K	7			
LANGUAGE.COM	1K	8			
LIB .COM	7K	56			

```

LINK      .COM      16K      123
LOGO      .COM      50K      394
LOGO      .SUB       1K       1
MAC       .COM      12K      92
MAIL232   .COM       4K      32
MATRIX    .COM       3K      17
PALETTE   .COM       1K       8
PAPER     .COM       3K      17
PATCH    .COM       3K      20
PIP       .COM       9K      68
PROFILE   .SUB       1K       1
PUT       .COM       7K      55
RAMTEST   .COM       5K      40
RENAME    .COM       3K      23
RMAC      .COM      14K     106
RPED      .BAS       7K      55

```

User 8:

Name	Bytes	Recs	Attr	update	create
RAMTEST .COM	5K	38			
54 Files occupying		163K,		10K Free.	

SAMdisk – SUPPORTS

The following containers types are supported for input:

EDSK – Extended disk image (Amstrad CPC, Spectrum +3, PC)
 MGT – MGT +D/Disciple/SAM (Sinclair Spectrum / SAM Coupé)
 SAD – Sam Disk (SAM Coupé)
 SBT – Sam BooTable disk (SAM Coupé)
 SDF – Sam Disk Format (SAM Coupé)
 CPM – Pro-DOS CP/M (SAM Coupé)
 TD0 – Sydex TeleDisk (various)
 RAW – KryoFlux stream format [IBM MFM and FM only].
 SCP – SuperCard Pro stream format [IBM MFM and FM only].
 DFI – DiscFerret stream format [IBM MFM and FM only].
 IPF – Interchangeable Preservation Format [IBM MFM only]
 DMK – David M Keil's disk format (mainly TRS-80)
 IMD – ImageDisk utility image (various)
 TRD – Beta128 disk for TR-DOS (Sinclair Spectrum)
 FDI – Full Disk Image (Sinclair Spectrum, not Disk2FDI!)
 OPD – OPus Discovery (Sinclair Spectrum)
 MBD – MB-02+ Disk (Sinclair Spectrum)
 UDI – Ultra Disk Image (Sinclair Spectrum)
 SCL – Sinclair betadisk archive (Sinclair Spectrum)
 DSK – Disk image (Amstrad CPC)
 DSC – WinAPE disk image (Amstrad CPC)
 CFI – Compressed Floppy Image (Amstrad)
 BPB – FAT12/16 BIOS Parameter Block (MS-DOS, Atari ST)
 MSA – Magic Shadow Archive (Atari ST)
 D80 – Didaktik D80
 D88 – Pasopeia D88 (NEC PC-xx)
 D81 – Commodore 1581
 D2M – Commodore CMD FD-2000

D2M – Commodore CMD FD-2000
 D4M – Commodore CMD FD-4000
 LIF – Logical Interchange Format (Hewlett-Packard)
 S24 – Sega System 24 (Arcade, 1.8M and 1.88M formats)
 RAW – some raw sector dumps, identified by file size only

The following types are supported for output of new or modified disk images:

.DSK – Extended DSK flexible format
 .MGT – Miles Gordon Technology +D/SAM (80/2/10/512)
 .CPM – SAM Coupé Pro-DOS (80/2/9/512)
 .TRD – Beta128 / TR-DOS (typically 80/2/16/256)
 .FDI – Full Disk Image flexible format (no connection to Disk2FDI)
 .OPD – Opus Discovery (typically 40/1/18/256)
 .IMD – ImageDisk flexible format
 .D81 – Commodore 1581 (80/2/10/512)
 .D88 – Pasopeia D88 (NEC PC-xx) flexible format
 .D2M – CMD FD-2000 (81/2/10/1024)
 .LIF – HP Logical Interchange Format (77/2/16/256)
 .RAW – Raw sector dump in suitable regular format

<http://simonowen.com/samdisk/formats/>

Is there a utility to convert the imagedisk .imd images back to .img raw?

Found the solution using simon owen's samdisk
 samdisk test.imd test.raw
 rename .raw to .img and it even booted !!!

Another solution is to use ImageDisks IMDU
 IMDU file.img file.raw /E /B
 If you want more information about the image use the following command:
 IMDU file.img file.raw /E /B /D > file-log.txt
 Then read the file-log.txt document.

Larry

Re: [HOWTO] - Access CP/M Floppy's via libdisk & cpmtools

by **Ikraemer**

Posted: **2014-03-25 14:12**

LIBDSK

Usage Commands:

dskscan
 dsktrans
 dskdump
 dskform
 dskparse
 dskconv
 dskid
 dskutil

For more detailed information on the commands:

man dskscan
 man dsktrans
 man dskdump
 man dskform

man dskparse
 man dskconv
 man dskid
 man dskutil
dskscan

Syntax:
 dskscan {options} image
 Options are:
 -type <type> type of input disc image
 -side <side> Force side 0 or side 1 of input
 -retry <count> Set number of retries on error
 -first <cyl> Start at specified cylinder
 -last <cyl> Scan up to specified cylinder
 -dstep Double-step
 -xml Output as XML
 -format Force a specified format name

Default type is autodetect.

eg: dskscan /dev/fd0
 dskscan -xml /dev/fd1

Valid DEFAULT formats are:

pcw180 : PCW / IBM 180k
 cpcsys : CPC System
 cpcdata : CPC Data
 pcw720 : PCW / IBM 720k
 pcw1440 : PcW16 / IBM 1440k
 ibm160 : IBM 160k (CP/M-86 / DOSPLUS)
 ibm320 : IBM 320k (CP/M-86 / DOSPLUS)
 pcpm320 : IBM 320k (CP/M-86 / DOSPLUS)
 ibm360 : IBM 360k (DOSPLUS)
 ibm720 : IBM 720k (144FEAT)
 ibm1200 : IBM 1.2M (144FEAT)
 ibm1440 : IBM 1.4M (144FEAT)
 acorn160 : Acorn 160k
 acorn320 : Acorn 320k
 acorn640 : Acorn 640k
 acorn800 : Acorn 800k
 acorn1600 : Acorn 1600k
 pcw800 : PCW 800k
 pcw200 : PCW 200k
 bbc100 : BBC 100k
 bbc200 : BBC 200k
 mbee400 : Microbee 400k
 mgt800 : MGT 800k
 trdos640 : TR-DOS 640k
 myz80 : MYZ80 8Mb

Additional CP/M Floppy Disk definition can be added to your

/home/user/.libdiskrc file.

/home/user/.libdiskrc file permissions are:

root@debian:~# ls -l /home/user/.libdiskrc
 -rw-r--r-- 1 user user 152012 Mar 10 20:57 .libdiskrc

```

# This is FMT_800K as a libdiskrc entry
# [xcf2dd]
# Description = 800k XCF2DD format
# Sides = Alt
# Cylinders = 80
# Heads = 2
# Sectors = 10
# SecBase = 1
# SecSize = 512
# DataRate = SD
# RWGap = 12
# FmtGap = 23
# [xcf2]
# Description = 200k XCF2 format
# Cylinders = 40
#
#
# [ibm3740]
# description = IBM3740 SS SD 77T 8" 26x128 b/s
# cylinders = 77
# heads = 1
# sectors = 26
# secbase = 1
# secsize = 512
# datarate = SD
# rwgap = 12
# fmtgap = 23
# fm = N
# multitrack = N
# skipdeleted = Y

```

/home/user/.libdiskrc contains:

```

# Anything after a semicolon or hash character is treated as a comment
# Blank lines are also ignored.
#
# description=DESC The description of the format as shown by (for examp
# sides=TREATMENT How a double-sided disk is handled. This can either b
#       (sides alternate - used by most PC-hosted operating systems), o
#       0 tracks 0-79, then side 1 tracks 79-0 - used by 144FEAT CP/M d
#       (use side 0 tracks 0-79, then side 1 tracks 0-79 - used by some
#       If the disk is single-sided, this parameter can be omitted.
# cylinders=COUNT Sets the number of cylinders (usually 40 or 80).
# heads=COUNT Sets the number of heads (usually 1 or 2 for single- or d
# sectors=COUNT Sets the number of sectors per track.
# secbase=NUMBER Sets the first sector number on a track. Usually 1; so
#       formats use 0.
# secsize=COUNT Sets the size of a sector in bytes. This should be a po
# datarate=VALUE Sets the rate at which the disk should be accessed. Th
#       HD, DD, SD or ED.
# rwgap=VALUE Sets the read/write gap.
# fmtgap=VALUE Sets the format gap.
# fm=Y or N Sets the recording mode - Y for FM, N for MFM.
# multitrack=Y or N Sets multitrack mode.
# skipdeleted=Y or N Sets whether to skip deleted data.
#
#libdiskrc example:
# This is FMT_800K as a libdiskrc entry
#[xcf2dd]
#Description = 800k XCF2DD format
#Sides = Alt
#Cylinders = 80
#Heads = 2
#Sectors = 10
#SecBase = 1
#SecSize = 512
#DataRate = SD
#RWGap = 12
#FmtGap = 23
#[xcf2]
#Description = 200k XCF2 format
#Cylinders = 40
#
#
# [ibm3740]
description = IBM3740 SS SD 77T 8" 26x128 b/s
cylinders = 77
heads = 1
sectors = 26
secbase = 1
seccsize = 512
datarate = SD
rwgap = 12
fmtgap = 23
fm = N
multitrack = N
skipdeleted = Y

```

```
#
# According to the printed documentation I have for Version 1.39 of 22D
#
#     SIDES specifies that first one side then the other is written/
#           before the cylinder number is advanced.
#
#     CYLINDERS specifies that all tracks on the first side are reco
#           then all tracks on second side. Tracks on side 0 are wr
#           from 0 up and side 1 down to 0.
#
#     EAGLE same as CYLINDERS except both sides start at zero or out
#           track.
#
#     COLUMBIA same as CYLINDERS except tracks are sequential from s
#           0 to side 1 ie. 0 - 79 for a 40 track disk.
#
#     EVEN-ODD means side 0 contains even-numbered tracks and side 1
#           contains odd-numbered tracks.
#
#
[kpii]
description = Kaypro II/2 - SSDD 48 tpi 5.25"
cylinders = 40
heads = 1
sectors = 10
secbase = 0
secsize = 512
datarate = DD
rwgap = 12
fmtgap = 23
fm = N
multitrack = N
skipdeleted = Y

[kpiv]
description = Kaypro 2X/4/10 - DSDD 48 tpi 5.25"
cylinders = 80
heads = 2
sectors = 10
secbase = 0
secsize = 512
datarate = DD
rwgap = 12
fmtgap = 23
fm = N
multitrack = N
skipdeleted = Y

[amp1]
description = Ampro - SSDD 48 tpi 5.25"
cylinders = 40
heads = 1
sectors = 10
secbase = 1
secsize = 512
datarate = DD
```

```
rwgap = 12
fmtgap = 23
fm = N
multitrack = N
skipdeleted = Y
```

```
[amp2]
description = Ampro - DSDD 48 tpi 5.25"
sides=alt
cylinders = 40
heads = 2
sectors = 10
secbase = 17
seccsize = 512
datarate = DD
rwgap = 12
fmtgap = 23
fm = N
multitrack = N
skipdeleted = Y
```

```
[amp3]
description = Ampro - SSDD 96 tpi 5.25"
cylinders = 80
heads = 1
sectors = 5
secbase = 17
seccsize = 1024
datarate = DD
rwgap = 12
fmtgap = 23
fm = N
multitrack = N
skipdeleted = Y
```

```
[amp4]
description = Ampro - DSDD 96 tpi 5.25"
sides=alt
cylinders = 80
heads = 2
sectors = 5
secbase = 17
seccsize = 1024
datarate = DD
rwgap = 12
fmtgap = 23
fm = N
multitrack = N
skipdeleted = Y
```

```
[amp5]
description = Ampro - SSDD 96 tpi 3.5"
cylinders = 80
heads = 1
sectors = 5
secbase = 1
seccsize = 1024
```

```

datarate = DD
rwgap = 12
fmtgap = 23
fm = N
multitrack = N
skipdeleted = Y

[amp6]
description = Ampro - DSDD 96 tpi 3.5"
sides=alt
cylinders = 80
heads = 2
sectors = 5
secbase = 17
seclsize = 1024
datarate = DD
rwgap = 12
fmtgap = 23
fm = N
multitrack = N
skipdeleted = Y

[ampro800]
description = Ampro800 - DSDD 96 tpi 3.5"
sides=alt
sectors = 5
secbase = 17
seclsize = 1024
datarate = DD
rwgap = 12
fmtgap = 23
fm = N
multitrack = N
skipdeleted = Y
LIBDSK DOCUMENTATION

```

Detailed LibDsk Documentation is located at:

/home/user/your/path/to/libdsk-1.3.5/doc/

Enjoy.

Larry

Re: [HOWTO] - Access CP/M Floppy's via libdsk & cpmtools

by **Ikraemer**

Posted: **2016-01-11 00:29**

UPDATE 01-25-2017

The current version of cpmtools is 2.20 and is located at:

[http://www.moria.de/~michael/cpmtools/f20.tar.gz](http://www.moria.de/~michael/cpmtools/f...20.tar.gz)

The current version of libdsk is 1.4.2 and is located at:

<http://www.seasip.info/Unix/LibDsk/libdsk-1.4.2.tar.gz>

Note that libdsk now requires lyx and lyx-common to be installed.

You may have built cpmtools (without libdsk) or you may have built cpmtools with

libdisk, to allow more disk image access possibilities.

In your searches for CP/M images you've likely stumbled on Dave Dunfield's site with several different System Images in .IMD format at:
<http://www.classiccmp.org/dunfield/img54306/system.htm>

Let's access a few of these IMD files with cpmtools (with libdisk).

I've selected one of the CP/M files at random, to test cpmtools and libdisk:

<http://www.classiccmp.org/dunfield/img54306/d/zorba.zip>

The first thing to do is download some .ZIP files, extract their contents in separate subdirectories, and read the included README.TXT

The first Windows command I use in XP is:

```
IMDU ZORBA1.IMD > ZORBA1.TXT
```

which gives me a lot of detail about the .IMD image:

```
IMageDisk Utility 1.18 / Mar 07 2012
IMD 1.01: 3/08/2005 11:16:41

Zorba Personal Computer

System Disk (BIOS version 1.7)A

CP/M 2.2 CP2-865-04553

CBASIC CBA-865-04553

Copyright 1981 by Digital Research, Inc.

0/0 250 kbps DD 10x512
80 tracks(40/40), 800 sectors (465 Compressed)
```

This information tells me it's a 40 Track Image with 2 Heads, and with 10 Sectors of 512 Bytes. You will need this information to select the proper definition for cpmtools and libdisk.

ZORBA.ZIP contains:

ZORBA1.IMD, ZORBA2.IMD, ZORBA3.IMD, ZORBA4.IMD, ZORBA5.IMD & README.TXT

Next, you need to create the .RAW image files of all .IMD files in Windows or in DOSBOX. I use IMDU:

```
IMDU ZORBA1.IMD ZORBA1.RAW /E /B
IMDU ZORBA2.IMD ZORBA2.RAW /E /B
IMDU ZORBA3.IMD ZORBA3.RAW /E /B
IMDU ZORBA4.IMD ZORBA4.RAW /E /B
IMDU ZORBA5.IMD ZORBA5.RAW /E /B
```

OR

```
IMDU ZORBA1.IMD ZORBA1.RAW /E /B /D > ZORBA1.TXT
IMDU ZORBA2.IMD ZORBA2.RAW /E /B /D > ZORBA2.TXT
IMDU ZORBA3.IMD ZORBA3.RAW /E /B /D > ZORBA3.TXT
IMDU ZORBA4.IMD ZORBA4.RAW /E /B /D > ZORBA4.TXT
IMDU ZORBA5.IMD ZORBA5.RAW /E /B /D > ZORBA5.TXT
```

To create the .RAW images for accessing with cpmtools, or cpmtools with libdisk.

The zorba definitions are:

```
# ZOR1 Zorba - DSDD 48 tpi 5.25" - 512 x 10
```

```
diskdef zor1
  seclen 512
  tracks 40
  sectrk 20
  blocksize 2048
  maxdir 64
  skew 1
  offset 10240
  boottrk 0
  os 2.2
end
```

```
# libdsk data below
```

```
[zor1]
description = ZOR1 Zorba - DSDD 48 tpi 5.25" - 512 x 10
#sides = extsurface
cylinders = 80
heads = 2
seclen = 512
sectors = 10
seclen = 1
datarate = DD
```

```
# ZOR2 Zorba - DSDD 96 tpi 5.25" - 512 x 10
```

```
diskdef zor2
  seclen 512
  tracks 80
  sectrk 20
  blocksize 4096
  maxdir 128
  skew 1
  offset 10240
  boottrk 0
  os 2.2
end
```

```
# libdsk data below
```

```
[zor2]
description = ZOR2 Zorba - DSDD 96 tpi 5.25" - 512 x 10
#sides = extsurface
cylinders = 160
heads = 2
seclen = 512
sectors = 10
seclen = 1
datarate = DD
```

The ZOR1 definition is the one that fits the IMDU Imagedisk information.

cpmtools (without libdsk) commands

```
man cpmls
```

```
man cpmcp
```

```
cpmls -f zor1 -d ZORRA1 RAW
```

cpmtools - ZORBA & ZORBA1.RAW

```
ASM COM : DDT COM : ED COM : LOAD COM
MOVCPM COM : PIP COM : STAT COM : SUBMIT COM
XSUB COM : DUMP COM : M80 COM : L80 COM
CREF80 COM : LIB80 COM : CBAS2 COM : CRUN2 COM
XREF COM : SGEN COM : PATCH COM : SETUP COM
FORMAT COM : TRACKRD COM : RELOAD COM : VT52 KEY
TALK COM
```

```
cpmcp -f zor1 ZORBA1.RAW 0:*.COM /home/larry/IMDs/zorba
cpmcp -f zor1 ZORBA1.RAW 0:vt52.key /home/larry/IMDs/zorba
more vt52.key
```

cpmtools with libdsk commands

```
cpmls -f zor1 -T raw,zor1 -d ZORBA1.RAW
cpmcp -f zor1 -T raw,zor1 ZORBA1.RAW 0:*.COM /home/larry/IMDs/zorba
cpmcp -f zor1 -T raw,zor1 ZORBA1.RAW 0:vt52.key /home/larry/IMDs/zorba
more vt52.key
cpmcp -f zor1 -T raw,zor1 ZORBA1.RAW /home/larry/IMDs/zorba/*.COM 0:
So, now you can easily copy files to and from your .RAW Images using cpmtools (or
cpmtools with libdsk).
```

But, maybe you also need to create a new .IMD from the .RAW image.

From DOSBox, the BIN2IMD utility will create an .IMD of a .RAW Image.
 BIN2IMD input-file output-image [option-file] [options]

```
BIN2IMD X.RAW X.IMD DM=2 N=80 SS=512 SM=1-8 /2
```

Where the Options are:

opts:	/1	- 1-sided output
	/2	- 2-sided output
	/C	- write Compressed sectors
	/U	- write Uncompressed sectors
	/V[0 1]	- Verbose output
	C=text @file	- image Comment
	N=#cylinders	- set Number of output cylinder
	DM[s]=0-5	- track Data Mode
	SS[s]=128-8192	- track Sector Size
	SM[s]=n[,n-n] [n.#]	- track Sector numbering Map
	CM[s]=n[,n-n] [n.#]	- track/sector Cylinder Map
	HM[s]=n[,n-n] [n.#]	- track/sector Head Map

and

DM= sets the Data Mode, which must be one of:

0	= 500 kbps FM	\	Note:	kbps indicates transfer
1	= 300 kbps FM	>		rate, not the data rate
2	= 250 kbps FM	/		which is 1/2 for FM
3	= 500 kbps MFM			encoding.
4	= 300 kbps MFM			
5	= 250 kbps MFM			

SS= sets the Sector Size, and must be one of:

128, 256, 512, 1024, 2048, 4096 or 8192.

See: Imagedisk's **IMD.TXT** for more details.

REF: [https://virtuallyfun.superglobalmegacor ... ent-177828](https://virtuallyfun.superglobalmegacor...ent-177828)

Larry

Re: [HOWTO] - Access CP/M Floppy's via libdisk & cpmtools

by Ikraemer

Posted: **2017-09-19 11:49**

UPDATE 09-19-2017

GETTING GARBAGE DIRECTORY LISTING??

If you have had trouble getting a directory listing, there are a few things that can be tried to get a valid directory listing.

Here is an example directory listing that is corrupt.

```
cpmls -D -f als1 61KCPM3B.RAW
```

```
larry@debian:~/IMD-118/Alspa$ cpmls -D -f als1 61KCPM3B.RAW
Name Bytes Recs Attr update create
-----
#6CO6.#P# 0K 0 R

User 3:

Name Bytes Recs Attr update create
-----
M2,IX2.-IM 0K 0 R

User 14:

Name Bytes Recs Attr update create
-----
'W2D!D.4M+ 0K 0 R
T~@L]8M.H> 0K 0 R
4 Files occupying 2K, 594K Free.
```

Looking at the output of the the imagedisk utilities, I see that the first track is 26 sectors x 128 bytes (3328 bytes), while the remaining tracks are 8 sectors x 1024 bytes (622592 bytes). I used hexedit in Debian to locate the CP/M Directory listing. It was at 0x2D00 which is at 11520 decimal. With that information I can try two stanzas in my als1 definition to see if that works.

```
imdu 61KCPM3B.IMD 61KCPM3B.RAW /B /E /D >> 61KCPM3B.TXT
```

[illegible]

```

D D D D D D D D
2/0 1 6 3 8 5 2 7 4
D D D D D D D DE5
..
..
..
75/0 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5
76/0 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 = 76*(8192) = 622592
77 tracks(77/0), 634 sectors (191 Compressed)

```

Original Definition

```
# ALS1  Alspa - SSDD 8" - 1024 x 8
```

```
diskdef als1
  seclen 1024
  tracks 77
  sectrk 8
  blocksize 2048
  maxdir 128
  skew 1
  boottrk 2
  os 2.2
end
```

Modified Definition:

```
# ALS1  Alspa - SSDD 8" - 1024 x 8
```

```
diskdef als1
  seclen 1024
  tracks 77
  sectrk 8
  blocksize 2048
  maxdir 128
  skew 1
  boottrk 0
#0x2D00 = 11520
  offset 11520
  os 2.2
end
```

I still get an error when I try to get a directory listing. So, I change back to my Original Definition.

There has to be another way to get the correct Directory listing, and then get the files extracted.

Maybe I can write the first track so that it matches the remaining 76 tracks and try to get a directory listing.

I know I won't be able to boot a floppy created from this file, but I want the files extracted, and don't need to boot this created image.

So, I use dd to write a track 0 which will contain 8 sectors of 1024 bytes. This will become the

first track of the output file. The Linux command to do this is:

```
dd if=/dev/zero of=test.raw count=8192 bs=1
```

That creates Track 0 which is now also 8192 bytes long. Then, I simply add the remainder of the data to

the "test.raw" file. That Linux command is:

```
dd if=61KCPM3B.RAW skip=3328 count=622592 bs=1 seek=8192 of=test.raw co
```

I skip the first 3328 bytes of the input file (Track 0), then copy the remaining bytes

```
((1024 * 8)*76) OR 76 Tracks
```

((1024 - 8) / 8) * 512 tracks.

The Directory listing is now sane, and I can copy the files to a folder.

```
cpmls -D -f als1 test.raw
```

OR

```
cpmls -f als1 -T raw,als1 test.raw
```

OR

```
cpmls -f als1 -T raw,als1 -D test.raw
```

```
Name Bytes Recs Attr update create
```

```
-----
ASM .COM 8K 64
CPMTODEC.ASM 20K 151
CPMTODEC.COM 4K 30
CPMTODEC.HEX 8K 55
CPMTODEC.PRN 38K 300
D .COM 4K 20
DBASEAPP.OVR 4K 31
DBASEBRO.OVR 2K 11
DBASEJOI.OVR 2K 7
DBASEMAI.OVR 8K 55
DBASEMOD.OVR 4K 23
DBASEMSC.OVR 4K 27
DBASEMSG.COM 8K 62
DBASERPG.OVR 4K 25
DBASESRT.OVR 2K 11
DBASETTL.OVR 2K 8
DBASEUPD.OVR 2K 8
DDT .COM 6K 38
DECTOCPM.ASM 18K 141
DO .COM 18K 142
DU .COM 6K 48
DUMP .COM 4K 24
ED .COM 8K 52
FCOPY .COM 4K 29
FORM1K .COM 6K 37
INSTRUCT.BAK 10K 74
LISTDEC .ASM 10K 79
LISTDEC .COM 4K 17
LISTDEC .HEX 4K 23
LISTDEC .PRN 20K 157
LOAD .COM 2K 14
LST . 12K 82
M73A .COM 10K 76
MBASIC .COM 24K 188
PIP .COM 8K 58
PRINT .SUB 2K 3
STAT .COM 6K 41
SUBMIT .COM 2K 10
SYST1K .COM 2K 16
UNLOAD .COM 2K 6
VME .DEF 2K 1
WIRE .LST 8K 58
WIRE108 .ASC 2K 16
WM .COM 10K 76
WM .HLP 4K 21
WSMSG .OVR 28K 218
WSOVLY1 .OVR 34K 266
```

```
WSVA .COM 16K 124
XSUB .COM 2K 6
ZIP .COM 30K 236
50 Files occupying 448K, 148K Free.
```

And it's now easy to copy the files to my created subdirectory.
`cpmcp -f als1 test.raw 0:*. * /home/larry/IMD-118/Alsps/61KCPM3B`
Thanks.

Larry

Re: [HOWTO] - Access CP/M Floppy's via libdisk & cpmtools

by Ikraemer

Posted: 2018-05-16 20:38

Post by Ikraemer » 2018-05-16 16:09

UPDATE 05-16-2018

I've just downloaded a *.TD0 or *.IMD file and created a *.RAW file but cpmtool is having problems with the first two tracks being a mixed variety of single & double density. How I proceed? How do I get a good Directory Listing, and extract good files?

This [HOWTO] assumes you know how to use DOSBox. If you don't know how to use DOSBox, here is a good Tutorial:

<http://forums.debian.net/viewtopic.php? ... TO#p359786>

If the Saved file is a .TD0 (TELEDISK) file, run the TD02IMD utility in DOSBox from a Linux Terminal (Command Window).
`C:>td02imd JADEDD.TD0 > JADEDD.TXT`

If the Saved file is a .IMD (IMAGEDISK) file, run the IMDA utility in DOSBox from a Linux Terminal.
`C:>imda JADEDD.IMD >> JADEDD.TXT`
`C:>imdu JADEDD.IMD JADEDD.RAW /B /E /D >> JADEDD.TXT`
Exit DOSBox.

The TD02IMD utility gave results similar to these:

```
TD 1.5 5.25 LD MFM S-step, 2 sides ADV 22/09/1995 17:12:04
CP/M 2.2 system disk for Wavemate Bullet
DSDD 256 byte sector, 1-16, 1:1
80 tracks, 1280 sectors converted.
```

Which gives you a bit of information on the version of TELEDISK and Image information such as DS/DD 80 Tracks, Sector Size, and total sectors converted.

The imda and imdu also add information to the log file (JADEDD.TXT)

IMDA finds:

```
IMageDisk Utility 1.18 / Mar 07 2012
IMD 1.10: 4/02/2006 11:47:02
```

CP/M version 2.2

Jade Double-D controller Serial# 2-187-01170

I/O unknown

Track 1 on this disk contains 50 128 byte DD sectors.

128 byte DD is not supported by the 765, and very few

PCs can read/write this track.

Required cylinders: 77

Required heads : 1

Data rate : 500kbps

Est. maximum track: 10735 bytes

Possible drives/options to write JADEDD.IMD :

3.5" HD 80-track NOTE: *1

Double-step: OFF

*1 77 track image likely from 8" drive.

IMDU creates a .RAW (Sector Dump) Image and shows more information about the image and sectors.

```
0/0 500 kbps SD 26x128
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
D D DE5D D D D D D D D00DE5D D D D D D D DE5DE5DE5DE5DE5
1/0 500 kbps DD 50x128
1 11 21 31 41 2 12 22 32 42 3 13 23 33 43 4 14 24 34 44 5 15 25 35 45 6 16 26
36 46 7 17 27 37 47 8 18 28 38 48 9 19 29 39 49 10 20 30 40 50
DE5D D D D D D D D D D D D D D D D D D D D D D DE5D D D D
DE5D D D D DE5D D D D DE5D D D D DE5
2/0 500 kbps SD 26x128
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
D DE5DE5D00D D D DE5DE5D00D D D DE5DE5D00D D D D DE5D00D D D D
3/0 D00D D D D D D00D D D D00D D00D D D D D D00D D D D D D
4/0 D D D D D D D D D D D D D D D D D D D D D
5/0 D D D D D00D D DE5D D D D D00DE5D D D D D00DE5D00D D D D00DE5
6/0 D D D D D D D D D D D D D D D D D D D D D
7/0 D D D D D D D D D D D D D D D D D D D D D
8/0 D D D D D DE5D D D D D DE5D D D D00D D D DE5D D00D D D DE5
9/0 D00D00D00D D00D D00D00D00D D00D D00D00D00D D00D D00D00D00D
D00D D00D
10/0 D D D D D00D D D D D D D D D D D D D D D D
11/0 D D D D D D D D D D D D D D D D D D D D D
12/0 D D D D D D D D D D D D D D D D D D D D D
13/0 D D D D D D D D D D D D D D D D D D D D D
14/0 D D D D D D D D D D D D D D D D D D D D D
15/0 D D D D DE5D D D D D D D D D D DE5D D D D DE5DE5D
16/0 D D D D D D D D D D D D D D D D D D D D D
```

[illegible]

So, now you know the first two tracks are different size, and the remainder of the Image is written at 500 kbps, is SD, and Number of sectors is 26 and they are 128 bytes, with 77 Tracks on side 0.

New CPMTOOLS definitions require the following stanzas:

It could be one of the JADE Definitions or another one, depending on the hardware on the system that wrote the Image.

```
#libdisk Definition
# libdisk data below
[jad1]
description = JAD1 Jade - SSDD 8" - 128 x 48
cylinders = 77
heads = 1
seclen = 128
sectors = 48
seclen = 1
datarate = ED
```

It could also be accessed with following Floppy Definition.

#cpmtools Definition

A1 Generic CP/M – SSSD 8" – 128 x 26

```
diskdef a1
seclen 128
tracks 77
sectrk 26
blocksize 1024
maxdir 64
skew 6
boottrk 2
os 2.2
end
```

#libdsk Definition

libdsk data below

```
[a1]
description = A1 Generic CP/M – SSSD 8" – 128 x 26
cylinders = 77
heads = 1
seclen = 128
sectors = 26
seclen = 1
datarate = HD
```

Or it may be another 8" – 26 x 128 Definition. More investigation and testing is required.

Using Hexedit in a Linux Terminal (Command Window) we can look at the HEX Data in JAEDDD.RAW, and locate the directory entries. The first entry is located at 0x2600. 0x2600 is 9728 Decimal.

```
00002600 00 4D 4F 56 43 50 4D 20 20 43 4F 4D 00 00 00 4C .MOVCPM COM...L
00002610 02 03 04 05 06 07 08 09 0A 0B 00 00 00 00 00 00 .....
00002620 00 53 55 42 4D 49 54 20 20 43 4F 4D 00 00 00 0A .SUBMIT COM....
00002630 0C 0D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00002640 00 58 53 55 42 20 20 20 20 43 4F 4D 00 00 00 06 .XSUB COM....
00002650 0E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00002660 00 45 44 20 20 20 20 20 43 4F 4D 00 00 00 34 .ED COM...4
00002670 0F 10 11 12 13 14 15 00 00 00 00 00 00 00 00 00 .....
```

cpmtools sometimes has problems with the first two tracks being a mixed variety of single & double density.

A way to skip the first two tracks is to define where the offset of the Directory is located. We already know this is 9729. So, the following stanzas can be modified to point cpmtools at the Directory information:

```
..
..
skew 1
offset 9728
boottrk 0
```

```
# boottrk 2
os 2.2
end
```

After editing the diskdefs file for cpmtools, you have to also copy the updated diskdefs file to

/usr/local/share/diskdefs (in Linux)

```
$sudo cp ~/path/to/cpmtools/diskdefs /usr/local/share/diskdefs
```

Now, you can try to access the directory listing from the Terminal.

```
$cpmls -f a1 -D JADED.D.RAW
```

```
$cpmls -f a1 -li JADED.D.RAW
```

```
$cpmls -f a1 -D JADED.D.RAW
```

```
Name Bytes Recs Attr update create
```

```
-----
ASM .COM 8K 64
BIOS .ASM 12K 96
BIOSGEN .COM 1K 6
BIOSLDER.ASM 6K 46
DCM2 .ASM 49K 391
DCM2 .HEX 3K 19
DCMGEN .COM 1K 6
DDBIOS .ASM 21K 168
DDBOOT .ASM 25K 200
DDT .COM 5K 38
DEBLOCK .ASM 10K 80
DISKDEF .LIB 7K 49
DUMP .ASM 5K 33
DUMP .COM 1K 4
ED .COM 7K 52
FORMAT .ASM 34K 268
FORMAT .COM 3K 22
LOAD .COM 2K 14
MOVCPM .COM 10K 76
OLDSYS .COM 1K 8
PIP .COM 8K 58
STAT .COM 6K 41
SUBMIT .COM 2K 10
SYSGEN .COM 1K 6
XSUB .COM 1K 6
```

```
25 Files occupying 229K, 19K Free.
```

```
$cpmls -f a1 -li JADED.D.RAW
```

```
0:
4 -rwxrwxrwx 8192 Dec 31 1969 asm.com
11 -rw-rw-rw- 12288 Dec 31 1969 bios.asm
16 -rwxrwxrwx 768 Dec 31 1969 biosgen.com
19 -rw-rw-rw- 5888 Dec 31 1969 bioslder.asm
20 -rw-rw-rw- 50048 Dec 31 1969 dcm2.asm
29 -rw-rw-rw- 2432 Dec 31 1969 dcm2.hex
17 -rwxrwxrwx 768 Dec 31 1969 dcmgen.com
24 -rw-rw-rw- 21504 Dec 31 1969 ddbios.asm
12 -rw-rw-rw- 25600 Dec 31 1969 ddboot.asm
5 -rwxrwxrwx 4864 Dec 31 1969 ddt.com
```

```

14 -rw-rw-rw- 10240 Dec 31 1969 deblock.asm
15 -rw-rw-rw- 6272 Dec 31 1969 diskdef.lib
10 -rw-rw-rw- 4224 Dec 31 1969 dump.asm
9 -rwxrwxrwx 512 Dec 31 1969 dump.com
3 -rwxrwxrwx 6656 Dec 31 1969 ed.com
26 -rw-rw-rw- 34304 Dec 31 1969 format.asm
30 -rwxrwxrwx 2816 Dec 31 1969 format.com
6 -rwxrwxrwx 1792 Dec 31 1969 load.com
0 -rwxrwxrwx 9728 Dec 31 1969 movcpm.com
8 -rwxrwxrwx 1024 Dec 31 1969 oldsys.com
31 -rwxrwxrwx 7424 Dec 31 1969 pip.com
7 -rwxrwxrwx 5248 Dec 31 1969 stat.com
1 -rwxrwxrwx 1280 Dec 31 1969 submit.com
18 -rwxrwxrwx 768 Dec 31 1969 sysgen.com
2 -rwxrwxrwx 768 Dec 31 1969 xsub.com

```

This gives the listing in order of the directory entries, which is nice to have.

The next thing to do is to extract a .TXT, .ASM, .M80, or .SUB file to see if they are SANE.

If the definition isn't correct, there will be all kinds of mixed data in the ASCII extracted file.

More information on the cpm_{ls} command can be found with this command:

```
$man cpmls
```

```
$man cpmcp
```

I chose bios_{lder}.asm to extract with cpm_{ls}: (be sure to notice the PERIOD at the end of some commands)

```
$pwd
```

```
$~/IMDs/jadedd
```

```
$cpmcp -f a1 JADEDD.RAW 0:bioslder.asm .
```

bios_{lder}.asm contains:

```

NAME DBSLDR
TITLE - JADE DOUBLE D - CP/M 2.2
;*****
;
; *
; PROGRAM ID: DDBIOS LOADER *
; *
; VERSION: 2.2 RELEASE 2 *
; *
;*****
;
; *
; PROPERTY OF: JADE COMPUTER PRODUCTS *
; 4901 W. ROSECRANS BLVD. *
; HAWTHORNE, CALIFORNIA *
; 90250, U.S.A. *
; *
;*****
; THE BIOS LOADER IS READ INTO THE DCM SECTOR BUFFER *
; AFTER DCM HAS INITIALIZED. THE BIOS LOADER PROGRAM *
; IS THEN EXECUTED WHICH READS THE DDBIOS MODULE *
; INTO BANK 1. THE COMMAND BLOCK (IN DCM) IS SET TO *
; INDICATE DDBIOS MODULE SIZE AND THE SYSTEM LOAD *
; ADDRESS. THE BIOS LOADER PROGRAM IS GENERATED BY *
; MOVCPM.COM AS THE COLD START LOADER (900-97F HEX). *
; THIS MODULE IS PROVIDED FOR REFERENCE PURPOSES. *

```

```

, THIS MODULE IS PROVIDED FOR REFERENCE PURPOSES.
;*****
; THE DDBIOS LOADER IS COMPATABLE WITH REV B AND C *
; DOUBLE D CONTROLLER BOARDS. IT IS COMPATABLE WITH *
; FD1791-01 / FD1793-01. IT WILL ALSO FUNCTION WITH *
; THE CURRENT FD179X-02 SERIES. *
;*****
EJECT
;*****
; CONTROLLER PORT ASSIGNMENTS *
;*****

BL$STS EQU 000H ;BOARD STATUS
BL$CTL EQU 000H ;BOARD CONTROLS
WD$CMD EQU 004H ;179X-02 COMMAND REGISTER
WD$STS EQU 004H ;179X-02 STATUS REGISTER
WD$SEC EQU 006H ;179X-02 SECTOR REGISTER
WD$DTA EQU 007H ;179X-02 DATA REGISTER
XP$MTO EQU 010H ;MOTOR TIME OUT
XP$MTX EQU 040H ;MOTOR TIME EXTEND
XP$DSH EQU 080H ;DATA SYNC HOLD

;*****
; 179X-02 COMMAND AND MASK. *
;*****

DC$RDS EQU 10001000B ;READ SECTOR.
DM$RER EQU 10011100B ;READ ERROR MASK.

;*****
; SYSTEM ASSIGNMENTS *
;*****

NMBR$K EQU 20 ;SYSTEM SIZE IN K.
LNG$1K EQU 1024 ;TOTAL BYTES IN 1K.
CPM$SZ EQU NMBR$K*LNG$1K ;TOTAL SYSTEM BYTES.
BIOS$S EQU LNG$1K*3/2 ;BIOS ALLOCATED SIZE.
BIOS$A EQU CPM$SZ-BIOS$S ;BIOS LOAD ADDRESS.

;*****
; INTERNAL MEMORY ASSIGNMENTS *
;*****

BANK$0 EQU 1000H ;LOWER BANK ADDRESS.
BANK$L EQU 0400H ;1K BANK LENGTH.
BANK$1 EQU BANK$0+BANK$L ;UPPER BANK ADDRESS.
IO$BLK EQU BANK$0+0370H ;I/O BLOCK ADDRESS.
CB$STS EQU IO$BLK+0007H ;COMMAND STATUS BYTE.
CW$LAD EQU IO$BLK+0008H ;BIOS LOAD ADDR LOC.
CW$LNG EQU IO$BLK+000AH ;BIOS LOAD LENGTH LOC.
SEC$BF EQU BANK$0+0380H ;SECTOR BUFFER AREA.

;*****
; BIOS PROGRAM LINKAGE. *
;*****

SEC$BG EQU 4 ;FIRST BIOS SECTOR.

```

```

SEC$NM EQU 8 ;NUMBER OF SECTORS.
SEC$EX EQU SEC$BG+SEC$NM-1 ;LAST BIOS SECTOR.

;*****
;
EJECT
;*****
; ASSEMBLER DIRECTIVES *
;*****

PSECT ABS ;ABSOLUTE ADDRESSING.
ORG SEC$BF ;PROGRAM START POINT.

;*****
;
; INITIALIZE BIOS READ OPERATION *
;*****

BEGIN: LD HL,LNG$1K ;BIOS LOAD LENGTH.
LD (CW$LNG),HL ;LOAD LENGTH SET.
LD HL,BIOS$A ;BIOS SYSTEM ADDR.
LD (CW$LAD),HL ;LOAD ADDRESS SET.
LD HL,BANK$1 ;BIOS LOAD POINT.

;*****
;
; SET-UP FOR EACH READ SECTOR COMMAND *
;*****

RD$SEC: LD IY,RD$TST ;SET NMI VECTOR.
LD A,(SECTOR) ;FIRST BIOS SECTOR.
XOR C ;INVERT (1791-01).
OUT (WD$SEC),A ;SET 179X-02 SEC REG.
LD A,DC$RDS ;READ SECTOR CMND.
XOR C ;INVERT (1791-01).
OUT (WD$CMD),A ;ISSUE 179X-02 COMMAND.

;*****
;
; READ SECTOR OPERATION *
;*****

RD$BYT: IN A,(XP$DSH) ;WAIT FOR DATA.
IN A,(WD$DTA) ;INPUT INV DATA.
XOR C ;INVERT (1791-01).
LD (HL),A ;STORE DCM BYTE.
INC HL ;INCREMENT POINTER.
JP RD$BYT ;REPEAT OPERATION.

;*****
;
EJECT
;*****
; CHECK READ SECTOR STATUS, REPEAT UNTIL BIOS LOADED *
;*****

RD$TST: AND DM$RER ;TEST FOR ERRORS.
JR NZ,ERRORS ;ERROR DETECTED.

LD A,(SECTOR) ;GET SECTOR NMBR.
CP SEC$EX ;CHECK IF LAST SEC.
JR Z,FINISH ;GO IF FINISHED.
INC A ;INCRFMFNT

```

```

LD (SECTOR),A ;STORE SECTOR NUMBER.
JR RD$SEC ;READ NEXT SECTOR.

;*****
; READ ERROR HAS BEEN DETECTED *
;*****

ERRORS: LD (CB$STS),A ;DISPLAY ERROR STATUS.
XOR A ;ZERO A REGISTER.
OUT (BL$CTL),A ;DESELECT DRIVE.
IN A,(XP$MTO) ;MOTOR OFF!
HALT ;TERMINATE.

;*****
; BIOS SECTOR HAVE BEEN LOADED *
;*****

FINISH: EI ;ENABLE INTERRUPTS.
HALT ;SHUTDOWN BOARD.

;*****
; SECTOR NUMBER STORAGE *
;*****

SECTOR: DEFB SEC$BG ;SECTOR COUNTER.

;*****
END

```

Which looks SANE. So, now all .asm files need to be extracted.

```
$cpmcp -f a1 JAEDEDD.RAW 0:*.asm .
```

All the ASCII files also appear SANE.

```
$cpmcp -f a1 JAEDEDD.RAW 0:*.asm .
```

```
$cpmcp -f a1 JAEDEDD.RAW 1:*.asm .
```

```
$cpmcp -f a1 JAEDEDD.RAW 2:*.asm .
```

Another way you can access the data is to modify the HEX DATA so it is in a standard layout. In the HEX bytes below it's easy to see that Track 1, Side 0 is an odd format for the floppy. Those two tracks are the boot information, and we really don't need that, because we are after the data on the Images versus the boot information.

```

0/0 500 kbps SD 26x128
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
D D DE5D D D D D D D D00DE5D D D D D D D DE5DE5DE5DE5DE5DE5
1/0 500 kbps DD 50x128
1 11 21 31 41 2 12 22 32 42 3 13 23 33 43 4 14 24 34 44 5 15 25 35 45 6 16 26
36 46 7 17 27 37 47 8 18 28 38 48 9 19 29 39 49 10 20 30 40 50
DE5D D D D D D D D D D D D D D D D D D D D D D DE5D D D D
DE5D D D D DE5D D D D DE5D D D D DE5
2/0 500 kbps SD 26x128
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
D DE5DE5D00D D D DE5DE5D00D D D DE5DE5D00D D D D DE5D00D D D D

```

Copying the information we need to work with gets:

```

0/0 500 kbps SD 26x128 = 3328 = 0x0D00 = Track 0, Side 0 = 3328

1/0 500 kbps DD 50x128 = 6400 = 0x1900 = Track 1, Side 0 = 3328 + 6400 =
9728 = 0x2600 for start of Track 2

2/0 500 kbps SD 26x128 = 3328 = 0x0D00 = Track 2, Side 0

```

What we want to do is remove some of the HEX bytes to get Track 1, Side 0 to be 3328 in size, to match Track 0, Side 0.

$3328 + 3328 = 6656 = 0x1A00$
 $0x2600 - 0x1A00 = 0xC00 = 3072$ Bytes need to be removed from the Image.

The Linux dd command can easily edit/splice files.

`$dd if=JADEDD.RAW skip=03072 count=256256 bs=1 of=LKJADEDD.RAW conv=not`
 Now, LKJADEDD.RAW has Track 0, Side 0 and Track 1, Side 0 the same size, but the file length is reduced.

That isn't a problem as we are after the data files in the Image. We just need to refine the A1 definition

to point at the directory location.

`$hexedit LKJADEDD.RAW` shows the Directory at `0x1A00 = 6656`

```

00001A00 00 4D 4F 56 43 50 4D 20 20 43 4F 4D 00 00 00 4C .MOVCPM COM...L
00001A10 02 03 04 05 06 07 08 09 0A 0B 00 00 00 00 00 .....
00001A20 00 53 55 42 4D 49 54 20 20 43 4F 4D 00 00 00 0A .SUBMIT COM....
00001A30 0C 0D 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00001A40 00 58 53 55 42 20 20 20 20 43 4F 4D 00 00 00 06 .XSUB COM....
00001A50 0E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00001A60 00 45 44 20 20 20 20 20 20 43 4F 4D 00 00 00 34 .ED COM...4
00001A70 0F 10 11 12 13 14 15 00 00 00 00 00 00 00 00 .....

```

#Back to original definition or a modified definition:

#cpmtools Definition

A1 Generic CP/M – SSSD 8" – 128 x 26

diskdef a1

seclen 128

tracks 77

sectrk 26

blocksize 1024

maxdir 64

skew 6

boottrk 2

os 2.2

end

#cpmtools Definition

A1 Generic CP/M – SSSD 8" – 128 x 26

diskdef a1

seclen 128

tracks 77

sectrk 26

blocksize 1024

maxdir 64

skew 6

```
offset 6656
boottrk 0
os 2.2
end
```

```
$cpmls -f a1 -li LKJADEDD.RAW
```

```
0:
4 -rwxrwxrwx 8192 Dec 31 1969 asm.com
11 -rw-rw-rw- 12288 Dec 31 1969 bios.asm
16 -rwxrwxrwx 768 Dec 31 1969 biosgen.com
19 -rw-rw-rw- 5888 Dec 31 1969 bioslder.asm
20 -rw-rw-rw- 50048 Dec 31 1969 dcm2.asm
29 -rw-rw-rw- 2432 Dec 31 1969 dcm2.hex
17 -rwxrwxrwx 768 Dec 31 1969 dcmgen.com
24 -rw-rw-rw- 21504 Dec 31 1969 ddbios.asm
12 -rw-rw-rw- 25600 Dec 31 1969 ddboot.asm
5 -rwxrwxrwx 4864 Dec 31 1969 ddt.com
14 -rw-rw-rw- 10240 Dec 31 1969 deblock.asm
15 -rw-rw-rw- 6272 Dec 31 1969 diskdef.lib
10 -rw-rw-rw- 4224 Dec 31 1969 dump.asm
9 -rwxrwxrwx 512 Dec 31 1969 dump.com
3 -rwxrwxrwx 6656 Dec 31 1969 ed.com
26 -rw-rw-rw- 34304 Dec 31 1969 format.asm
30 -rwxrwxrwx 2816 Dec 31 1969 format.com
6 -rwxrwxrwx 1792 Dec 31 1969 load.com
0 -rwxrwxrwx 9728 Dec 31 1969 movcpm.com
8 -rwxrwxrwx 1024 Dec 31 1969 oldsys.com
31 -rwxrwxrwx 7424 Dec 31 1969 pip.com
7 -rwxrwxrwx 5248 Dec 31 1969 stat.com
1 -rwxrwxrwx 1280 Dec 31 1969 submit.com
18 -rwxrwxrwx 768 Dec 31 1969 sysgen.com
2 -rwxrwxrwx 768 Dec 31 1969 xsub.com
```

Thanks.

Larry

Re: [HOWTO] - Access CP/M Floppy's via libdisk & cpmtools

by **Ikraemer**

Posted: **2018-05-18 20:45**

Post by Ikraemer » 2018-05-18 16:00

UPDATE 05-18-2018

I've just downloaded some ACORN .DSD (.SSD) files, and cpmtools is having problems accessing the images. How I proceed?

How do I get a good Directory Listing, and extract good files?

```
cpm1.dsd
cpm2.dsd
cpm3.dsd
cpm4.dsd
cpm5.dsd
cpm6.dsd
cpm7.dsd
ciscobol.dsd
CPM Utilities Disk.dsd
```

```
CPM_UTILITIES_DISC.DSD
```

```
Basic_Program_Disc.dsd
```

```
CIS_Cobol_Program_Disc.dsd
```

The *.DSD (.SSD) images interleaved Cylinders and heads. CHS The proper way to convert these files is to use OMNIDISK or the ZIP file (CPMUTILS.ZIP) to change the Interleave back to HCS.

I used my DOS Computer and these commands from within Omnidisk:

```
*DRIVE 0
```

```
*ORDER CHS
```

```
*IMAGE CPM1.DSD CHS
```

```
*BACKUP * CPM1.IMG HCS
```

I did this for all of these *.DSD files.

```
409600 Apr 1 2006 CIS_Cobol_Program_Disc.dsd (renamed to CISCOBOL.DSD)
409600 Apr 1 2006 Basic_Program_Disc.dsd (renamed to BASICPRG.DSD)
409600 Apr 1 2006 CPM_Utilites_Disc.dsd (renamed to UTILS.DSD)
409600 Apr 1 2006 cpm7.dsd
409600 Apr 1 2006 cpm6.dsd
409600 Apr 1 2006 cpm5.dsd
409600 Apr 1 2006 cpm4.dsd
409600 Apr 1 2006 cpm3.dsd
409600 Apr 1 2006 cpm2.dsd
409600 Apr 1 2006 cpm1.dsd
```

```
409600 May 8 05:01 BASICPRG.IMG
409600 May 8 05:00 CISCOBOL.IMG
409600 May 8 04:59 UTILS.IMG
409600 May 8 04:59 CPM7.IMG
409600 May 8 04:58 CPM6.IMG
409600 May 8 04:58 CPM5.IMG
409600 May 8 04:57 CPM4.IMG
409600 May 8 04:56 CPM3.IMG
409600 May 8 04:56 CPM2.IMG
409600 May 8 04:55 CPM1.IMG
```

Now, the CPM1.IMG compares EXACTLY with the old Acorn CPM1.IMG file I had saved on my Hard drive.

But, I can't get cpmtools to extract a proper file. I've looked at my Definitions and tried lots of different settings.

```
# ACN1 Acorn CP/M on Acorn Z80 second processor. - 256 x 10
```

```
diskdef acn1
  seclen 256
  tracks 160
  sectrk 10
  secbase 0
  blocksize 2048
  maxdir 128
  skew 1
  boottrk 3
  os 2.2
end
```

```
# libdisk data below
```

```
lacn1]
description = ACN1  Acorn CP/M on Acorn Z80 second processor. - 256 x 1
sides = outback
cylinders = 160
heads = 2
seclen = 256
sectors = 10
seclen = 0
datarate = SD

# ACN2  Acorn CP/M on Acorn Z80 second processor (Alternate). - 256 x 1
diskdef acn2
    seclen 256
    tracks 160
    sectrk 10
    seclen 0
    blocksize 2048
    maxdir 128
    skew 1
    boottrk 3
    os 2.2
end

# libdsk data below
[acn2]
description = ACN2  Acorn CP/M on Acorn Z80 second processor (Alternate)
sides = outback
cylinders = 160
heads = 2
seclen = 256
sectors = 10
seclen = 0
datarate = SD

# ACN3  Sloger DDCPM on Acorn Z80 second processor. - 512 x 10
diskdef acn3
    seclen 512
    tracks 160
    sectrk 10
    seclen 0
    blocksize 4096
    maxdir 256
    skew 2
    boottrk 3
    os 2.2
end

# libdsk data below
[acn3]
description = ACN3  Sloger DDCPM on Acorn Z80 second processor. - 512 x
sides = outback
cylinders = 160
heads = 2
seclen = 512
sectors = 10
seclen = 0
datarate = DD
```

```
# ACN4 Sloger DDCPM on Acorn Z80 second processor (alternate). - 512 x
diskdef acn4
  seclen 512
  tracks 160
  sectrk 10
  secbase 0
  blocksize 4096
  maxdir 256
  skew 2
  boottrk 3
  os 2.2
end
```

```
# libdsk data below
[acn4]
description = ACN4 Sloger DDCPM on Acorn Z80 second processor (alterna
sides = outback
cylinders = 160
heads = 2
seclen = 512
sectors = 10
secbase = 0
datarate = DD
acornsd Definition has to be modified for location of directory:
diskdef acornsd
  seclen 256
  tracks 160
  sides outback      #= Order of Cylinders
  sectrk 10
  blocksize 2048
  maxdir 128
  datarate DD
  fm YES
  skewtab 0,1,4,5,8,9,2,3,6,7
  secbase 0
  #skew 3
  #0x1E00 is where directory is located. 7680 offset to directory.
  offset 7680
  boottrk 0
                                     #= 2, 2.2, or 3 (NO comment on next line)
  os 2.2
end
```

I can get a good directory listing.
 cpmfs -f acornsd -D CISC0B0L.IMG

```
Name Bytes Recs Attr update create
-----
$ANIM .V45 14K 97
CALL .ASM 8K 61
CALL .HEX 2K 3
CALL .PRN 12K 85
COBOL .COM 38K 295
COBOL .I01 8K 61
COBOL .I02 14K 106
COBOL .I03 8K 50
```

```

COBOL .MSG 8K 58
COBOL .I04 4K 30
COBOL .MSG 8K 58
CONFIG .COM 34K 269
FILEMARK.COM 2K 12
FORMS2 .CH1 2K 1
FORMS2 .CH2 2K 4
FORMS2 .COM 40K 315
FORMS2 .GN1 2K 4
FORMS2 .GN2 8K 55
FORMS2 .H01 2K 8
FORMS2 .H02 2K 9
FORMS2 .H03 2K 9
FORMS2 .H04 2K 9

FORMS2 .I01 2K 4
FORMS2 .I02 2K 6
FORMS2 .I51 2K 16
FORMS2 .I52 6K 47
FORMS2 .I53 6K 42
FORMS2 .W01 2K 7
FORMS2 .W02 2K 9
PI .CBL 4K 19
RUNA .COM 36K 275
STOCK1 .CBL 4K 17
STOCK2 .CBL 6K 37
32 Files occupying 286K, 110K Free.

```

I extracted the STOCK1.CBL with:

```
$cpmcp -f acornsd CISC0B0L.IMG 0:STOCK1.CBL /home/larry/IMDs/acorn/STOCK1.CBL
```

contains:

```

000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID. STOCK-FILE-SET-UP.
000030 AUTHOR. MICRO FOCUS LTD.
000040 ENVIRONMENT DIVISION.
000050 CONFIGURATION SECTION.
000060 SOURCE-COMPUTER.
000070 OBJECT-COMPUTER.
000075 SPECIAL-NAMES. CONSOLE IS CRT.
000080 INPUT-OUTPUT SECTION.
000090 FILE-CONTROL.
000100 SELECT STOCK-FILE ASSIGN "STOCK.IT"
000110 ORGANIZATION INDEXED
000120 ACCESS DYNAMIC
000130 RECORD KEY STOCK-CODE.
000140 DATA DIVISION.
000150 FILE SECTION.
000160 FD STOCK-FILE; RECORD 32.
000170 01 STOCK-ITEM.
000180 02 STOCK-CODE PIC X(4).
000190 02 PRODUCT-DESC PIC X(20).
000200 02 UNIT-SIZE PIC 9(4).
000210 WORKING-STORAGE SECTION.
000220 01 SCREEN-HEADINGS.
000230 02 ASK-CODE PIC X(21) VALUE "STOCK CODE < >".
000240 02 FILLER PIC X(59).
000250 02 ASK-DESC PIC X(16) VALUE "DESCRIPTION <".

```

```

000260 02 SI-DESC PIC X(21) VALUE " >".
000270 02 FILLER PIC X(43).
000280 02 ASK-SIZE PIC X(21) VALUE "UNIT SIZE < >".
000290 01 ENTER-IT REDEFINES SCREEN-HEADINGS.
000300 02 FILLER PIC X(16).
000310 02 CRT-STOCK-CODE PIC X(4).
000320 02 FILLER PIC X(76).
000330 02 CRT-PROD-DESC PIC X(20).
000340 02 FILLER PIC X(60).
000350 02 CRT-UNIT-SIZE PIC 9(4).
000360 02 FILLER PIC X.
000370 PROCEDURE DIVISION.
000380 SR1.
000390 DISPLAY SPACE.
000400 OPEN I-O STOCK-FILE.
000410 DISPLAY SCREEN-HEADINGS.
000420 NORMAL-INPUT.
000430 MOVE SPACE TO ENTER-IT.
000440 DISPLAY ENTER-IT.
000450 CORRECT-ERROR.
000460 ACCEPT ENTER-IT.
000470 IF CRT-STOCK-CODE = SPACE GO TO END-IT.
000480 IF CRT-UNIT-SIZE NOT NUMERIC GO TO CORRECT-ERROR.
000490 MOVE CRT-PROD-DESC TO PRODUCT-DESC.
000500 MOVE CRT-UNIT-SIZE TO UNIT-SIZE.
000510 MOVE CRT-STOCK-CODE TO STOCK-CODE.
000520 WRITE STOCK-ITEM; INVALID GO TO CORRECT-ERROR.
000530 GO TO NORMAL-INPUT.
000540 END-IT.
000550 CLOSE STOCK-FILE.
000560 DISPLAY SPACE.
000570 DISPLAY "END OF PROGRAM".
000580 STOP RUN.

```

The cpmutils.zip will also do the same job. I can't upload it because of the Forum Quota.

Larry

Re: [HOWTO] - Access CP/M Floppy's via libdisk & cpmtools

by **Ikraemer**

Posted: **2018-06-07 14:34**

Post by Ikraemer » 2018-06-07 16:09

UPDATE 06-07-2018

I've just located a good URL for TIKI .dsk Images.

These disk images are in the .dsk format and were created with ANADISK for MS-DOS by using the sector dump feature to dump the sectors. The .dsk files I've downloaded are:

```

204800 Jan 25 2001 kpm11.dsk
204800 Jan 10 2001 kpm115.dsk
204800 Sep 14 2003 kpm117.dsk
204800 Jan 28 2001 tik200.dsk

```

```

204800 Jun 28 2001 tiko200r.dsk
409600 Aug 26 2000 tkjer401.dsk

819200 Jan 10 2001 wcsys.dsk
204800 Jun 2 19:51 wordstar.dsk
409600 Jun 2 19:50 Kryssord_Generator.dsk
204800 Jun 2 19:43 kommunikasjonsprogram.dsk
409600 Jun 2 09:59 kermit.dsk
409600 Jun 2 09:58 Turbo_Pascal_v2_00a.dsk
409600 Jun 2 09:58 dbase_II_v2_4.dsk
409600 Jun 2 09:57 Eksempler_(Tiki-Comal_Basic).dsk
409600 Jan 28 2001 tiko208r-400k.dsk
204800 Jan 28 2001 tiko208r-200k.dsk

```

I tried RAWRITE and other utilities, but couldn't get the Images written with my DOS 7.1 Computer. So, I tried creating IMD files with the Imagedisk Utility BIN2IMD from DOSBox on my Linux Laptop.

REF: [http://www.vcfed.org/forum/showthread.p ... ages/page2](http://www.vcfed.org/forum/showthread.p...ages/page2)

RAW to .IMD format:

You can convert raw sector images to ImageDisk using the following commands:

```

160k raw to imd: (163,840 bytes)
BIN2IMD <image.img> <image.imd> DM=5 N=40 SS=512 SM=1-8 /1

180k raw to imd: (184,320 bytes)
BIN2IMD <image.img> <image.imd> DM=5 N=40 SS=512 SM=1-9 /1

320k raw to imd: (327,680 bytes)
BIN2IMD <image.img> <image.imd> DM=5 N=40 SS=512 SM=1-8 /2

360k raw to imd: (368,640 bytes)
BIN2IMD <image.img> <image.imd> DM=5 N=40 SS=512 SM=1-9 /2

1.2mb raw to imd: (1,228,800 bytes)
BIN2IMD <image.img> <image.imd> DM=3 N=80 SS=512 SM=1-15 /2

720k (3.5") raw to imd: (737,280 bytes)
BIN2IMD <image.img> <image.imd> DM=5 N=80 SS=512 SM=1-9 /2

1.44mb raw to imd: (1,474,560 bytes)
BIN2IMD <image.img> <image.imd> DM=3 N=80 SS=512 SM=1-18 /2

```

Create some test *.IMD files to write with IMD:

```

BIN2IMD KPM11.DSK KPM11.IMD DM=5 N=40 SS=512 SM=1-10 /1
BIN2IMD KPM115.DSK KPM115.IMD DM=5 N=40 SS=512 SM=1-10 /1
BIN2IMD KPM117.DSK KPM115.IMD DM=5 N=40 SS=512 SM=1-10 /1
BIN2IMD TIK200.DSK TIK200.IMD DM=5 N=40 SS=512 SM=1-10 /1
BIN2IMD TIK400.DSK TIK400.IMD DM=5 N=40 SS=512 SM=1-10 /2
BIN2IMD TIK0S20.DSK TIK0S20.IMD DM=5 N=40 SS=512 SM=1-10 /2
BIN2IMD TKJER401.DSK TKJER401.IMD DM=5 N=40 SS=512 SM=1-10 /2
BIN2IMD WCSYS.DSK WCSYS.IMD DM=5 N=80 SS=512 SM=1-10 /2

```

Create .RAW Images with IMDU (to VERIFY with .DSK Images)

```

IMDU KPM11.IMD KPM11.RAW /B /E /D > KPM11.TXT
IMDU KPM115.IMD KPM115.RAW /B /E /D > KPM115.TXT
IMDU KPM117.IMD KPM115.RAW /B /E /D > KPM117.TXT

```

```
IMDU TIK200.IMD TIK200.RAW /B /E /D > TIK200.TXI
IMDU TIK400.IMD TIK400.RAW /B /E /D > TIK400.TXT
IMDU TIK0S20.IMD TIK0S20.RAW /B /E /D > TIK0S20.TXT
IMDU TKJER401.IMD TKJER401.RAW /B /E /D > TKJER401.TXT
IMDU WCSYS.IMD WCSYS.RAW /B /E /D > WCSYS.TXT
The KPM11.TXT file contains information on the Image:
```

IMageDisk Utility 1.18 / Mar 07 2012

IMD 1.18: 1/06/2018 18:02:59

Assuming 1:1 for Binary output

0/0 250 kbps DD 10x512

1 2 3 4 5 6 7 8 9 10

D D D D D D D D D D

1/0 D D D D D D D D D D

2/0 D D DE5 DE5 D DE5 D D D D

3/0 D D D D D D D D D D

4/0 D D D D D D D D D D

5/0 D D D D D D D DE5 D D

6/0 D D D D D D D D D D

7/0 D D D DE5 D D D D D D

8/0 D DE5 D D D D D D D D

9/0 D D D D D D D D D D

10/0 D D D D D D D D D D

11/0 D D D D D D D D D D

12/0 D D D D D DE5 D D D D

13/0 D D D D D D D D D D

14/0 D D D D D D D D D D

15/0 D D D D D D D D D D

16/0 D D D D D D D D D D

17/0 D D D D D D D D D D

18/0 D D D D D D D D D D

19/0 D D D DE5 D DE5 D D D D

20/0 D D D DE5 D D D D D D

21/0 D D D D D D D D D D

22/0 D D D D D D D D D D

23/0 D D D D D00 D D D D D

24/0 D D00 D00 D00 D D D00 D D D

25/0 D D D D D D00 D D D D00

26/0 D00 D00 D D D D00 D00 D00 D D

27/0 D D D D D D D D D D

28/0 D D D D D D D D D D

29/0 D D D D D D D D D D

30/0 D D D D D D D D00 D D00

31/0 D00 D00 D00 D00 D00 D00 D00 D00 D00

32/0 D00 D00 D00 D00 D00 D00 D00 D00 D00

33/0 D00 D00 D00 D00 D00 D00 D00 D00 D00

34/0 D00 D00 D00 D DFF DFF DFF DFF D DE5

35/0 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5

36/0 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5

37/0 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5

38/0 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5

39/0 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5

40 tracks(40/0), 400 sectors (112 Compressed)

Notice the 40 tracks and 10 Sectors of 512 Bytes.

```
$ls -alt *.RAW
204800 Jun  1 18:10 KPM11.RAW
204800 Jun  1 18:11 KPM115.RAW
204800 Jun  1 18:10 KPM117.RAW
204800 Jun  1 12:40 TIK200.RAW
409600 Jun  1 12:51 TIK400.RAW
409600 Jun  1 12:51 TIK0S20.RAW
409600 Jun  1 18:11 TKJER401.RAW
819200 Jun  1 18:12 WCSYS.RAW
```

The .RAW images can be accessed with cpmtools with the proper definitions.

New CPMTTOOLS definitions require the following information:

```
diskdef title
seclen xxx #= Sectors xx,1024
tracks xx #= (Cylinders * Sides) = 80*2 = 160
sectrk xx #= Sectors 5,xxx
blocksize xxxx #= (128*(BLM+1)) = 2048
maxdir xxx #= (DRM+1) = 256
skew x #= may be 1 thru 6, or so
boottrk x #= OFS = 2
os x.x #= 2.2, or 2, or 3
end
```

New LIBDSK definitions require the following information:

```
[title]
description = DESC The description of the format as shown by (for example) dskform-
help.
sidedness =TREATMENT How a double-sided disk is handled. This can either be alt
(sides alternate - used by most PC-hosted operating systems),
outback (use side 0 tracks 0-79, then side 1 tracks 79-0 - used
by 144FEAT CP/M disks), or outout (use side 0 tracks 0-79, then
side 1 tracks 0-79 - used by some Acorn formats). If the disk is
single-sided, this parameter can be omitted.
cylinders = COUNT Sets the number of cylinders (usually 40 or 80).
heads = COUNT Sets the number of heads (usually 1 or 2 for single- or double- sided).
sectors = COUNT Sets the number of sectors per track.
secbase = NUMBER Sets the first sector number on a track. Usually 1; some Acorn
formats use 0.
sectsize = COUNT Sets the size of a sector in bytes. This should be a power of 2.
datarate = VALUE Sets the rate at which the disk should be accessed. This is: HD, DD,
SD or ED.
rwgap = VALUE Sets the read/write gap.
fmtgap = VALUE Sets the format gap.
fm = Y or N Sets the recording mode - Y for FM, N for MFM.
multitrack = Y or N Sets multitrack mode.
skipdeleted = Y or N Sets whether to skip deleted data.
```

The LIBDSK Data rate will be one of:

```
RATE_HD, /* Rate for High-density disc - 1.2Mb in 5.25" 96 tpi drive, or 1.44Mb in 3.5"
96 tpi drive */
RATE_DD, /* Rate for Double-density disc - 360k in 5.25" 48 tpi drive, or 720K in 3.5"
48 tpi drive */
RATE_SD, /* Rate for Double-density disc - 180k in 5.25" 48 tpi drive, or 360k in 3.5"
48 tpi drive */
RATE_ED /* Data rate for 2.8Mb 3.5" in 3.5" 96 tpi drive */
```

The libdisk definitions must be inserted into the /home/user/ libdisk file and the

The libdisk definitions must be inserted into the /home/user/libdisk file, and the cpmtools diskdefs file.

UPDATED Definitions for cpmtools & libdisk:

NOTE TIKI-100 DEFINISJONER FOR 22DISK
 NOTE ASBJØRN DJUPDAL, JØRN E. HAUGAN (TIK4)
 NOTE TIK1 (90K) ER IKKE TESTET!
 NOTE 09-JAN-2001
 NOTE cpmtools & libdisk by LARRY KRAEMER
 NOTE 06-01-2018

```
BEGIN TIK1 TIKI-100, 90K - SSSD 48 tpi 5.25" - 128 x 18
DENSITY FM, LOW
CYLINDERS 40 SIDES 1 SECTORS 18,128 LABEL TIK1
SIDE1 0 1,6,11,16,3,8,13,18,5,10,15,2,7,12,17,4,9,14
BSH 3 BLM 7 EXM 0 DSM 82 DRM 31 AL0 080H AL1 0 OFS 3
END
```

```
# TIK1 TIKI-100 90K - SSSD 48 tpi 5.25" - 128 x 18
diskdef tik1
  seclen 128
  tracks 40
  sectrk 18
  blocksize 1024
  maxdir 32
  skew 5
  boottrk 3
  os 2.2
end
```

```
[tik1]
description = TIK1 TIKI-100 90K - SSSD 48 tpi 5.25" - 128 x 18
cylinders = 40
heads = 1
secsize = 128
sectors = 18
secbase = 1
datarate = SD
```

```
BEGIN TIK2 TIKI-100, 200K - SSDD 48 tpi 5.25" - 512 x 10
DENSITY MFM, LOW
CYLINDERS 40 SIDES 1 SECTORS 10,512 SKEW 2 LABEL TIK2
SIDE1 0 1,2,3,4,5,6,7,8,9,10
BSH 3 BLM 7 EXM 0 DSM 189 DRM 63 AL0 0C0H AL1 00H OFS 2
END
```

```
# TIK2 TIKI-100 200K - SSDD 48 tpi 5.25" - 512 x 10
diskdef tik2
  seclen 512
  tracks 40
  sectrk 10
  blocksize 1024
  maxdir 64
  skew 1
  boottrk 2
  os 2.2
end
```

```
[tik2]
description = TIK2  TIKI-100 200K - SSDD 48 tpi 5.25" - 512 x 10
cylinders = 40
heads = 1
seclen = 512
sectors = 10
secbase = 1
datarate = DD
```

```
BEGIN TIK3  TIKI-100, 400K - DSDD 48 tpi 5.25" - 512 x 10
DENSITY MFM, LOW
CYLINDERS 40 SIDES 2 SECTORS 10,512 SKEW 2 LABEL TIK3
SIDE1 0 1,2,3,4,5,6,7,8,9,10
SIDE2 1 1,2,3,4,5,6,7,8,9,10
ORDER SIDES
BSH 4 BLM 15 EXM 1 DSM 194 DRM 127 AL0 0C0H AL1 00H OFS 2
END
```

```
# TIK3  TIKI-100 400K - DSDD 48 tpi 5.25" - 512 x 10
diskdef tik3
    seclen 512
    tracks 80
    sectrk 10
    blocksize 2048
    maxdir 128
    skew 1
    boottrk 2
    os 2.2
end
```

```
[tik3]
description = TIK3  TIKI-100 400K - DSDD 48 tpi 5.25" - 512 x 10
sides = alt
cylinders = 80
heads = 2
seclen = 512
sectors = 10
secbase = 1
datarate = DD
```

```
BEGIN TIK4  TIKI-100, 800K - DSDD 96 tpi 5.25" - 512 x 10
DENSITY MFM, LOW
CYLINDERS 80 SIDES 2 SECTORS 10,512 SKEW 2 LABEL TIK4
SIDE1 0 1,2,3,4,5,6,7,8,9,10
SIDE2 1 1,2,3,4,5,6,7,8,9,10
ORDER SIDES
BSH 4 BLM 15 EXM 0 DSM 394 DRM 127 AL0 0C0H AL1 00H OFS 2
END
```

```
# TIK4  TIKI-100 800K - DSDD 96 tpi 5.25" - 512 x 10
diskdef tik4
    seclen 512
    tracks 160
    sectrk 10
    blocksize 2048
    maxdir 128
```

```
skew 1
boottrk 2
os 2.2
end
```

```
[tik4]
description = TIK4  TIKI-100 800K - DSDD 96 tpi 5.25" - 512 x 10
sides = alt
cylinders = 160
heads = 2
seccsize = 512
sectors = 10
secbase = 1
datarate = DD
Directory Listings Using the Ampro Definitions:
$cpmls -f amp1 KPM11.RAW
```

```
0:
120.com
160.com
40.com
80.com
80adm.com
demo.tik
edit.com
farge.tik
install.com
kat.com
kon.tik
kopi.com
kpmkopi.com
les.meg
nypla-ed.com
nyplate.com
py.tik
pyton.com
r]kop-ed.com
r]kopi.com
sett.com
test.tik
tiki.com
```

```
$cpmls -f amp1 KPM115.RAW
```

```
0:
120.com
120x.com
160.com
40.com
40x12.com
80.com
80adm.com
80inf.com
80mon.com
80x.com
demo.tik
edit.com
```

```
eur.com
install.com
kat.com
kon.tik
kopi.com
kpm11.sys
kpmkopi.com
les.meg
mon-130.com
nypla-ed.com
nyplate.com
par-revb.asm
par-revb.pdr
py.tik
python.com
r]kop-ed.com
r]kopi.com
sett.com
sjakk.com
test.tik
tiki.com
tit.com
utf\r.com
```

```
$cpmls -f amp1 KPM117.RAW
```

```
0:
120.com
160.com
160x.com
40.com
40x12.com
72i.com
80.com
80adm.com
80inf.com
80mon.com
80x.com
defterm.com
demo.tik
edit.com
install.com
kat.com
kon.tik
kopi.com
kpm117.sys
kpmkopi.com
les.meg
meny.com
meny.txt
menybig.txt
netp-p1.com
netp-p2.com
nyplate.com
plater.std
py.tik
python.com
```

```
r]kopi.com
sett.com
sjakk.com
terminal.com
test.tik
tiki.com
utf\r.com
```

```
$cpmls -f amp2 -d TKJER401.RAW
```

```
KOPI COM : KAT COM : 80 COM : 40 COM
40X12 COM : 72I COM : NYPLATE COM : R]KOPI COM
MENY COM : INSTALL COM : TDV COM : ! COM

PLATER BAK : 80ADM COM : 120 COM : 160 COM
80INF COM : 80MON COM : 80X COM : TDVDEF COM
PALETT COM : PYTON COM : PLATER STD : PALETT DAT
INFO TEX : ROBOT COM : SJAKK COM : TIKOKOPI COM
UTF\R COM : 160X COM : SETT COM : MENY TXT
EDIT COM : TIKO200B SYS : TIKOAB SYS : LES MEG
VIKTIG TXT : ROBOT DOC : SCANDIS COM : DYR FNT
IBM COM : LESMEG COM : BASIC TXT : ANIMAL BBB
FRAND2 BBB : ROTHUS BBB : FRANDO BBB : PIANO BBB
ENVELOPE BBB : ANIMAL DAT : FARGER BBB : UR BBB
SINUS BBB : SINC BBB : ICOS2 BBB : SOL BBB
FRAND1 BBB : SLAGER BBB : MERGE BBB : SINC2 BBB
ISINC BBB : SIRKEL BBB : FLATE2 BBB : DYR BBB
ASEMBLY BBB : FINDEX BBB : ADDRESS DTA : BASIC COM
UNLIST COM : NAME NDX : KAT COM
```

```
$cpmls -f amp2 -d WCSYS.RAW
```

```
KOPI COM : KAT COM : 80 COM : 40 COM
40X12 COM : 72I COM : NYPLATE COM : R]KOPI COM
MENY COM : INSTALL COM : PLATER STD : ! COM
MENY TXT : 80ADM COM : 120 COM : 160 COM
80INF COM : 80MON COM : 80X COM : TDV COM
PALETT COM : PYTON COM : MENY SYS : TDVDEF COM
PALETT DAT : INFO TEX : SJAKK COM : SCANDIS COM
TIKOKOPI COM : UTF\R COM : 160X COM : SETT COM
ROBOT COM : LES MEG : EDIT COM : WD200C SYS
IBM COM : WD400C SYS : ROBOT DOC : LESMEG COM
HARDLES TXT : DYR FNT : BASIC TXT : ANIMAL BBB
FRAND2 BBB : ROTHUS BBB : FRANDO BBB : PIANO BBB
ENVELOPE BBB : ANIMAL DAT : FARGER BBB : UR BBB
SINUS BBB : SINC BBB : ICOS2 BBB : SOL BBB
FRAND1 BBB : SLAGER BBB : MERGE BBB : SINC2 BBB
ISINC BBB : SIRKEL BBB : FLATE2 BBB : DYR BBB
ASEMBLY BBB : FINDEX BBB : ADDRESS DTA : BASIC COM
UNLIST COM : NAME NDX
```

Extract a file to verify it is SANE.

```
$cpmcp -f amp2 tikos20.dsk 0:VIKTIG.TXT .
```

```
KOPIERING AV 200K DISKETTER
```

```
De fleste programmer leveres på 200K disketter. Av praktiske grunner
```

De neste programmer leveres på 200K disketter. Vi anbefaler vi at disse kopieres over til 400K disketter.

For å kopiere fra 200K disketter til 400K disketter gjør en fram
på følgende måte:

Med en diskettstasjon:

- * Formater en ny 400K diskett ved hjelp av NYPLATE (Valg M i menyen).
- * Sett inn TIKO KJERNE disketten. Tast A for å avslutte menyen.
- * Skriv: Install tikoab.sys Trykk RETUR-tasten.
- * Velg G (oppdater hukommelsen) i INSTALL-menyen. Trykk RETUR to ganger.
- * Avslutt programmet med X
- * Når stasjonen midlertidig installert slik at maskinen når du har tegnet a>> venter å finne en 400K diskett, mens den når du har tegnet b>> på skjermen venter å finne en 200K diskett i diskettstasjonen.
- * Når skjermen viser klartegnet a>> skriver du: Kopi a:=b:*. * -W <RETUR>
- * Det er viktig å skrive -w til slutt. Det lar deg stoppe og bytte mellom 200k og 400K disketten når du kopierer.
- * Du blir spurt om å sette inn fraplate. Sett inn 200K disketten du skal kopiere fra og trykk <RETUR>
- * Sett inn den formaterte 400K disketten og trykk <RETUR> når du blir spurt om å sette inn tilplate.
- * Slik fortsetter du helt til disketten er ferdig kopiert.

Hvis du har to diskettstasjoner kan du bruke oppskriften i APPENDIX A2 i INSTRUKSJONSBOK for TIKI-100. Bytt ut ordet 800K med 400K.

Thanks.

Larry

Re: [HOWTO] - Access CP/M Floppy's via libdisk & cpmtools

by Ikraemer

Posted: 2019-11-21 13:21

Post by Ikraemer » 2019-11-21 07:00

UPDATE 11-21-2019

I've just located some Teletek .TD0 files at the following URL:

[http://www.retroarchive.org/maslin/disk ... index.html](http://www.retroarchive.org/maslin/disk...index.html)

Name	Size	Format	Description
tltk-cad.zip	90689	ZIP	No description available.
tsys-src.zip	44519	ZIP	Source code for CP/M-3.0 BIOS etc.
tsysmstr.td0	129320	SSDD	CP/M-2.2 system disk for Teletek Systeme
tsysmtrc.td0	136030	SSDD	/

tsysmtrc.zip 100388 5500 /
 There is a PDF of the Teletek Technical Manual at:
<https://www.hartetechnologies.com/manuals/TeleTek/>

After you have downloaded the *.TD0 files, what do you do with the *.TD0 files? How do you extract the files?
 How do you insert files into the *.TD0 file.

The Teletek Image had been created by TELEDISK, a program by SYDEX. The Version of TELEDISK is encoded in the first few bytes of the TD0 Image. You can view the first 16 bytes with hexedit to see the version. The fifth byte

determines what version was used. If it is 15 then version 1.5 or greater was used.
 00000000 74 64 00 4C 15 02 05 80 00 01 A1 13 66 FD 77 B1 td.L....

This .TD0 file can be made into a .IMD file by the IMAGEDISK utility TD02IMD. I use the IMAGEDISK utilities in Debian

Linux Ver 9.x - 64 Bit in DOSBox running on Debian. The command in Linux would be:
 \$td02imd tsysmstr.td0 > tsysmstr.txt

Gives the following information:

TD 1.5 8" HD MFM S-step, 1 sides ADV 2/03/1997 21:19:09
 CP/M-2.2 system disk for Teletek Systemmaster 64k
 256 byte sector, 1-26, 1:1
 77 tracks, 2002 sectors converted.

Now, you have the first clue as to what is in the .TD0 image, along with an .IMD file.

You now use the

tsysmstr.imd file to get more information.

\$imda tsysmstr.imd >> tsysmstr.txt

Which gives this information:

IMageDisk Analyzer 1.18 / Mar 12 2012

IMD TD 1.5 8" HD MFM S-step, 1 sides ADV 2/03/1997 21:19:09

CP/M-2.2 system disk for Teletek Systemmaster 64k

256 byte sector, 1-26, 1:1

Required cylinders: 77

Required heads : 1

Data rate : 500kbps

Est. maximum track: 8951 bytes

Possible drives/options to write TSYSMSTR.IMD :

3.5" HD 80-track NOTE: *1 *2
 Double-step: OFF

5.25" HD 80-track NOTE: *1
 Double-step: OFF

8" 77-track
 Double-step: OFF

*1 77 track image likely from 8" drive.

*2 Should fit on 360rpm drive, 300rpm drive will leave long end gap.
 It should be possible to write the image to your Floppy on a 5.25" 1.2 floppy or a 3.5" 80 track floppy.

You can now create the *.RAW file for accessing with cpmtools and libdisk

```
$imdu tsysmstr.imd tsysmstr.raw /B /E /D >> tsysmstr.txt
```

Which shows a lot more information about the image:

IMaGeDisk Utility 1.18 / Mar 07 2012

IMD 1.18: 15/11/2017 19:12:44

Balcones 59K CP/M Version 2.25 Teletex FDC-1

BNV-205 Mixed Density Version (200-85-4050)

M-10 Hard Disk (Ins -e M10.PRD EC00)

C1979 Digital Research C1983 Balcones Computer Corporation

Assuming 1:1 for Binary output

```
0/0 500 kbps SD 26x128 = 728 = 0x02D8 <-- First Track is SD 26x128
  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
  D D D D D D D D D D D D D D D D D D D D D D D00D00
1/0 500 kbps DD 26x256 = 1456 = 0x05B0
  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
  D D D D D D D D D D D D D D D D D D D D D D D
2/0 D DE5DE5D00D D D DE5DE5D D D00D DE5DE5D D D00D D DE5D
3/0 D D D D D D D D D D D D D D D D D D D D D D
4/0 D D D D D00D D D D D D D D D DE5D D D D D D D
5/0 D D D D D D D D D D DE5D D D DE5D DE5D D D DE5D DE5
6/0 D D D D DE5D D D D D DE5D D D D D DE5D DE5D D D
7/0 D DE5DE5D00D D00D D DE5D00D D00D D DE5D00D D00D D00DE5D
8/0 D D D D D D D D D D D D D D D D D D D D D
9/0 D DE5D D D D D DE5D D D D D DE5D D D D D DE5D DE5
10/0 D D D DE5D D D D D DE5D D D D D D D D D D D D
11/0 D DE5D D DE5DE5D D DE5DE5D DE5D D DE5DE5D DE5D DE5DE5DE5
12/0 DE5D D D D D DE5D D D D D DE5D D D D D DE5D D D
13/0 D D DE5DE5DE5D DE5D D DE5DE5DE5D D D DE5DE5DE5D D D DE5
14/0 D D D DE5D D D D D DE5D D00D D D D D D D D D D
15/0 D DE5D D D D D D D D D D D D D D D D D D DE5D
16/0 D DE5D DE5D DE5DE5DE5D DE5D DE5DE5DE5D DE5D D DE5DE5D DE5
17/0 DE5D DE5D DE5D DE5D D D DE5D D D D DE5DE5D DE5D D DE5
18/0 D D D D D D D DE5D D D D D DE5D D D D D D D D
19/0 D DE5DE5D D DE5D D DE5D D DE5D D DE5D DE5DE5D DE5DE5D
20/0 D D D D D D D D D D D D D D D D D00D D D D
21/0 D D D D D D D D D D D D D D D D D D D D D
22/0 D D D D D D D D D D D D D D D D D D D D D
23/0 D D D D D D D D D D D D D D D D D D D D D
24/0 D D D D D D D D D D D D D D D D D D D D D
25/0 D DE5D D D D D DE5D D D D D DE5D D D D D DE5DE5D
26/0 D D D DE5D DE5D D D DE5D DE5D D D D D DE5D D D D
27/0 D D D D D D D D D D D D D D D D D D D D D
28/0 DE5D D D D D DE5D D D U D DE5D D D D D DE5D D D
29/0 D D D D D D D D D D D U D D D D D D D D D D
30/0 D D D D D D D D D D D U D D D D D D D D D D
31/0 D D D D D D D D D D D U D D D D D D D D D D
32/0 D D D D D D D D D D D U D D D D D D D D D D
33/0 D D DE5D D D D D D D U D D D D D DE5D D D D D
34/0 D D D D D D D D D D D U D D D D D D D D D D
35/0 D D D D D D D D D D D U D D D D D D D D D D
36/0 D D D D D DE5D D D D U DE5D D D D D D D D D D
37/0 D D D D D D D D D D D U D D D D D D D D D D
38/0 D DE5D D D D DE5D D D D D D DE5DE5D D D D D DE5D
```

This drives cpmtools crazy, although cpmtools should be able to handle a SD Track 0. I've found that the easiest way to access the data is to replace the first SD track, with a DD Track. Or you can replace the first track with enough zero's to make the directory entries at the proper track 2 location (2 Boot Tracks as noted in the Definition).

[illegible]

```
The Teletek disk definition is:
# TEL1 Teletek - SSDD 8" - 256 x 26
diskdef tel1
    seclen 256
    tracks 77
    sectrk 26
    blocksize 2048
    maxdir 128
    skew 6
```

```

DOOTTRK 2
os 2.2
end

```

```

# libdsk data below
[tell]
description = TEL1 Teletek - SSDD 8" - 256 x 26
cylinders = 77
heads = 1
seccsize = 256
sectors = 26
secbase = 1
datarate = ED

```

These definitions need to be added to the appropriate cpmtools and libdsk definition files.

To get a directory listing with cpmtools use:

```
cpmls -f tell -D lktsysmstr.raw
```

Name	Bytes	Recs	Attr	update	create
-VERSION.B00	2K	1			
ASM.COM	8K	64			
BACKUP.COM	4K	19			
BACKUP.MAC	12K	81			
BRKPNT.MAC	10K	76			
CONSYS.COM	6K	35			
CONSYS.MAC	16K	120			
CPM64.COM	12K	86			
DDT.COM	6K	38			
DEBLOCK.ASM	10K	80			
DISKDEF.LIB	8K	49			
DUMP.ASM	6K	33			
DUMP.COM	2K	4			
ED.COM	8K	52			
EPROM.MAC	4K	20			
FORMAT.COM	4K	18			
FORMAT.MAC	10K	78			
LOAD.COM	2K	14			
MOVCPM.COM	12K	96			
PIP.COM	8K	58			
SMBIOS.MAC	4K	31			
SMBOOT.MOD	6K	42			
SMCP/M.DOC	24K	190			
SMCSL.MAC	6K	38			
SMDIOS.MOD	14K	102			
SMDISK.MOD	14K	111			
SMDVAR.MOD	6K	42			
SMINIT.MOD	8K	53			
SMJUMP.MOD	2K	16			
SMPIOS.MOD	8K	64			
SMVCTR.MOD	2K	12			
STAT.COM	6K	41			
SUBMIT.COM	2K	10			
SYSGEN.COM	2K	12			
SYSGEN.MAC	8K	58			
XSUB.COM	2K	6			

36 Files occupying 262K, 220K Free.

To insert files in to the image the command is a bit modified. To extract all the files

use:

```
$cpmcp -f tell lktsysmstr.raw 0:*. * /home/larry/IMDs/teletek/td0/tsysms
```

If you have inserted files into the *.RAW Image with cpmtools, you can use the Imagedisk utility bin2imd to create an updated .IMD file.

The next thing to do is verify all source files *.ASM, *.Z80, *.TXT, *.MAC, *.DOC are SANE.

M.DOC needs to be stripped to ASCII, and then it appears SANE. I used the DOS program

WSX21A.COM to strip the *.DOC file to ASCII.

IMPORTANT: READ ALL OF THIS DOCUMENTATION BEFORE DOING ANYTHING. MAKE NOTE OF THE AUTOMATIC SELECTION OF SINGLE AND DOUBLE SIDED DISKETTE UNDER DISKETTE DENSITIES IN PARTICULAR.

Documentation for Teletek's SYSTEMASTER CP/M 2.2 [Release B.00]

CONGRATULATIONS!!!, you have just acquired a copy of Teletek's SYSTEMASTER CP/M 2.2 operating system. This operating system permits data storage in either single or double density formats, on single or double sided drives and on either 8 and/or 5.25 inch diskettes. The inch, single density format is fully CP/M 1.4 compatible, allowing standard format for information interchange. You should use the 8 inch single density format whenever exchanging any software.

A companion EPROM comes with this system. It contains the auto-boot code to boot up your CP/M system. There is no resident monitor that takes up room in memory. The EPROM resides at 0000h and moves a portion of itself up to fd00h. It then jumps to that portion and enables the SYSTEMASTER for CP/M. It loads the first 128 bytes of the cold-start-loader [SMCSL] from the first sector of the first track of the system diskette at location 0000h. It finally jumps to SMCSL, and SMCSL loads the rest of itself from the second sector, using the utilities of the EPROM module at location fd00h. Then SMCSL prepares to load the BIOS into memory. It overlays the EPROM module with the BIOS. After loading the BIOS and initializing page 0 parameters it then jumps to the BIOS via the COL BOOT vector. The BIOS then reinitializes and controls the entire SYSTEMASTER board. The BIOS now waits until the baud rate of the console is determined. You just press the [RETURN] key an appropriate number of times. The BIOS starts at 19200 baud and reduces the baud rate until it recognizes the [RETURN]. Once the baud rate is recognized the sign-on message will be displayed. This means that you must have an RS-232 compatible serial terminal as your console device and this device is connected to the [SIO B] port of your SYSTEMASTER. That is, it closes one to the drive cable connector. The present implementation of CP/M 2.2 release B.00 also allows for static setting of the baud rate for the console device. If you prefer to have the system boot all the way up without waiting for carriage returns, use CONSYS to select Static baud mode and the appropriate static baud.

Once the terminal is signed on, the cold boot initialization is disabled, and the rest of CP/M is booted in, i.e.: CCP and BDOS.

The standard system is a 64K system, making the most use of the memory. Release B.00 leaves a 256 byte memory window at ff00h to ffff that you can use for code outside of CP/M, or for expansion of SMBIOS. You may want to open more memory above CP/M. If so then use MOVCPM to regenerate a new system with a lower number than 64K. This will give you additional 1K increments of memory. Refer to your CP/M user's guide.

[CAUTION].....[CAUTION]

The first operation that should be performed is to make a backup copy of the enclosed diskette. This is done as follows:

1. Power up or Reset your system, then put the original diskette in drive A (physical drive 0). It should have, at this point, loaded the SMCSL and SMBIOS into memory. It is now waiting for you to press your console [RETURN] key and will wait until it finds the proper baud rate.
2. After finding the baud rate, a sign-on message should be printed on your terminal, telling you that you are now running a 64K CP/M 2.2 system.
3. Put a blank diskette in drive B (physical drive 1).
4. Enter "FORMAT"[CR]. You are now running the format program.
5. When asked for the drive enter "B"[CR], and when asked for the side enter "0"[CR].
6. Next the format program will ask for an option. Choose the option that will format the diskette in double density but with track 0 in single density.
7. When asked if you are ready to format the diskette, enter "Y"[CR].
8. After the diskette is formatted the program will ask the ready question again. This time enter "^C" (control C) and you should get "A>" meaning that drive A is logged on.
9. Enter "BACKUP"[CR]. You are now running the backup program.
10. When asked for the source drive enter "A"[CR].
11. When asked for the destination drive enter "B"[CR].
12. Next the backup program will ask for an option. Choose the option that will backup in double density but with track 0 in single density and only on one side.
13. When asked if you are ready to copy A onto B enter "Y"[CR].
14. After the backup copy is made, you will be asked the ready question again. Before answering, take the original diskette from drive A and store it in a very safe place. Take the backup copy from drive B and put it in drive A. Now enter "^C" (control C) and you should get "A>", telling you that drive A is logged on.

You should use this backup diskette for the operation of the system. It would be a good idea to make several backup copies.

Diskette Densities

Teletex's CP/M 2.2 system is set up to allow both single and double density diskettes to be used. It will automatically find the density of the diskette when the drive is first logged on. If the system is unable to find the density of the diskette, it will print an error message. To transfer files between densities use PIP. For example, if drive A has a double density diskette in it and drive B has a single density diskette in it the following command will transfer all the "COM" from drive B to drive A: "PIP A:=B:*.COM[V]".

If the density is double, the system will determine whether the diskette is single or double sided. If the drive is single sided then the system is set up for single sided diskettes. Teletex strongly recommends that the user use only double sided diskettes as double sided. There are some 8 inch drives such as the Calcomp 143M that allow the user to format a single sided diskette on both sides. If the user formats a single sided diskette on both sides it will cause problems if the user tries to use the diskette on a single sided drive. On 5.25 inch drive problems may arise since there are no physical differences between single and double sided diskettes as for 8 inch diskettes.

Floppy Disk Errors

If an error occurs during the course of operation, the system will print "Disk Err on d: type" (where "d" is the drive the error occurred on and "type" is the error type) along with the error code and on which track head and sector the error occurred on. The error codes are defined in Table 1. The system then waits for a key from the user specifying what action to take as follows:

^C (control C) - reboot the system,
Carriage return - return the error to the BDOS,
any other key - retry the operation.

When performing random access on files sometimes a file will not be closed properly. In this case "BDOS Err on d: File Open" will be printed. If the user wants to ignore the error he should enter "^C". Otherwise he should make sure the correct diskette is in drive "d" and hit any key.

Table 1
EXPLANATION OF ERROR CODES

Error Code	Definition
AT	Abnormal Termination: The ready line of the drive changed state during the disk operation. This will occur very rarely on single sided drives. This will occur on double sided drives if the drive is logged on with a double sided

diskette and then a single sided diskette is insert without relogging the drive.

- EC Equipment Check: The drive did not recalibrate (step 0 to track zero). Rarely occurs but if it occurs too oft the drive should be checked.
- NR Not Ready: The drive is not ready. Make sure t diskette is properly inserted and the drive door closed.
- DE Data Error: A CRC error has occurred on the sector. Th occurs if a diskette that was written with pre-wri compensation is read on a drive that does not require pr write compensation or vice-versa. This may also occur the diskette is getting worn out.
- ND No Data: The desired sector was not found. This usual occurs if there is an error in the software unless occurs with WC.
- NW Not Writable: The diskette is write protected. Make su the "write protect" notch is covered (uncovered for 5.2 drives).
- MA Missing Address Mark: The address mark on the disket was not found. This usually occurs if a drive is logged with a double density diskette and then a single densi is inserted or vice-versa.
- WC Wrong Cylinder: The drive is on the wrong track. Th occurs with error ND. Most likely the step rate is t fast for the drive.

Utility Programs

Four utilities are included with the CP/M 2.2 system. All of th utilities are self-explanatory but further information on their use i given in the following paragraphs. If an error occurs during th operation of any of the utilities, an appropriate error message will b given. All of the utilities will run on both 8 or 5.25 inch systems.

SYSGEN - This utility allows the generation of new CP/M systems. Its operation is explained in the CP/M manuals. The diskette that a system is going to be put on **MUST** be formatted double density with track 0 in single density on side 0.

FORMAT - This utility will format diskettes in either single or double density formats. It asks for the drive, side and density option. If a system is going to be put on the diskette, it should be formatted in double density with track 0 in single density, but only on side 0. Normally, single density is used only on side 0. When using double sided drives, side 1 is formatted in double density including track 0.

BACKUP - This utility makes backup copies of diskettes. It copies the diskettes on a track by track basis verifying each

copies the diskettes on a track by track basis, verifying each track. It will make a backup copy much faster than PIP but it does it on the whole diskette not just selective files. It asks for both the source and destination drives along with the desired density option. Both the source and destination diskettes have to be of the same size, same number of sides, and formatted with the same format.

CONSYS - This utility will configure the system. It will print the current configuration of the system or it will configure a new system. It will set up the following parameters:

1. The number of physical drives in the system. The system always has four logical drives, but when only 1 or 2 drives are specified then logical drives A and C refer to physical drive 0 and logical drives B and D refer to physical drive 1. This allows data transfers from single to single density on a two drive system.
2. The size of each of the drives, either 8 or 5.25 inch. If different drive sizes are used, the appropriate drive cable is needed for the SYSTEMASTER.
3. The step rate for 8 inch drives (if any).
4. The step rate for 5.25 inch drives (if any).
5. The baud rate for SIO A is for the serial list device.
6. The baud assignment mode for SIO B is for the console. If static baud mode, the baud rate for SIO B must also be set.
7. The default LIST device, serial (LPT:) or parallel (UL1:).
8. If a serial LIST device, the type of communication protocol. Two types of protocol are available: ETX/ACK or X-ON/X-OFF.
9. Some printers require that bit 7 of the bytes of list data stream be set to zero, otherwise, strange things happen. This option is available. The data stream byte may be masked by [7FH].

After configuring a new system, a cold boot operation must be performed for the parameters to take effect, i.e. hitting reset and booting from scratch with the newly configured diskette in drive A.

CP/M I/O Byte

The CP/M I/O byte is only implemented for the LIST device. The following devices are implemented:

LPT: - the serial printer,
UL1: - the parallel printer,

IIY: OR UKI: - the console.

It is possible to set the initial LIST device with CONSYS. To change the LIST device use STAT. In other words, if the default LIST device is a serial device it is possible to output to a parallel printer by connecting the printer and changing the I/O byte for UL1:. Once changed, the I/O byte will remain the same unless changed by the user on a cold boot. A warm boot will not change the I/O byte.

The READER is defined to be the device connected to SIO A.

Diskette Organization

The enclosed BIOS allows the user to intermix single and double density diskettes on either 8 and/or 5.25 inch drives using Teletek's SYSTEMASTER. In order to implement these features the BIOS is larger than the 896 bytes allowed in the standard CP/M 2.2 from Digital Research. Thus, the diskette organization is different from the Digital Research organization. The CP/M system has to boot from a double density diskette that has track 0 in single density. Teletek's CP/M 2.0 organization for 8 inch drives is:

1. Track 0, sectors 1 and 2 - Cold Start Loader
2. Track 0, sectors 3 through 26 and
Track 1, sectors 1 through 10 - CCP and BDOS
3. Track 1, sectors 11 through 25 - BIOS
4. Track 1, sector 26 - reserved for future volume id information.

For 5.25 inch drives the organization is:

1. Track 0, sectors 1 and 2 - Cold Start Loader
2. Track 0, sectors 3 through 18 and
Track 1, sectors 1 through 14 - CCP and BDOS
3. Track 1, sectors 15 through 18 and
Track 2, sectors 1 through 11 - BIOS
4. Track 2, sectors 12 through 17 - not used
5. Track 2, sector 18 - reserved for future volume id information

All of the utilities expect this organization and can be used on both 8 and 5.25 inch systems. Note: If the BIOS is modified, the hard code should not exceed 3 3/4K bytes (15 times 256).

Generation of a Mixed Size System

If a mixed size system is desired (both 8 and 5.25 inch drives) the following requirements are necessary: a cable for the SYSTEMASTER that can handle both drive sizes.

The following steps show how to generate a two 8 inch and two 5.25 inch drive system from an all 8 inch system:

1. Make a backup copy of your current system.
2. Cold boot with the backup copy.
3. Run "CONSYS" using "A" as the source drive , "4" as the number of physical drives and configure a system with drives C and D 5.25 inch.
4. Press reset and cold boot the system. Physical drives 2 and 3 are now 5.25 inch drives to the system.

To generate an all 5.25 inch system from an all 8 inch system repeat the above steps and do the following steps:

1. Run "FORMAT" and format a 5.25 diskette on drive C in double density with track 0 in single density.
2. Run "SYSGEN" using "A" as the source drive and "C" as the destination drive.
3. Run "CONSYS" using "C" as the source drive and configure a system with all the drives 5.25 inch.

A similar process is necessary to generate an all 8 inch system from a all 5.25 inch system.

SYSTEM SOURCE CODE

The system SOURCE code comes in modular form. These modules are assembled together in different combinations. This is to minimize the programming and duplication errors encountered when using code that has the same function in different programs.

The source code modules are:

SMINIT.MOD	- This module initializes the S-M board.
SMDISK.MOD	- This module controls the DISK and RTC functions
SMVCTR.MOD	- This module vectors to the functions of SMDISK.MOD through one entry point.
SMJUMP.MOD	- This module provides the code necessary to jump to the entry point of SMVCTR.MOD.
SMBOOT.MOD	- This module initializes CP/M and boots in the CCP and BDOS of CP/M on each warm boot.
SMDVAR.MOD	- This module provides the equates and data for SMDIOS.MOD and SMBOOT.MOD.
SMDIOS.MOD	- This module provides the CP/M DISK I/O interface to SMDISK.MOD.
SMPIOS.MOD	- This module provides the CP/M NON-DISK I/O code to control the PIOs and SIOs of the board.

The source code files are:

EPROM.MAC	- This program is the auto-boot program that
-----------	--

resides in the EPROM on the SYSTEMASTER board. It initializes the board and boots in the first sector of the cold start loader, SMCSL.MAC. It uses modules, SMINIT.MOD, SMVCTR.MOD, and SMDISK.MOD. It passes the vector address to SMVCTR.MOD in the HL register and the SYSTEMASTER control register status byte in register A.

- SMCSL.MAC - This program is the cold start loader for CP/M. It boots the second sector of itself in and then the CP/M bios, SMBIOS.MAC. It initializes address locations 40H to 4EH for the bios to specify the configuration of the CP/M system. It then jumps to the bios. It uses none of the modules.
- SMBIOS.MAC - This program is the CP/M bios that interfaces to the SYSTEMASTER board. It uses all of the above modules, except SMJUMP.MOD. This program reinitializes and takes full control of the SYSTEMASTER board for CP/M.
- CONSYS.MAC
SYSGEN.MAC
BACKUP.MAC
FORMAT.MAC - These programs are the utilities for the system. They are explained earlier in this documentation. They all use SMJUMP.MOD. These utilities are essentially the same utilities supplied with earlier products of TELETEK ENTERPRISES, INC. Modifications were made to conform with the SYSTEMASTER board.

MONITOR DOCUMENTATION

The file BRKPNT.MAC contains a break-point monitor. It is on the diskette for your use. The CP/M system that came with this diskette does not have a monitor. If for any reason you are changing the configuration of the system, you may need a resident monitor to debug the system. An auto boot system does not lend itself to debugging. There is no way that it can react to bugs if these bugs interfere with the auto booting and accessing of CP/M.

BRKPNT is similar to the functions of DDT.COM of CP/M. It was used to debug SMBIOS.MAC that came with your CP/M. It needs an experienced programmer to use it. A listing of the source should be taken and thoroughly studied. It requires a working system to load it into memory.

Assuming that you want to change the SMBIOS some way and patch it into the system, you should do the following.

1. Using MOVCPM.COM make the appropriate CP/M size, say 64k system then save it.

```
MOVCPM 64 *
[move cpm messages]
SAVE 43 NCPM64.COM
```

2. You now have a system that you can patch. Make the appropriate changes in the bios that you want, then make a COM file of it. Make sure that ENDUSER flag is set TRUE in bios code and the right size is set.

```
M80 =BIOS
L80 /P:100,BIOS,BIOS/N/E
```

3. You now have a new bios to patch into the CP/M system. Patch it the bios the following way:

```
DDT NCPM64.COM
IBIOS.COM
R1F00
G0
SAVE 43 NCPM64.COM [this wipes out the prior NCPM64.COM]
SYSGEN
[bypass loading source by making just a CR]
[load cpm in memory to the new system diskette]
```

4. Now at this point, if your new system does not work, you need BRKPNT monitor to come to the rescue. With your working system load it NCPM64.COM with DDT:

```
DDT NCPM64.COM
[look for the sign-on message in the boot section of the bios
[the bios starts at 2000h]
[patch the last 4 bytes of the message with B0,C3,00,80]
G0
SYSGEN
[put the new system on the diskette]
```

5. Next, using DDT, load BRKPNT.COM to 8000H,

```
DDT[CR]
IBRKPNT.COM
R7F00
```

6. Put the new system in the boot drive and restart. You should have entered BRKPNT. Now, happily debug your new BIOS. GOOD LUCK!!!!!!

BREAK POINT MONITOR COMMANDS

CA NNNN	- Change register pair AF with NNNN
CB NNNN	- Change register pair BC with NNNN
CD NNNN	- Change register pair DE with NNNN
CH NNNN	- Change register pair HL with NNNN
CS NNNN	- Change stack pointer with NNNN
CP NNNN	- Change program counter with NNNN
DA NNNNN	- Display with ASCI from NNNNN

E	NNNN	- Enter data starting at NNNN
EX	BBBB	- Execute from PC to break point BBBB
F	IIII JJJJ K	- Fill memory from IIII to JJJJ with K
H	IIII JJJJ	- Hex add and subtract JJJJ to and from IIII
G	NNNN	- Go to or jump unconditionally to NNNN
I	NN	- Input from port NN
O	NN BB	- Output to port NN with BB
M	IIII JJJJ KKKK	- Move memory contents IIII-JJJJ to KKKK

(This monitor does not consider alternate registers or the index registers IX and IY.)

I did locate one file that had several corrupt bytes, but there was enough information remaining to rebuild the single file.

Somewhere in my travels on the Internet, I located a Teletek file named 59KCPM.IMD. If anyone has a clue where that file was located I'd like to have the URL.

Enjoy!

Larry

Re: [HOWTO] - Access CP/M Floppy's via libdisk & cpmtools

by **lkraemer**

Posted: **2021-03-08 15:32**

Re: [HOWTO] - Access CP/M Floppy's via libdisk & cpmtools

Post by lkraemer » 2021-03-08 09:00

UPDATE 03-08-2021

I've just located some Piccolo RC70? files at the following URL:
<http://www.jbox.dk/rc702/disks.shtml>

The first thing needed is some Floppy Definitions for 22DISK, cpmtools, and libdisk to be able to access the Images *.TD0 or *.IMD.

Definitions for 22DISK, cpmtools, & libdisk:

```
BEGIN RC70  PICC0L0 RC702 - DSDD 5.25" - 48 TPI  512 x 9
DENSITY MFM, LOW
CYLINDERS 35 SIDES 2 SECTORS 9,512 SKEW 5
SIDE1 0 1,3,5,7,9,2,4,6,8
SIDE2 1 1,3,5,7,9,2,4,6,8
ORDER CYLINDERS
BSH 4 BLM 15 EXM 1 DSM 134 DRM 127 AL0 0C0H AL1 0 OFS 2
END
```

#cpmtools definition

```
# RC70  PICC0L0 RC702 - DSDD 5.25" - 48 TPI  512 x 9
diskdef rc70
    seclen 512
    tracks 70
    sectrk 9
    blocksize 2048
    maxdir 128
    skew 5
    . . . . .
```

```

    boottrk 2
    os 2.2
end

```

```

#libdisk definition
[rc70]

```

```

description = RC70 PICCOLO RC702 - DSDD 5.25" - 48 TPI 512 x 9
sides = outback
cylinders = 70
heads = 2
seclen = 512
sectors = 9
seclen = 1
datarate = DD

```

```

BEGIN RC71 PICCOLO RC702 - DSDD 5.25" - 48 TPI 512 x 9
DENSITY MFM, LOW
CYLINDERS 35 SIDES 2 SECTORS 9,512 SKEW 5
SIDE1 0 0,2,4,6,8,1,3,5,7
SIDE2 1 0,2,4,6,8,1,3,5,7
ORDER CYLINDERS
BSH 4 BLM 15 EXM 1 DSM 134 DRM 127 AL0 0C0H AL1 0 OFS 2
END

```

```

#cpmtools definition

```

```

# RC71 PICCOLO RC702 - DSDD 5.25" - 48 TPI 512 x 9
diskdef rc71
    seclen 512
    tracks 70
    sectrk 9
    seclen 0
    blocksize 2048
    maxdir 128
    skew 5
    boottrk 2
    os 2.2
end

```

```

#libdisk definition
[rc71]

```

```

description = RC71 PICCOLO RC702 - DSDD 5.25" - 48 TPI 512 x 9
sides = outback
cylinders = 70
heads = 2
seclen = 512
sectors = 9
seclen = 0
datarate = DD

```

If the Images are a .TD0 format, the first thing needed is to create the *.IMD file with the TD02IMD utility of Imagedisk Ver 1.18.

```
TD02IMD CPM56K.TD0 > CPM56K.TXT
```

I downloaded the following images:

DOTEST.IMD

GAMES.IMD

MBASIC521.IMD

WORDSTAR.IMD CPM56K.IMD

I use DOSBox on my Linux Laptop to get more information on the *.IMD files, by using the IMDA Utility.

```
IMDA D0TEST.IMD > D0TEST.TXT
IMDA GAMES.IMD > GAMES.TXT
IMDA MBASIC521.IMD > MBASIC521.TXT
IMDA WORDSTAR.IMD > WORDSTAR.TXT
IMDA CPM56K.IMD > CPM56K.TXT
```

I then extract the *.RAW Files (Sector Dumps) so I can get directory listings with cpmtools.

```
IMDU GAMES.IMD GAMES.RAW /B /E /D >> GAMES.TXT
IMDU MBASIC521.IMD MBASIC521.RAW /B /E /D >> MBASIC521.TXT
IMDU WORDSTAR.IMD WORDSTAR.RAW /B /E /D >> WORDSTAR.TXT
IMDU CPM56K.IMD CPM56K.RAW /B /E /D >> CPM56K.TXT
```

The first TXT file has:

#IMDA's Information

IMageDisk Analyzer 1.18 / Mar 12 2012

IMD 1.18: 9/03/2014 15:17:55

CP/M 56k vers 2.2

rel. 2.1

UTILITY PROGRAMS

Michael Ringgaard

Required cylinders: 35
Required heads : 2
Data rate : 250kbps
Est. maximum track: 5541 bytes

Possible drives/options to write CPM56K.IMD :

5.25" DD 40-track
Double-step: OFF

5.25" QD 80-track
Double-step: ON

5.25" HD 80-track
Double-step: ON
250 kbps -> 300 kbps

3.5" DD 80-track NOTE: *1
Double-step: OFF

3.5" HD 80-track NOTE: *1
Double-step: OFF

*1 40 track image will use only first 1/2 of 80-track drive.

#IMDU's added Information

IMageDisk Utility 1.18 / Mar 07 2012

IMD 1.18: 9/03/2014 15:17:55

CP/M 56k vers 2.2

rel. 2.1

UTILITY PROGRAMS

Michael Ringgaard

Assuming 1:1 for Binary output

```

0/0 250 kbps SD 16x128
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D
0/1 250 kbps DD 16x256
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D
1/0 250 kbps DD 9x512
  1  2  3  4  5  6  7  8  9
  D  D  D  D  D  D  D  D  D
1/1 D  DE5 D  DE5 DE5 DE5 DE5 DE5 DE5
2/0 D  DE5 D  DE5 D  DE5 DE5 D  DE5
2/1 D  D  D  D  D  D  D  D  D
3/0 D  D  D  D  D  DE5 D  DE5 D
3/1 DE5 D  D  D  D  D  D  D  D
4/0 D  D  D  D  D  D  D  D  D
4/1 D  D  D  D  D  D  D  D  D
5/0 D  D  D  D  D  D  D  D  D
5/1 D  D  D  D  D  D  DE5 D  DE5
6/0 D  D  D  D  D  D  D  D  D
6/1 D  DE5 D  DE5 D  D  D  D  DE5
7/0 D  D  D  D  D  D  D  DE5 D
7/1 DE5 D  D  D  DE5 D  DE5 D  DE5
8/0 D  DE5 D  DE5 D  DE5 D  D  D
8/1 D  D  D  D  D  D  D  D  D
9/0 D  DE5 D  D  D  D  DE5 D  DE5
9/1 DE5 D  D  D  D  D  D  D00 D
10/0 D  D  D  D  D  D  D  D  D
10/1 D  DE5 D  DE5 D  D  D  D  DE5
11/0 D  D  D  D  D  D  D  D  D
11/1 D  D  D  D  D  D  D  DE5 D
12/0 D  D  D  D  D  D  D  D  D
12/1 D  D  D  D  D  D00 D  D00 D
13/0 D00 D  D00 D  D  D  D  D  D
13/1 D  D  D  D  D  D  D  D  D
14/0 D  D  D  D  D  D  D  D  D
14/1 DE5 D  DE5 D  DE5 D  D  D  D
15/0 D  DE5 D  D  D  D  D  D  D
15/1 D  D  D  D  D  D  D  D  D
16/0 D  D  D  D  D  D  D  D  D
16/1 D  D  D  DE5 D  D  D  D  D
17/0 D  D  DE5 D  D  D  D  DE5 D

```

```

17/1 D D D D D D D D D
18/0 D DE5 D DE5 D DE5 D D D
18/1 D D DE5 D DE5 D D D D
19/0 D D DE5 D D D D D D
19/1 D D D D D D D D D
20/0 D D D D D D D D D
20/1 D D D D D D D D D
21/0 D D D D D D D D D
21/1 D D D D D D D D D
22/0 D D D D D D D DE5 D
22/1 DE5 D DE5 D D D D D D
23/0 D D D D D D D D D
23/1 D D DE5 D D D DE5 D DE5
24/0 D D D D D D D D D
24/1 D D D D D D D D D
25/0 D D D D D D D D D
25/1 D D D D D D D D D
26/0 D D D D D D D DE5 D
26/1 DE5 D DE5 D DE5 D D D
27/0 D DE5 D D DE5 D DE5 D DE5
27/1 D D D D D D D D D
28/0 D D D D D D D D D
28/1 D D D D D D D D D
29/0 D DE5 D D D DE5 D DE5 DE5
29/1 DE5 D D D00 D D00 D D00 D
30/0 D00 D D D D D D D D
30/1 D D D D D D D D D
31/0 D D D D D D D DE5 D
31/1 DE5 D D D D D D D D
32/0 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5
32/1 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5
33/0 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5
33/1 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5
34/0 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5
34/1 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5
70 tracks(35/35), 644 sectors (134 Compressed)

```

Notice the First Tracks are all different:

```

0/0 250 kbps SD 16x128 = 2048 = 0x800
  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16
  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D
0/1 250 kbps DD 16x256 = 4096 = 0x1000
  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16
  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D
1/0 250 kbps DD 9x512
  1  2  3  4  5  6  7  8  9
  D  D  D  D  D  D  D  D  D
1/1 D DE5 D DE5 DE5 DE5 DE5 DE5 DE5 DE5

```

The Directory listing can be located by viewing the LKM56K.RAW file:

```

00004400 00 42 4C 4F 43 4B 53 20 20 C2 41 44 00 00 00 80 .BLOCKS
00004410 5C 61 6E 71 00 00 00 00 00 00 00 00 00 00 00 \anq....
00004420 00 41 53 53 49 47 4E 20 20 43 4F 4D 00 00 00 03 .ASSIGN
00004430 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00004440 00 46 49 4C 45 58 20 20 20 41 53 4D 01 00 00 18 .FILEX
00004450 07 08 09 0A 0B 0C 0D 0E 0F 10 00 00 00 00 00 .....
00004460 00 43 41 54 20 20 20 20 20 43 4F 4D 00 00 00 2F .CAT
00004470 11 12 13 00 00 00 00 00 00 00 00 00 00 00 00 .....
00004480 00 46 49 4C 45 58 20 20 20 43 4F 4D 00 00 00 11 .FILEX

```

```

00004490  14 15 00 00  00 00 00 00  00 00 00 00  00 00 00 00  .....
000044A0  00 44 44 54  20 20 20 20  20 43 4F 4D  00 00 00 26  .DDT
000044B0  16 17 18 00  00 00 00 00  00 00 00 00  00 00 00 00  .....
000044C0  00 44 55 4D  50 20 20 20  20 43 4F 4D  00 00 00 04  .DUMP
000044D0  19 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  .....
000044E0  00 44 55 4D  50 20 20 20  20 41 53 4D  00 00 00 21  .DUMP
000044F0  1A 1B 1C 00  00 00 00 00  00 00 00 00  00 00 00 00  .....
00004500  00 45 44 20  20 20 20 20  20 43 4F 4D  00 00 00 34  .ED
00004510  1D 1E 1F 20  00 00 00 00  00 00 00 00  00 00 00 00  ... ....
00004520  00 46 4F 52  4D 41 54 20  20 43 4F 4D  00 00 00 0C  .FORMAT
00004530  21 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  !.....
00004540  00 56 45 52  49 46 59 20  20 43 4F 4D  00 00 00 53  .VERIFY
00004550  22 23 24 25  26 27 00 00  00 00 00 00  00 00 00 00  "$%&'..

```

The Directory is located at 0x4400 which is 17408 Decimal. This offset must be added to the cpmtools definition.

By editing the definition to point at the first of the Directory I can get a Directory listing, but may not be able to extract valid files.

```
# RC01 RC-702 Piccolo - DSDD 48 tpi 5.25" - 512 x 9
```

```

diskdef rc01
  seclen 512
  tracks 70
  sectrk 9
  blocksize 2048
  maxdir 128
  skew 2
#15360 = 0x3C00
#17408 = 0x4400
  boottrk 0 #4
  offset 17408
  os 2.2
end

```

```
#cpmtools
# RC02 RC-750 Piccolo - DSHD 96 TPI 5.25" - 1024 x 8
```

```

diskdef rc02
  seclen 1024
  tracks 154
  sectrk 8
  blocksize 2048
  maxdir 512
#  skew 2
  boottrk 4
  os 3
end

```

```

diskdef rtek1
  seclen 512
  tracks 70
  sectrk 9
  blocksize 2048
  maxdir 128
  skew 2
  boottrk 2
  os 2.2
.

```

end

#cpmtools

RC03 RC-700 Piccolo RCTEKST - DSDD 48 tpi 5.25" - 512 x 9

diskdef rc03

seclen 512

tracks 70

sectrk 9

blocksize 2048 #BSH 4 BLM 15

maxdir 128 #512 /32 = 16 * 8 = 128

skew 2

boottrk 2

os 2.2

end

\$ cpmfs -f rc01 -D LKM56K.RAW

Name	Bytes	Recs	Attr	update	create
ASM	.COM	8K	64		
ASSIGN	.COM	2K	3		
AUTOEXEC	.COM	2K	8		
BACKUP	.COM	10K	80		
BLOCKS	.BAD	16K	128 R		
CAT	.COM	6K	47		
CONFI	.COM	8K	51		
DDT	.COM	6K	38		
DESCRIP	.TXT	2K	5		
DUMP	.ASM	6K	33		
DUMP	.COM	2K	4		
DUP	.COM	30K	231		
ED	.COM	8K	52		
FILEX	.ASM	20K	152		
FILEX	.COM	4K	17		
FORMAT	.COM	2K	12		
HDINST	.COM	4K	28		
LOAD	.COM	2K	14		
LOADRAM	.COM	8K	62		
PIP	.COM	8K	58		
RESTORE	.COM	22K	175		
SELECT	.COM	2K	2		
SEMI	.CHR	6K	40		
STAT	.COM	6K	41		
STORE	.COM	12K	94		
SUBMIT	.COM	2K	10		
SYSGEN	.COM	2K	11		
TRANSFER	.COM	2K	13		
TTY	.COM	22K	168		
US-ASCII	.COM	2K	4		
VERIFY	.COM	12K	83		
XSUB	.COM	2K	6		

32 Files occupying 238K, 72K Free.

Updated Definitions for 22DISK, cpmtools, & libdisk:

BEGIN RC70 PICC0L0 RC702 - DSDD 5.25" - 48 TPI 512 x 9

DENSITY MFM, LOW

CYLINDERS 35 SIDES 2 SECTORS 9,512 SKEW 5

SIDE1 0 1,3,5,7,9,2,4,6,8

SIDE2 1 1,3,5,7,9,2,4,6,8

ORDER CYLINDERS

```
BSH 4 BLM 15 EXM 1 DSM 134 DRM 127 AL0 0C0H AL1 0 OFS 2
END
```

#cpmtools definition

```
# RC70 PICCOLO RC702 - DSDD 5.25" - 48 TPI 512 x 9
```

```
diskdef rc70
    seclen 512
    tracks 70
    sectrk 9
    blocksize 2048
    maxdir 128
    skew 5
    boottrk 2
    os 2.2
```

```
end
```

#libdsk definition

```
[rc70]
```

```
description = RC70 PICCOLO RC702 - DSDD 5.25" - 48 TPI 512 x 9
sides = outback
cylinders = 70
heads = 2
seclen = 512
sectors = 9
secbase = 1
datarate = DD
```

```
BEGIN RC71 PICCOLO RC702 - DSDD 5.25" - 48 TPI 512 x 9
```

```
DENSITY MFM, LOW
```

```
CYLINDERS 35 SIDES 2 SECTORS 9,512 SKEW 5
```

```
SIDE1 0 0,2,4,6,8,1,3,5,7
```

```
SIDE2 1 0,2,4,6,8,1,3,5,7
```

```
ORDER CYLINDERS
```

```
BSH 4 BLM 15 EXM 1 DSM 134 DRM 127 AL0 0C0H AL1 0 OFS 2
END
```

#cpmtools definition

```
# RC71 PICCOLO RC702 - DSDD 5.25" - 48 TPI 512 x 9
```

```
diskdef rc71
    seclen 512
    tracks 70
    sectrk 9
    secbase 0
    blocksize 2048
    maxdir 128
    skew 5
    boottrk 2
    os 2.2
```

```
end
```

#libdsk definition

```
[rc71]
```

```
description = RC71 PICCOLO RC702 - DSDD 5.25" - 48 TPI 512 x 9
sides = outback
cylinders = 70
```

```

heads = 2
secksize = 512
sectors = 9
seckbase = 0
datarate = DD

```

Try these updated definitions. Remember, that when trying to access the Floppy's data (RAW SECTOR DUMP)

the first two tracks can be adjusted by 2048 bytes (16 x 128) to make Track 0/0 & 0/1 the same size. I use dd to

copy Track 0/0 to a file, then append the remaining total file to get Track0 Side 0 & Side 1 both at 4096 bytes.

This allows cpmtools to access the data files. cpmtools does not like a SD first track and DD second track even

though it is supposed to work correctly using the offset (SOFS) value. OFS is basically the Boot Tracks while

SOFS is sector offset to the Directory. The modified definitions are just used to Extract the data files.

```

0/0 250 kbps SD 16x128 = 2048 = 0x800
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D
0/1 250 kbps DD 16x256 = 4096 = 0x1000
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D
1/0 250 kbps DD 9x512 = 4608 = 0x1200
  1  2  3  4  5  6  7  8  9
  D  D  D  D  D  D  D  D  D
1/1  D  DE5 D  DE5 DE5 DE5 DE5 DE5 DE5

```

I started with CPM56K.IMD and created the CPM56K.RAW file, then copied the first 2048 bytes to a new file.

Then appended the complete file.

```
$ dd if=CPM56K.RAW bs=1 count=2048 of=LKM56K.RAW conv=notrunc
```

```
$ dd if=CPM56K.RAW bs=1 seek=2048 of=LKM56K.RAW conv=notrunc, sync
```

A directory listing shows:

```
321536 Mar 7 09:17 LKM56K.RAW
```

```
2048 + 319488
```

```
319488 May 6 2017 CPM56K.RAW
```

```
$ cpm ls -f rc01 -D LKM56K.RAW
```

Name	Bytes	Recs	Attr	update	create
ASM	.COM	8K	64		
ASSIGN	.COM	2K	3		
AUTOEXEC	.COM	2K	8		
BACKUP	.COM	10K	80		
BLOCKS	.BAD	16K	128 R		
CAT	.COM	6K	47		
CONFI	.COM	8K	51		
DDT	.COM	6K	38		
DESCRIP	.TXT	2K	5		
DUMP	.ASM	6K	33		
DUMP	.COM	2K	4		
DUP	.COM	30K	231		
ED	.COM	8K	52		
FILEX	.ASM	20K	152		
FILEX	.COM	4K	17		

FURMAI	.COM	2K	12
HDINST	.COM	4K	28
LOAD	.COM	2K	14
LOADRAM	.COM	8K	62
PIP	.COM	8K	58
RESTORE	.COM	22K	175
SELECT	.COM	2K	2
SEMI	.CHR	6K	40
STAT	.COM	6K	41
STORE	.COM	12K	94
SUBMIT	.COM	2K	10
SYSGEN	.COM	2K	11
TRANSFER	.COM	2K	13
TTY	.COM	22K	168
US-ASCII	.COM	2K	4
VERIFY	.COM	12K	83
XSUB	.COM	2K	6

32 Files occupying 238K, 72K Free.

Now, to extract all the files, and see if the ASCII ones are sane.

```
$ cpmcp -f rc01 LKM56K.RAW 0:*. * /home/larry/IMD-118/piccoline/CPM56K
```

Directory Listing:

```
8192 Mar 7 09:26 asm.com
640 Mar 7 09:26 descrip.txt
21504 Mar 7 09:26 tty.com
29568 Mar 7 09:26 dup.com
7936 Mar 7 09:26 loadram.com
5120 Mar 7 09:26 semi.chr
512 Mar 7 09:26 us-ascii.com
1024 Mar 7 09:26 autoexec.com
10240 Mar 7 09:26 backup.com
6528 Mar 7 09:26 confi.com
3584 Mar 7 09:26 hdinst.com
12032 Mar 7 09:26 store.com
1280 Mar 7 09:26 submit.com
1408 Mar 7 09:26 sysgen.com
1664 Mar 7 09:26 transfer.com
768 Mar 7 09:26 xsub.com
1792 Mar 7 09:26 load.com
7424 Mar 7 09:26 pip.com
22400 Mar 7 09:26 restore.com
256 Mar 7 09:26 select.com
5248 Mar 7 09:26 stat.com
6016 Mar 7 09:26 cat.com
4864 Mar 7 09:26 ddt.com
4224 Mar 7 09:26 dump.asm
512 Mar 7 09:26 dump.com
6656 Mar 7 09:26 ed.com
19456 Mar 7 09:26 filex.asm
2176 Mar 7 09:26 filex.com
1536 Mar 7 09:26 format.com
10624 Mar 7 09:26 verify.com
384 Mar 7 09:26 assign.com
16384 Mar 7 09:26 blocks.bad
$gedit descrip.txt
```

;

```
;
; THIS IS THE PARAMETER FILE FOR THE RC700 TERMINAL EMULATOR
;
; THIS FILE INCLUDES EXAMPLES OF ALL PARAMETERS
;

; BAUDRATE'S : 50,75,110,150,300,600,1200,2400,4800,9600,19200
; PARITY      : EVEN,ODD,NO

/TERMINAL
BAUDRATE=2400
PARITY=EVEN

; MODE        : FDX,HDX

/MODE=FDX

/HOST=DC1DC3

/PRINTER
BAUDRATE=1200
PARITY=EVEN

; VALUES FOR /SLAVE, /DUMP, /RECEIVE, /TRANSMIT AND /RETURN ARE ASCII

/DUMP=4
/SLAVE=16
/RECEIVE=18
/TRANSMIT=20
/RETURN=5

/END
```

semi.chr also looks sane.

```
*
* SEMIGRAPHIC CHARACTER SET WITH FRAME CHARACTERS
*
* 1 2 3 4 5 6 7 8 9 10 11
*
00 00 00 00 00 00 78 08 08 08 08 08
01 00 00 00 00 00 0F 08 08 08 08 08
02 08 08 08 08 08 78 00 00 00 00 00
03 08 08 08 08 08 0F 00 00 00 00 00
04 00 00 00 00 00 7F 08 08 08 08 08
05 08 08 08 08 08 0F 08 08 08 08 08
06 08 08 08 08 08 78 08 08 08 08 08
07 08 08 08 08 08 7F 00 00 00 00 00
08 00 00 00 00 00 7F 00 00 00 00 00
09 08 08 08 08 08 08 08 08 08 08 08
0A 08 08 08 08 08 7F 08 08 08 08 08
0B 01 02 02 04 04 08 10 10 20 20 40
0C 40 20 20 10 10 08 04 04 02 02 01
0D 01 02 04 18 20 40 00 00 00 00 00
0E 40 20 10 0C 02 01 00 00 00 00 00
0F 00 00 00 00 00 01 02 0C 10 20 40
10 00 00 00 00 00 40 20 18 04 02 01
```

```

11  41 22 1C 00 00 00 00 00 00 00 00 00
12  00 00 00 00 00 00 00 00 00 1C 22 41
13  01 02 04 08 10 60 00 00 00 00 00 00
14  40 20 10 08 04 03 00 00 00 00 00 00
15  00 00 00 00 00 03 04 08 10 20 40
16  00 00 00 00 00 60 10 08 04 02 01
17  41 22 22 14 14 08 14 14 22 22 41
18  08 08 08 08 08 04 04 02 02 01 01
19  08 08 08 08 08 10 10 20 20 40 40
1A  01 01 02 02 04 04 08 08 08 08 08
1B  40 40 20 20 10 10 08 08 08 08 08
1C  08 08 1C 3E 7F 3E 1C 08 08 00 00 DIAMOND
1D  08 08 1C 3E 7F 7F 2A 08 1C 00 00 SPADE
1E  00 00 36 77 7F 7F 7F 3E 1C 08 00 HEART
1F  1C 1C 08 6B 7F 6B 08 1C 00 00 00 CLUB
20  00 00 00 00 00 00 00 00 00 00 00 00 SPACE
21  0F 0F 0F 00 00 00 00 00 00 00 00 00
22  70 70 70 00 00 00 00 00 00 00 00 00
23  7F 7F 7F 00 00 00 00 00 00 00 00 00
24  00 00 00 0F 0F 0F 00 00 00 00 00 00

25  0F 0F 0F 0F 0F 0F 00 00 00 00 00 00
26  70 70 70 0F 0F 0F 0F 00 00 00 00 00
27  7F 7F 7F 0F 0F 0F 0F 00 00 00 00 00
28  00 00 00 70 70 70 70 00 00 00 00 00
29  0F 0F 0F 70 70 70 70 00 00 00 00 00
2A  70 70 70 70 70 70 70 00 00 00 00 00
2B  7F 7F 7F 70 70 70 70 00 00 00 00 00
2C  00 00 00 7F 7F 7F 7F 00 00 00 00 00
2D  0F 0F 0F 7F 7F 7F 7F 00 00 00 00 00
2E  70 70 70 7F 7F 7F 7F 00 00 00 00 00
2F  7F 7F 7F 7F 7F 7F 7F 00 00 00 00 00
30  00 00 00 00 00 00 00 0F 0F 0F 0F 0F
31  0F 0F 0F 00 00 00 00 0F 0F 0F 0F 0F
32  70 70 70 00 00 00 00 0F 0F 0F 0F 0F
33  7F 7F 7F 00 00 00 00 0F 0F 0F 0F 0F
34  00 00 00 0F 0F 0F 0F 0F 0F 0F 0F 0F
35  0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F
36  70 70 70 0F 0F 0F 0F 0F 0F 0F 0F 0F
37  7F 7F 7F 0F 0F 0F 0F 0F 0F 0F 0F 0F
38  00 00 00 70 70 70 70 0F 0F 0F 0F 0F
39  0F 0F 0F 70 70 70 70 0F 0F 0F 0F 0F
3A  70 70 70 70 70 70 70 0F 0F 0F 0F 0F
3B  7F 7F 7F 70 70 70 70 0F 0F 0F 0F 0F
3C  00 00 00 7F 7F 7F 7F 0F 0F 0F 0F 0F
3D  0F 0F 0F 7F 7F 7F 7F 0F 0F 0F 0F 0F
3E  70 70 70 7F 7F 7F 7F 0F 0F 0F 0F 0F
3F  7F 7F 7F 7F 7F 7F 7F 0F 0F 0F 0F 0F
40  00 14 00 22 22 32 2E 20 00 00 00 00 @
41  00 08 14 22 22 3E 22 22 00 00 00 00 A
42  00 1E 24 24 1C 24 24 1E 00 00 00 00 B
43  00 1C 22 02 02 02 22 1C 00 00 00 00 C
44  00 1E 24 24 24 24 24 1E 00 00 00 00 D
45  00 3E 02 02 0E 02 02 3E 00 00 00 00 E
46  00 3E 02 02 0E 02 02 02 00 00 00 00 F
47  00 3C 02 02 02 32 22 3C 00 00 00 00 G
48  00 22 22 22 3E 22 22 22 00 00 00 00 H

```

```

49 00 1C 08 08 08 08 1C 00 00 00 I
4A 00 20 20 20 20 22 1C 00 00 00 J
4B 00 22 12 0A 06 0A 12 22 00 00 00 K
4C 00 02 02 02 02 02 02 3E 00 00 00 L
4D 00 22 36 2A 2A 22 22 22 00 00 00 M
4E 00 22 26 2A 32 22 22 22 00 00 00 N
4F 00 3E 22 22 22 22 22 3E 00 00 00 O
50 00 1E 22 22 1E 02 02 02 00 00 00 P
51 00 1C 22 22 22 2A 12 2C 00 00 00 Q
52 00 1E 22 22 1E 0A 12 22 00 00 00 R
53 00 1C 22 02 1C 20 22 1C 00 00 00 S
54 00 3E 08 08 08 08 08 08 00 00 00 T
55 00 22 22 22 22 22 22 1C 00 00 00 U
56 00 22 22 22 14 14 08 08 00 00 00 V
57 00 22 22 22 22 2A 36 22 00 00 00 W

58 00 22 22 14 08 14 22 22 00 00 00 X
59 00 22 22 14 08 08 08 08 00 00 00 Y
5A 00 3E 20 10 08 04 02 3E 00 00 00 Z
5B 00 3C 0A 0A 3E 0A 0A 3A 00 00 00 [
5C 00 1C 22 32 2A 26 22 1C 00 00 00 \
5D 00 1C 14 3E 22 3E 22 22 00 00 00 ]
5E 00 08 1C 2A 08 08 08 08 00 00 00 ^
5F 00 00 00 00 00 00 00 00 7F 00 00 _
60 00 00 00 00 00 00 00 70 70 70 70
61 0F 0F 0F 00 00 00 00 70 70 70 70
62 70 70 70 00 00 00 00 70 70 70 70
63 7F 7F 7F 00 00 00 00 70 70 70 70
64 00 00 00 0F 0F 0F 0F 70 70 70 70
65 0F 0F 0F 0F 0F 0F 0F 70 70 70 70
66 70 70 70 0F 0F 0F 0F 70 70 70 70
67 7F 7F 7F 0F 0F 0F 0F 70 70 70 70
68 00 00 00 70 70 70 70 70 70 70 70
69 0F 0F 0F 70 70 70 70 70 70 70 70
6A 70 70 70 70 70 70 70 70 70 70 70
6B 7F 7F 7F 70 70 70 70 70 70 70 70
6C 00 00 00 7F 7F 7F 7F 70 70 70 70
6D 0F 0F 0F 7F 7F 7F 7F 70 70 70 70
6E 70 70 70 7F 7F 7F 7F 70 70 70 70
6F 7F 7F 7F 7F 7F 7F 7F 70 70 70 70
70 00 00 00 00 00 00 00 7F 7F 7F 7F
71 0F 0F 0F 00 00 00 00 7F 7F 7F 7F
72 70 70 70 00 00 00 00 7F 7F 7F 7F
73 7F 7F 7F 00 00 00 00 7F 7F 7F 7F
74 00 00 00 0F 0F 0F 0F 7F 7F 7F 7F
75 0F 0F 0F 0F 0F 0F 0F 7F 7F 7F 7F
76 70 70 70 0F 0F 0F 0F 7F 7F 7F 7F
77 7F 7F 7F 0F 0F 0F 0F 7F 7F 7F 7F
78 00 00 00 70 70 70 70 7F 7F 7F 7F
79 0F 0F 0F 70 70 70 70 7F 7F 7F 7F
7A 70 70 70 70 70 70 70 7F 7F 7F 7F
7B 7F 7F 7F 70 70 70 70 7F 7F 7F 7F
7C 00 00 00 7F 7F 7F 7F 7F 7F 7F 7F
7D 0F 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F
7E 70 70 70 7F 7F 7F 7F 7F 7F 7F 7F
7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F

```

And other ASCII files I have viewed also appear to valid and sane.

These *.RAW files were processed the same way, and used the same rc01 definition.

```
319488 Mar 6 19:10 D0TEST.RAW
319488 Mar 6 19:10 GAMES.RAW
319488 Mar 6 19:09 MBASIC521.RAW
319488 Sep 22 2017 WORDSTAR.RAW
```

Larry

Re: [HOWTO] - Access CP/M Floppy's via libdisk & cpmtools

by Ikraemer

Posted: 2021-03-16 20:34

Re: [HOWTO] - Access CP/M Floppy's via libdisk & cpmtools

Post by Ikraemer » 2021-03-08 09:00

UPDATE 03-16-2021 - for BIN2IMD Command file

I just found a small problem with the Imagedisk ver 1.18 utility BIN2IMD. The *.B2I file parameters in the track stanzas point to the incorrect data.

As an example I have an Image named RC702F33.IMD and it has the following layout.

```
; 0/0 500 kbps SD 26x128 = 3328 = 0x0D00
; 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
; D D D D D D D D D D D D D D D D D D D D D D
; 0/1 500 kbps DD 26x256 = 6656 = 0x1000
; 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
; D D D00D00D D D D D D D DE5DE5DE5DE5DE5DE5DE5DE5DE5DE
; 1/0 500 kbps DD 15x512
; 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
; D D D D D D D D D D D DE5 D D DE5
; 1/1 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5
; 2/0 D D D D D D D D D D D D D D D
; 2/1 D D D D D D D D D D D D D D D
;
```

If I create a .B2I file named LKF34.B2I and it contains:

```
;
; Example command option file to demonstrate a 40 Cylinder
; double-sided disk which is formatted 10x256 bytes sectors
; at 250kbps FM on the first two Cylinders, and 10x512 byte
; sectors at 250kbps MFM on the remaining 38 Cylinders.
;
;0 N=40 DM=2 SS=256 SM=1-10 /2
;2 DM=5 SS=512 SM=1-10 /2
;
;
;
; 0/0 500 kbps SD 26x128 = 3328 = 0x0D00
; 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
; D D D D D D D D D D D D D D D D D D D D D D
; 0/1 500 kbps DD 26x256 = 6656 = 0x1000
; 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
; D D D00D00D D D D D D D DE5DE5DE5DE5DE5DE5DE5DE5DE5DE
; 1/0 500 kbps DD 15x512
; 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
; D D D D D D D D D D D DE5 D D DE5
; 1/1 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5
```

```
; 1/1 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5
; 2/0 D D D D D D D D D D D D D D D D
; 2/1 D D D D D D D D D D D D D D D
;
;BIN2IMD LKF33.RAW LKF35.IMD LKF34.B2I
;
0 N=77 DM0=0 SS0=128 SM0=1-26 DM1=3 SS1=256 SM1=1-26 /2
1 DM=3 SS=512 SM=1-15 /2
It should create the proper layout, but does not. Instead I get the following:
Assuming 1:1 for Binary output
0/0 500 kbps DD 26x256
  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
  D D D D D D D D D D D D D D D D00D00D D D D D
0/1 500 kbps SD 26x128
  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
  DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5
1/0 500 kbps DD 15x512
  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
  D D D D D D D D D D D D DE5 D D DE5
1/1 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5 DE5
2/0 D D D D D D D D D D D D D D D
2/1 D D D D D D D D D D D D D D D
```

The following changes in the .B2I file will make the software functional

```
;
;DM=0 -> FM SD 500 kbps
;DM=1 -> FM SD 300 kbps
;DM=2 -> FM SD 250 kbps
;DM=3 -> MFM DD 500 kbps
;DM=4 -> MFM DD 300 kbps
;DM=5 -> MFM DD 250 kbps
;
;Until BIN2IMD is fixed the following changes will allow the program to
;Be sure to add the /1 or /2 at the end of the stanzas
;
;FOR CYL 0 SIDE 0 USE DM1=X, SS1=X, SM1=X
;FOR CYL 0 SIDE 1 USE DM0=X, SS0=X, SM0=X
;
;
;
;BIN2IMD LKF33.RAW LKF35..IMD LKF34.B2I
;
0 N=77 DM1=0 SS1=128 SM1=1-26 DM0=3 SS0=256 SM0=1-26 /2
1 DM=3 SS=512 SM=1-15 /2
```

Now, the Image layout is proper.

Thanks.

Larry

Re: [HOWTO] - Access CP/M Floppy's via libdisk & cpmtools

by **lkraemer**

Posted: **2023-02-17 16:53**

Re: [HOWTO] - Access CP/M Floppy's via libdisk & cpmtools

Post by lkraemer » 2023-02-17 10:39

UPDATE 02-17-2023 - Correcting a RAW Image file

I just found a small issue with the IMDU Utility when decoding a *.IMD file. Depending on the Original image there may be odd layouts of the .RAW images. Here are a few samples:
\$./imdu DIAG_204.IMD DIAG_204.RAW -B -E -D X1=0,39 >> DIAG_204.TXT

IMageDisk Utility 1.18-devel / Jun 7 2018

IMD 1.17: 25/12/2010 8:39:39

2.04

diagnostics

Assuming 1:1 for Binary output

```
0/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    D D D D D D D D D D
0/1 250 kbps SD 1x256
    1
HD: 0
    D
1/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    D D D D D D D D D D
1/1 250 kbps SD 1x256
    1
HD: 0
    D
2/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    D D D D D D D D D D
2/1 250 kbps SD 1x256
    1
HD: 0
    D
3/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    De5 D De5 De5 De5 De5 D De5 D D
3/1 250 kbps SD 1x256
    1
HD: 0
    D
4/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    D D D D D D D D D D
4/1 250 kbps SD 1x256
    1
HD: 0
    D
5/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    D D D D D D D D D D
5/1 250 kbps SD 1x256
    1
HD: 0
    D
6/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    D D D D De5 D De5 D De5 D
6/1 250 kbps SD 1x256
    1
```

```

~
HD: 0
D
7/0 250 kbps SD 10x256
  2 3 4 5 6 7 8 9 10 1
  D D D D D D D D D D
7/1 250 kbps SD 1x256
  1
HD: 0
D
8/0 250 kbps SD 10x256
  2 3 4 5 6 7 8 9 10 1
  D D D D D D D D D D
8/1 250 kbps SD 1x256
  1
HD: 0
D
9/0 250 kbps SD 10x256
  2 3 4 5 6 7 8 9 10 1
  D D D D D D D D D D
9/1 250 kbps SD 1x256
  1
HD: 0
D
10/0 250 kbps SD 10x256
  2 3 4 5 6 7 8 9 10 1
  D De5 D De5 D D D D D De5
10/1 250 kbps SD 1x256
  1
HD: 0
D
11/0 250 kbps SD 10x256
  2 3 4 5 6 7 8 9 10 1
  D D D D D D D D D D
11/1 250 kbps SD 1x256
  1
HD: 0
D
12/0 250 kbps SD 10x256
  2 3 4 5 6 7 8 9 10 1
  D D D D D D D D D D
12/1 250 kbps SD 1x256
  1
HD: 0
D
13/0 250 kbps SD 10x256
  2 3 4 5 6 7 8 9 10 1
  D D D D D D D D D D
13/1 250 kbps SD 1x256
  1
HD: 0
D
14/0 250 kbps SD 10x256
  2 3 4 5 6 7 8 9 10 1
  De5 D De5 D De5 D De5 De5 De5 D
14/1 250 kbps SD 1x256
  1
...
~

```

```

HD: 0
D
15/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    D D D D De5 D D D D D
15/1 250 kbps SD 1x256
    1
HD: 0
D
16/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    D D D D D D D D D D
16/1 250 kbps SD 1x256
    1
HD: 0
D
17/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    D D D D D D D D D D
17/1 250 kbps SD 1x256
    1
HD: 0
D
18/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    D D D D D D D D D D
18/1 250 kbps SD 1x256
    1
HD: 0
D
19/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    D D D D D D D D D D
19/1 250 kbps SD 1x256
    1
HD: 0
D
20/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    D D D D D D D D D D
20/1 250 kbps SD 1x256
    1
HD: 0
D
21/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    D D D D D D D D De5 D
21/1 250 kbps SD 1x256
    1
HD: 0
D
22/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    D De5 D D D D De5 D De5 De5
22/1 250 kbps SD 1x256
    1
HD: 0

```

```

D
23/0 250 kbps SD 10x256
      2 3 4 5 6 7 8 9 10 1
      D D De5 D De5 D D D D D
23/1 250 kbps SD 1x256
      1
      HD: 0
      D
24/0 250 kbps SD 10x256
      2 3 4 5 6 7 8 9 10 1
      D D D D D D D D D D
24/1 250 kbps SD 1x256
      1
      HD: 0
      D
25/0 250 kbps SD 10x256
      2 3 4 5 6 7 8 9 10 1
      De5 De5 De5 De5 De5 De5 De5 D De5 D
25/1 250 kbps SD 1x256
      1
      HD: 0
      D
26/0 250 kbps SD 10x256
      2 3 4 5 6 7 8 9 10 1
      D De5 D D De5 D De5 D De5 De5
26/1 250 kbps SD 1x256
      1
      HD: 0
      D
27/0 250 kbps SD 10x256
      2 3 4 5 6 7 8 9 10 1
      D D D D De5 D D D D D
27/1 250 kbps SD 1x256
      1
      HD: 0
      D
28/0 250 kbps SD 10x256
      2 3 4 5 6 7 8 9 10 1
      De5 D D De5 D De5 D De5 D D
28/1 250 kbps SD 1x256
      1
      HD: 0
      D
29/0 250 kbps SD 10x256
      2 3 4 5 6 7 8 9 10 1
      D D D D D D D D De5 D
29/1 250 kbps SD 1x256
      1
      HD: 0
      D
30/0 250 kbps SD 10x256
      2 3 4 5 6 7 8 9 10 1
      D De5 D D D D D D De5
30/1 250 kbps SD 1x256
      1
      HD: 0
      n

```

```

~
31/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    De5 D De5 D De5 De5 D De5 D D
31/1 250 kbps SD 1x256
    1
    HD: 0
    D
32/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    D D D D D D D D D D
32/1 250 kbps SD 1x256
    1
    HD: 0
    D
33/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    D De5 D De5 D De5 D D D D
33/1 250 kbps SD 1x256
    1
    HD: 0
    D
34/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    De5 D De5 De5 De5 De5 De5 De5 De5 D
34/1 250 kbps SD 1x256
    1
    HD: 0
    D
35/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
35/1 250 kbps SD 1x256
    1
    HD: 0
    D
36/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
36/1 250 kbps SD 1x256
    1
    HD: 0
    D
37/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
37/1 250 kbps SD 1x256
    1
    HD: 0
    D
38/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
38/1 250 kbps SD 1x256
    1
    HD: 0
    D
39/0 250 kbps SD 10x256
    2 3 4 5 6 7 8 9 10 1
    De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
39/1 250 kbps SD 1x256
    1
    HD: 0
    D

```

```

39/0 250 kbps SD 10x256
      2 3 4 5 6 7 8 9 10 1
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
39/1 250 kbps SD 1x256
      1
      HD: 0
          D

```

80 tracks(40/40), 440 sectors (113 Compressed)

and

```
$ ./imdu MILSTN-2.IMD MILSTN-2.RAW -B -E -D X1=0,39 >> MILSTN-2.TXT
```

IMageDisk Utility 1.18-devel / Jun 7 2018

IMD 1.17: 2/09/2010 16:15:38

Milestone - Diskette # 2 - Sample Data Files

Assuming 1:1 for Binary output

```

0/0 250 kbps SD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
0/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      D D D Df6 Df6 Df6 Df6 Df6 Df6
1/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
1/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
2/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
2/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
3/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      D De5 D De5 De5 De5 De5 D De5 D
3/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
4/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 De5 De5 D De5 D De5 D De5 D
4/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
5/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 D De5 De5 De5 De5 De5 De5 D De5
5/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
6/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 D De5 De5 D De5 D De5 D De5
6/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6

```

```

7/0 250 kbps DD 10x256
  1  2  3  4  5  6  7  8  9 10
  D  De5 D  De5 De5 De5 De5 D  De5 D
7/1 250 kbps SD 9x512
  1  2  3  4  5  6  7  8  9
  Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
8/0 250 kbps DD 10x256
  1  2  3  4  5  6  7  8  9 10
  D  D  D  D  D  D  D  De5 D  De5
8/1 250 kbps SD 9x512
  1  2  3  4  5  6  7  8  9
  Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
9/0 250 kbps DD 10x256
  1  2  3  4  5  6  7  8  9 10
  De5 D  De5 De5 De5 De5 De5 De5 D  De5
9/1 250 kbps SD 9x512
  1  2  3  4  5  6  7  8  9
  Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
10/0 250 kbps DD 10x256
  1  2  3  4  5  6  7  8  9 10
  De5 D  De5 D  D  De5 D  De5 D  De5
10/1 250 kbps SD 9x512
  1  2  3  4  5  6  7  8  9
  Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
11/0 250 kbps DD 10x256
  1  2  3  4  5  6  7  8  9 10
  D  De5 D  De5 De5 De5 De5 D  De5 D
11/1 250 kbps SD 9x512
  1  2  3  4  5  6  7  8  9
  Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
12/0 250 kbps DD 10x256
  1  2  3  4  5  6  7  8  9 10
  D  De5 D  D  D  D  D  De5 De5 De5
12/1 250 kbps SD 9x512
  1  2  3  4  5  6  7  8  9
  Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
13/0 250 kbps DD 10x256
  1  2  3  4  5  6  7  8  9 10
  De5 D  De5 De5 De5 De5 De5 De5 D  De5
13/1 250 kbps SD 9x512
  1  2  3  4  5  6  7  8  9
  Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
14/0 250 kbps DD 10x256
  1  2  3  4  5  6  7  8  9 10
  De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
14/1 250 kbps SD 9x512
  1  2  3  4  5  6  7  8  9
  Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
15/0 250 kbps DD 10x256
  1  2  3  4  5  6  7  8  9 10
  De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
15/1 250 kbps SD 9x512
  1  2  3  4  5  6  7  8  9
  Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
16/0 250 kbps DD 10x256
  1  2  3  4  5  6  7  8  9 10
  De5 De5 De5 De5 De5 De5 De5 De5 De5 De5

```

```

250 250 250 250 250 250 250 250 250
16/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
17/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
17/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
18/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
18/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
19/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
19/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
20/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
20/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
21/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
21/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
22/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
22/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
23/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
23/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
24/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
24/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
25/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
25/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      250 250 250 250 250 250 250 250

```

```

      DTb DTb DTb DTb DTb DTb DTb DTb DTb
26/0 250 kbps DD 10x256
      1  2  3  4  5  6  7  8  9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
26/1 250 kbps SD 9x512
      1  2  3  4  5  6  7  8  9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
27/0 250 kbps DD 10x256
      1  2  3  4  5  6  7  8  9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
27/1 250 kbps SD 9x512
      1  2  3  4  5  6  7  8  9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
28/0 250 kbps DD 10x256
      1  2  3  4  5  6  7  8  9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
28/1 250 kbps SD 9x512
      1  2  3  4  5  6  7  8  9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
29/0 250 kbps DD 10x256
      1  2  3  4  5  6  7  8  9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
29/1 250 kbps SD 9x512
      1  2  3  4  5  6  7  8  9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
30/0 250 kbps DD 10x256
      1  2  3  4  5  6  7  8  9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
30/1 250 kbps SD 9x512
      1  2  3  4  5  6  7  8  9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
31/0 250 kbps DD 10x256
      1  2  3  4  5  6  7  8  9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
31/1 250 kbps SD 9x512
      1  2  3  4  5  6  7  8  9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
32/0 250 kbps DD 10x256
      1  2  3  4  5  6  7  8  9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
32/1 250 kbps SD 9x512
      1  2  3  4  5  6  7  8  9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
33/0 250 kbps DD 10x256
      1  2  3  4  5  6  7  8  9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
33/1 250 kbps SD 9x512
      1  2  3  4  5  6  7  8  9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
34/0 250 kbps DD 10x256
      1  2  3  4  5  6  7  8  9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
34/1 250 kbps SD 9x512
      1  2  3  4  5  6  7  8  9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
35/0 250 kbps DD 10x256
      1  2  3  4  5  6  7  8  9 10

```

```

      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
35/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
36/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
36/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
37/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
37/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
38/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 D
38/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6
39/0 250 kbps DD 10x256
      1 2 3 4 5 6 7 8 9 10
      De5 De5 De5 De5 De5 De5 De5 De5 De5 De5
39/1 250 kbps SD 9x512
      1 2 3 4 5 6 7 8 9
      Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6 Df6

```

80 tracks(40/40), 760 sectors (711 Compressed)

Notice that Side 1 of both .RAW files is very much different than we are expecting for a CP/M Floppy.

These type extractions need to somehow block the incorrect information contained in these tracks.

The IMDU utility has that capability:

Use: `IMDU image [[merge-image] [output-image]] [options]`

```

opts:  -B          - output Binary image
        -C          - Compress "all-same" sectors
        -D          - display track/sector Detail
        -E          - Expand compressed sectors to full dat
        -M          - ignore Mode difference in merge/compa
        -NB         - force Non-Bad data
        -ND         - force Non-Deleted data
        -Q          - Quiet: supress warnings
        -Y          - auto-Yes (no overwrite prompt)
        AC=file[.TXT] - Append Comment from file      [none]
        EC=file[.TXT] - Extract Comment to file        [none]
        F=xx        - missing sector Fill value        [00]
        IL=[1-99]   - reInterLeave(blank=BestGuess)    [As rea
        RC=file[.TXT] - Replace Comment from file      [none]
        T2=250/300/500 - 250khz Translate              [250]
        T3=250/300/500 - 300khz Translate              [300]
        T5=250/300/500 - 500khz Translate              [500]
        X=track[,to_track] - eXclude entire track(s)  [None]
        X0=track[,to_track] - eXclude track(s) side 0 only [None]
        X1=track[,to_track] - eXclude track(s) side 1 only [None]

```

By using the X1 switch we can strip Side 1 from the .RAW images.

```
$ ./imdu DIAG_204.IMD DIAG_204.RAW -B -E -D X1=0,39 >> DIAG_204.TXT
```

```
$ ./imdu MILSTN-2.IMD MILSTN-2.RAW -B -E -D X1=0,39 >> MILSTN-2.TXT
```

The *.RAW Images now have ONLY SIDE 0 contained in the .RAW image(s).

This allows cpmtools to correctly get Directory listings and extract valid files.

Larry

Re: [HOWTO] - Access CP/M Floppy's via libdsk & cpmtools

by Ikraemer

Posted: **2023-04-24 13:27**

Post by Ikraemer » 2023-02-17 10:53

Re: [HOWTO] - Access CP/M Floppy's via libdsk & cpmtools

Post by Ikraemer » 2023-04-24 08:00

UPDATE 04-29-2023 - How to deal with complemented (*.IMD, *.RAW) files for CP/M Images.

If you have a CP/M Computer that uses **COMPLEMENTED** Data Files such as these Computers:

Columbia Commander 964 - DSDD 48 tpi 5.25" - 512 x 10

Columbia 1600 - DSDD 96 tpi 5.25" - 512 x 10

Columbia M64 - SSDD 48 tpi 5.25" - 512 x 10

Compustar Model 30 - DSDD 48 tpi 5.25" - 512 x 10

Compustar 30 Super IOS - DSDD 48 tpi 5.25" - 512 x 10

Michels and Kleberhoff CP/M 3 - DSDD 96 tpi 5.25" - 512 x 10

Michels and Kleberhoff CP/M 3 - DSDD 48 tpi 5.25" - 512 x 10

Sharp MZ-80 - DSDD 48 tpi 5.25" - 512 x 10

Sharp MZ-80B - DSDD 48 tpi 5.25" - 512 x 10

Shelton SIG/NET 2 - SSDD 48 tpi 5.25" - 512 x 10

Shelton SIG/NET 2 - DSDD 48 tpi 5.25" - 512 x 10

Shelton SIG/NET 2 - DSDD 96 tpi 5.25" - 512 x 10

Superbrain JR - SSDD 48 tpi 5.25" - 512 x 10

Superbrain 40 track - SSDD 48 tpi 5.25" - 512 x 10

Superbrain QD - DSDD 48 tpi 5.25" - 512 x 10

Superbrain II - DSDD 96 tpi 5.25" - 512 x 10

Superbrain - SSDD 48 tpi 5.25" - 128 x 30

Wren Executive - SSDD 96 tpi 5.25" - 512 x 10

The following code will be of interest to you.

I just finished creating a Perl program that will complement the Binary Bits of a Floppy Image. This allows

you to use CP/M COMPLEMENTED Image files to extract/insert CP/M files as needed with cpmtools or cpmtools

built with libdsk. Note that cpmtools can NOT access COMPLEMENTED Data Files.

Remember that libdsk can

also write these images to a Physical floppy, assuming you have the proper libdsk definition.

The complement.pl program is:

```
#!/usr/bin/perl
```

```
use 5.010;
```

```
use strict;
```

```
use warnings;
```

```

my ($infile, $outfile) = @ARGV;
die "Usage: $0 INFILE OUTFILE\n" if not $outfile;

open my $in, '<', $infile or die;
binmode $in;

my $cont = '';

while (1) {
    my $success = read $in, $cont, 512, length($cont);
    die $! if not defined $success;
    last if not $success;
}
close $in;

open my $out, '>', $outfile or die;
binmode $out;

print $out ~$cont ;
close $out;

```

```

say length($cont);
say -s $infile;
say -s $outfile;
To prepare a *.IMD file I create a Batch file named A.BAT
To Prepare a *.TD0 file I create a Batch file named B.BAT

```

The DOSBox tutorial at: viewtopic.php?p=609667&hilit=%5BHOWTO%5D+DOSBox#p609667

These files are executed in DOSBox running on Debian Linux. I also copied TD02IMD, BIN2IMD, IMDA, and IMDU to the directory where the batch file is located, then I mount that directory.

From within DOSBox I mount the subdirectory.

```
mount c: ~/path/to/IMDs/superbrain
```

```
c:
```

```
A.BAT
```

A.BAT contains:

```

IMDA CB00T.IMD > CB00T.TXT
IMDA CSYSGEN.IMD > CSYSGEN.TXT
IMDA CUTILS.IMD > CUTILS.TXT
IMDA ZHELP.IMD > ZHELP.TXT
IMDA ZMAST.IMD > ZMAST.TXT
IMDA ZUTILS.IMD > ZUTILS.TXT

```

```

IMDU CB00T.IMD CB00T.RAW /B /E /D >> CB00T.TXT
IMDU CSYSGEN.IMD CSYSGEN.RAW /B /E /D >> CSYSGEN.TXT
IMDU CUTILS.IMD CUTILS.RAW /B /E /D >> CUTILS.TXT
IMDU ZHELP.IMD ZHELP.RAW /B /E /D >> ZHELP.TXT
IMDU ZMAST.IMD ZMAST.RAW /B /E /D >> ZMAST.TXT
IMDU ZUTILS.IMD ZUTILS.RAW /B /E /D >> ZUTILS.TXT

```

B.BAT contains:

```
TD02IMD CB00T.TD0 > CB00T.TXT
```

```
IMDA CB00T.IMD >> CB00T.TXT
```

```
IMDU CB00T.IMD CB00T.RAW /B /E /D >> CB00T.TXT
```

When the batch files are done, I unmount the directory, then exit DOSBox.

```
Z:
mount -u c
exit
```

This will create the *.RAW files (Sector Dump's) from the *.IMD's that will need their Binary Data complemented.

```
CUTILS.RAW
CSYSGEN.RAW
CB00T.RAW
```

It's at this point that I copy the *.TXT files to *.R.TXT files. (complemented binary data)

Now, process each file with Perl.

```
$ perl complement.pl CB00T.RAW CB00TR.RAW
$ perl complement.pl CSYSGEN.RAW CSYSGENR.RAW
$ perl complement.pl CUTILS.RAW CUTILSR.RAW
```

cpmtools and libdisk can now access these files to extract/insert CP/M files as needed.

The 22DISK Definitions that have COMPLEMENTED Data are:

```
BEGIN COLA Columbia Commander 964 - DSDD 48 tpi 5.25" - 512 x 10
INSERT 22DISK DEFINITION HERE
END
```

```
# cpmtools
# COLA Columbia Commander 964 - DSDD 48 tpi 5.25" - 512 x 10
diskdef cola
    seclen 512
    tracks 80
    sectrk 10
    blocksize 2048
    maxdir 64
    skew 1
    boottrk 2
# data should be complemented
# cpmtools doesn't do that yet! libdisk works
    os 2.2
end
```

```
# libdisk
[cola]
description = COLA Columbia Commander 964 - DSDD 48 tpi 5.25" - 512 x
sides = columbia
cylinders = 80
heads = 2
seclen = 512
sectors = 10
secbase = 1
Complement=Y
datarate = DD
```

```
# Flashfloppy/GOTEK
[cola]
cyls = 80
heads = 2
secs = 10
interleave = 1
bps = 512
```

```
id = 1
#rpm = 360
rpm = 300
rate = 250
mode = mfm
iam = no

BEGIN COLB  Columbia 1600 - DSDD 96 tpi 5.25" - 512 x 10
INSERT 22DISK DEFINITION HERE
END

# cpmtools
# COLB  Columbia 1600 - DSDD 96 tpi 5.25" - 512 x 10
diskdef colb
    seclen 512
    tracks 160
    sectrk 10
    blocksize 4096
    maxdir 256
    skew 1
    boottrk 2
# data should be complemented
# cpmtools doesn't do that yet!  libdsk works
    os 2.2
end

# libdsk
[colb]
description = COLB  Columbia 1600 - DSDD 96 tpi 5.25" - 512 x 10
sides = columbia
cylinders = 160
heads = 2
seclen = 512
sectors = 10
secbase = 1
Complement=Y
datarate = DD

# Flashfloppy/GOTEK
[colb]
cyls = 160
heads = 2
secs = 10
interleave = 1
bps = 512
id = 1
#rpm = 360
rpm = 300
rate = 250
mode = mfm
iam = no

BEGIN COLC  Columbia M64 - SSDD 48 tpi 5.25" - 512 x 10
INSERT 22DISK DEFINITION HERE
END

# cpmtools
```

```
# COLC Columbia M64 - SSDD 48 tpi 5.25" - 512 x 10
diskdef colc
    seclen 512
    tracks 40
    sectrk 10
    blocksize 1024
    maxdir 128
    skew 1
    boottrk 2
# data should be complemented
# cpmtools doesn't do that yet! libdisk works
    os 2.2
end

# libdisk
[colc]
description = COLC Columbia M64 - SSDD 48 tpi 5.25" - 512 x 10
cylinders = 40
heads = 1
seclen = 512
sectors = 10
secbase = 1
Complement=Y
datarate = DD

# Flashfloppy/GOTEK
[colc]
cyls = 40
heads = 1
secs = 10
interleave = 1
bps = 512
id = 1
#rpm = 360
rpm = 300
rate = 250
mode = mfm
iam = no

BEGIN COM3X Compustar Model 30 - DSDD 48 tpi 5.25" - 512 x 10
INSERT 22DISK DEFINITION HERE
END

# cpmtools
# COM3X Compustar Model 30 - DSDD 48 tpi 5.25" - 512 x 10
diskdef com3x
    seclen 512
    tracks 70
    sectrk 10
    blocksize 2048
    maxdir 64
    skew 2
    boottrk 2
# data should be complemented
# cpmtools doesn't do that yet! libdisk works
    os 2.2
end
```

```

~..

```

```

# libdsk
[com3x]
description = COM3X  Compustar Model 30 - DSDD 48 tpi 5.25" - 512 x 10
sides = eagle
cylinders = 70
heads = 2
secsize = 512
sectors = 10
secbase = 1
Complement=Y
datarate = DD

```

```

# Flashfloppy/GOTEK
[com3x]
cyls = 70
heads = 2
secs = 10
interleave = 2
bps = 512
id = 1
#rpm = 360
rpm = 300
rate = 250
mode = mfm
iam = no

```

```

BEGIN COM4X  Compustar 30 Super IOS - DSDD 48 tpi 5.25" - 512 x 10
INSERT 22DISK DEFINITION HERE
END

```

```

# cpmtools
# COM4X  Compustar 30 Super IOS - DSDD 48 tpi 5.25" - 512 x 10
diskdef com4x
    seclen 512
    tracks 70
    sectrk 10
    blocksize 2048
    maxdir 128
    skew 2
    boottrk 2
# data should be complemented
# cpmtools doesn't do that yet!  libdsk works
    os 2.2
end

```

```

# libdsk
#ORDER CYLINDERS
[com4x]
description = COM4X  Compustar 30 Super IOS - DSDD 48 tpi 5.25" - 512 x
sides = outback
cylinders = 70
heads = 2
secsize = 512
sectors = 10
secbase = 1
~

```

```
Complement=Y
datarate = DD

# Flashfloppy/GOTEK
[com4x]
cyls = 70
heads = 2
secs = 10
interleave = 2
bps = 512
id = 1
#rpm = 360
rpm = 300
rate = 250
mode = mfm
iam = no

BEGIN MIC7 Michels and Kleberhoff CP/M 3 - DSDD 96 tpi 5.25" - 512 x 1
INSERT 22DISK DEFINITION HERE
END

# cpmtools
# MIC7 Michels and Kleberhoff CP/M 3 - DSDD 96 tpi 5.25" - 512 x 10
diskdef mic7
    seclen 512
    tracks 160
    sectrk 10
    blocksize 2048
    maxdir 128
    skew 2
    boottrk 2
# data should be complemented
# cpmtools doesn't do that yet! libdisk works
    os 2.2
end

# libdisk
[mic7]
description = MIC7 Michels and Kleberhoff CP/M 3 - DSDD 96 tpi 5.25" -
sides = eagle
cylinders = 160
heads = 2
seclen = 512
sectors = 10
seclen = 1
Complement=Y
datarate = DD

# Flashfloppy/GOTEK
[mic7]
cyls = 160
heads = 2
secs = 10
interleave = 2
bps = 512
id = 1
#rpm = 360
```

```
rpm = 300
rate = 250
mode = mfm
iam = no
```

```
BEGIN MIC8 Michels and Kleberhoff CP/M 3 - DSDD 48 tpi 5.25" - 512 x 1
INSERT 22DISK DEFINITION HERE
END
```

```
# cpmtools
# MIC8 Michels and Kleberhoff CP/M 3 - DSDD 48 tpi 5.25" - 512 x 10
diskdef mic8
    seclen 512
    tracks 70
    sectrk 10
    blocksize 2048
    maxdir 64
    skew 2
    boottrk 2
# data should be complemented
# cpmtools doesn't do that yet! libdisk works
    os 2.2
end
```

```
# libdisk
#ORDER CYLINDERS
[mic8]
description = MIC8 Michels and Kleberhoff CP/M 3 - DSDD 48 tpi 5.25" -
sides = outback
cylinders = 70
heads = 2
secssize = 512
sectors = 10
secbase = 1
Complement=Y
datarate = DD
```

```
# Flashfloppy/GOTEK
[mic8]
cyls = 70
heads = 2
secs = 10
interleave = 2
bps = 512
id = 1
#rpm = 360
rpm = 300
rate = 250
mode = mfm
iam = no
```

```
BEGIN SHA1 Sharp MZ-80 - DSDD 48 tpi 5.25" - 512 x 10
INSERT 22DISK DEFINITION HERE
END
```

```
# cpmtools
# MIC8 Michels and Kleberhoff CP/M 3 - DSDD 48 tpi 5.25" - 512 x 10
```

```
# SHA1 Sharp MZ-80 - DSDD 48 tpi 5.25" - 512 x 10
diskdef sha1
  seclen 512
  tracks 80
  sectrk 10
  blocksize 2048
  maxdir 128
  skew 1
  boottrk 2
# data should be complemented
# cpmtools doesn't do that yet! libdisk works
  os 2.2
end

# libdisk
[sha1]
description = SHA1 Sharp MZ-80 - DSDD 48 tpi 5.25" - 512 x 10
sides = alt
cylinders = 80
heads = 2
seclen = 512
sectors = 10
seclen = 1
Complement=Y
datarate = DD

# Flashfloppy/GOTEK
[sha1]
cyls = 80
heads = 2
secs = 10
interleave = 1
bps = 512
id = 1
#rpm = 360
rpm = 300
rate = 250
mode = mfm
iam = no

BEGIN SHA2 Sharp MZ-80B - DSDD 48 tpi 5.25" - 512 x 10
INSERT 22DISK DEFINITION HERE
END

# cpmtools
# SHA2 Sharp MZ-80B - DSDD 48 tpi 5.25" - 512 x 10
diskdef sha2
  seclen 512
  tracks 80
  sectrk 10
  seclen 11
  blocksize 2048
  maxdir 128
  skew 1
  boottrk 2
# data should be complemented
# cpmtools doesn't do that yet! libdisk works
```

```
    os 2.2
end

# libdisk
[sha2]
description = SHA2  Sharp MZ-80B - DSDD 48 tpi 5.25" - 512 x 10
sides = alt
cylinders = 80
heads = 2
secsize = 512
sectors = 10
secbase = 11
Complement=Y
datarate = DD

# Flashfloppy/G0TEK
[sha2]
cyls = 80
heads = 2
secs = 10
interleave = 1
bps = 512
id = 1
#rpm = 360
rpm = 300
rate = 250
mode = mfm
iam = no

BEGIN SHL1  Shelton SIG/NET 2 - SSDD 48 tpi 5.25" - 512 x 10
INSERT 22DISK DEFINITION HERE
END

# cpmtools
# SHL1  Shelton SIG/NET 2 - SSDD 48 tpi 5.25" - 512 x 10
diskdef shl1
    seclen 512
    tracks 40
    sectrk 10
    blocksize 2048
    maxdir 64
    skew 2
    boottrk 2
# data should be complemented
# cpmtools doesn't do that yet!  libdisk works
    os 2.2
end

# libdisk
[shl1]
description = SHL1  Shelton SIG/NET 2 - SSDD 48 tpi 5.25" - 512 x 10
cylinders = 40
heads = 1
secsize = 512
sectors = 10
secbase = 1
Complement=Y
```

```

complement = 1
datarate = DD

# Flashfloppy/GOTEK
[shl1]
cyls = 40
heads = 1
secs = 10
interleave = 2
bps = 512
id = 1
#rpm = 360
rpm = 300
rate = 250
mode = mfm
iam = no

BEGIN SHL2  Shelton SIG/NET 2 - DSDD 48 tpi 5.25" - 512 x 10
INSERT 22DISK DEFINITION HERE
END

# cpmtools
# SHL2  Shelton SIG/NET 2 - DSDD 48 tpi 5.25" - 512 x 10
diskdef shl2
    seclen 512
    tracks 80
    sectrk 10
    blocksize 2048
    maxdir 64
    skew 2
    boottrk 2
# data should be complemented
# cpmtools doesn't do that yet!  libdisk works
    os 2.2
end

# libdisk
[shl2]
description = SHL2  Shelton SIG/NET 2 - DSDD 48 tpi 5.25" - 512 x 10
sides = alt
cylinders = 80
heads = 2
seclen = 512
sectors = 10
secbase = 1
Complement=Y
datarate = DD

#Flashfloppy/ GOTEK
[shl2]
cyls = 80
heads = 2
secs = 10
interleave = 2
bps = 512
id = 1
#rpm = 360
    ~~~

```

```
rpm = 300
rate = 250
mode = mfm
iam = no
```

```
BEGIN SHL3  Shelton SIG/NET 2 - DSDD 96 tpi 5.25" - 512 x 10
INSERT 22DISK DEFINITION HERE
END
```

```
# cpmtools
# SHL3  Shelton SIG/NET 2 - DSDD 96 tpi 5.25" - 512 x 10
diskdef shl3
    seclen 512
    tracks 160
    sectrk 10
    blocksize 2048
    maxdir 256
    skew 2
    boottrk 2
# data should be complemented
# cpmtools doesn't do that yet!  libdsk works
    os 2.2
end
```

```
# libdsk
[shl3]
description = SHL3  Shelton SIG/NET 2 - DSDD 96 tpi 5.25" - 512 x 10
sides = alt
cylinders = 160
heads = 2
seclen = 512
sectors = 10
seclen = 1
Complement=Y
datarate = DD
```

```
# Flashfloppy/GOTEK
[shl3]
cyls = 160
heads = 2
secs = 10
interleave = 2
bps = 512
id = 1
#rpm = 360
rpm = 300
rate = 250
mode = mfm
iam = no
```

```
BEGIN SUP1  Superbrain JR - SSDD 48 tpi 5.25" - 512 x 10
INSERT 22DISK DEFINITION HERE
END
```

```
# cpmtools
# SUP1  Superbrain JR - SSDD 48 tpi 5.25" - 512 x 10
diskdef sup1
```

```
    seclen 512
    tracks 35
    sectrk 10
    blocksize 2048
    maxdir 64
    skew 2
    boottrk 2
# data should be complemented
# cpmtools doesn't do that yet!  libdisk works
    os 2.2
end

# libdisk
[sup1]
description = SUP1  Superbrain JR - SSDD 48 tpi 5.25" - 512 x 10
cylinders = 35
heads = 1
seclen = 512
sectors = 10
seclen = 1
Complement=Y
datarate = DD

# Flashfloppy/GOTEK
[sup1]
cyls = 35
heads = 1
secs = 10
interleave = 2
bps = 512
id = 1
#rpm = 360
rpm = 300
rate = 250
mode = mfm
iam = no

BEGIN SUP2  Superbrain 40 track - SSDD 48 tpi 5.25" - 512 x 10
INSERT 22DISK DEFINITION HERE
END

# cpmtools
# SUP2  Superbrain 40 track - SSDD 48 tpi 5.25" - 512 x 10
diskdef sup2
    seclen 512
    tracks 40
    sectrk 10
    blocksize 2048
    maxdir 64
    skew 2
    boottrk 2
# data should be complemented
# cpmtools doesn't do that yet!  libdisk works
    os 2.2
end

# libdisk
```

```

" 22DISK
[sup2]
description = SUP2  Superbrain 40 track - SSDD 48 tpi 5.25" - 512 x 10
cylinders = 40
heads = 1
secssize = 512
sectors = 10
secbase = 1
Complement=Y
datarate = DD

# Flashfloppy/GOTEK
[sup2]
cyls = 40
heads = 1
secs = 10
interleave = 2
bps = 512
id = 1
#rpm = 360
rpm = 300
rate = 250
mode = mfm
iam = no

BEGIN SUP3  Superbrain QD - DSDD 48 tpi 5.25" - 512 x 10
INSERT 22DISK DEFINITION HERE
END

# cpmtools
# SUP3  Superbrain QD - DSDD 48 tpi 5.25" - 512 x 10
diskdef sup3
    seclen 512
    tracks 70
    sectrk 10
    blocksize 2048
    maxdir 64
    skew 2
    boottrk 2
# data should be complemented
# cpmtools doesn't do that yet!  libdisk works
    os 2.2
end

# libdisk
[sup3]
description = SUP3  Superbrain QD - DSDD 48 tpi 5.25" - 512 x 10
sides = outback
cylinders = 70
heads = 2
secssize = 512
sectors = 10
secbase = 1
Complement=Y
datarate = DD

# Flashfloppy/GOTEK

```

```
[sup3]
cyls = 70
heads = 2
secs = 10
interleave = 2
bps = 512
id = 1
#rpm = 360
rpm = 300
rate = 250
mode = mfm
iam = no
```

```
BEGIN SUP4 Superbrain II - DSDD 96 tpi 5.25" - 512 x 10
INSERT 22DISK DEFINITION HERE
END
```

```
# cpmtools
# SUP4 Superbrain II - DSDD 96 tpi 5.25" - 512 x 10
diskdef sup4
    seclen 512
    tracks 160
    sectrk 10
    blocksize 4096
    maxdir 128
    skew 2
    boottrk 3
# data should be complemented
# cpmtools doesn't do that yet! libdsk works
    os 2.2
end
```

```
# libdsk
[sup4]
description = SUP4 Superbrain II - DSDD 96 tpi 5.25" - 512 x 10
sides = outback
cylinders = 160
heads = 2
seclen = 512
sectors = 10
seclen = 1
Complement=Y
datarate = DD
```

```
# Flashfloppy/GOTEK
[sup4]
cyls = 160
heads = 2
secs = 10
interleave = 2
bps = 512
id = 1
#rpm = 360
rpm = 300
rate = 250
mode = mfm
iam = no
```

```
BEGIN SUP5 Superbrain - SSDD 48 tpi 5.25" - 128 x 30
INSERT 22DISK DEFINITION HERE
END

# cpmtools
# SUP5 Superbrain - SSDD 48 tpi 5.25" - 128 x 30
diskdef sup5
    seclen 128
    tracks 40
    sectrk 30
    blocksize 1024
    maxdir 64
    skew 3
    boottrk 2
# data should be complemented
# cpmtools doesn't do that yet! libdisk works
    os 2.2
end

# libdisk
[sup5]
description = SUP5 Superbrain - SSDD 48 tpi 5.25" - 128 x 30
cylinders = 40
heads = 1
seclen = 128
sectors = 30
seclen = 1
Complement=Y
datarate = DD

# Flashfloppy/GOTEK
[sup5]
cyls = 40
heads = 1
secs = 30
interleave = 3
bps = 128
id = 1
#rpm = 360
rpm = 300
rate = 250
mode = mfm
iam = no

BEGIN WRE1 Wren Executive - SSDD 96 tpi 5.25" - 512 x 10
INSERT 22DISK DEFINITION HERE
END

# cpmtools
# WRE1 Wren Executive - SSDD 96 tpi 5.25" - 512 x 10
diskdef wre1
    seclen 512
    tracks 40
    sectrk 10
    blocksize 2048
    maxdir 64
```

```

        skew 2
        boottrk 2
# data should be complemented
# cpmtools doesn't do that yet!  libdisk works
    os 2.2
end

# libdisk
[wre1]
description = WRE1  Wren Executive - SSDD 96 tpi 5.25" - 512 x 10
cylinders = 40
heads = 1
secssize = 512
sectors = 10
secbase = 1
Complement=Y
datarate = DD

# Flashfloppy/GOTEK
[wre1]
cyls = 40
heads = 1
secs = 10
interleave = 2
bps = 512
id = 1
#rpm = 360
rpm = 300
rate = 250
mode = mfm
iam = no

```

In the above example, I will be using the SUP3 definition.

```
$ cpm1s -f sup3 -D CB00TR.RAW
```

Name	Bytes	Recs	Attr	update	create
64KTEST .COM	2K	4			
ASM .COM	8K	64			
CONFIGUR.COM	10K	73			
CS30CPM .COM	12K	96			
DATE .COM	2K	5			
DDT .COM	6K	38			
DUMP .COM	2K	5			
ED .COM	8K	53			
F .COM	2K	8			
HEXDUMP .COM	2K	12			
LOAD .COM	2K	16			
PIP .COM	8K	58			
QD31BIOS.ASM	22K	169			
RX .COM	2K	7			
STAT .COM	6K	42			
SUBMIT .COM	2K	11			
SYSGEN .COM	2K	10			
TIME .COM	2K	1			
TX .COM	2K	13			
TX/RX .DES	6K	44			
XSUB .COM	2K	7			

User 1:

Name	Bytes	Recs	Attr	update	create
64KTEST .COM	2K	4			
CMS .COM	2K	12			
CONFIGUR.COM	10K	72			
FORMAT35 .COM	6K	47			
FORMAT40 .COM	6K	47			
PIP .COM	8K	58			
SCROLL .COM	2K	3			
STAT .COM	4K	24			
SYSGEN .COM	2K	10			
XDIR .COM	4K	24			

User 2:

Name	Bytes	Recs	Attr	update	create
32BS5/5 .ASM	22K	162			
32CPM5/1.COM	12K	96			
32CPM5/5.COM	12K	96			
32KTEST .COM	2K	4			
64CPM5/1.COM	12K	96			
64CPM5/5.COM	12K	96			
64KTEST .COM	2K	4			
ASM22 .COM	8K	64			
CONFIGUR.COM	4K	28			
DDT22 .COM	6K	38			
DUMP22 .COM	2K	5			
ED22 .COM	8K	53			
FORMAT .COM	2K	10			
HEXDUMP .COM	2K	12			
LOAD22 .COM	2K	16			
PIP22 .COM	8K	58			
STAT22 .COM	6K	42			
SUBMIT22.COM	2K	11			
SYSGEN22.COM	2K	10			
SYSTEM .DES	6K	45			
TEST .	6K	37			
XSUB22 .COM	2K	7			

53 Files occupying 296K, 52K Free.

\$ cpmfs -f sup3 -D CSYSGENR.RAW

Name	Bytes	Recs	Attr	update	create
64KTEST .COM	2K	4			
ASM .COM	8K	64			
CONFIGUR.COM	10K	73			
CS30CPM .COM	12K	96			
DATE .COM	2K	5			
DDT .COM	6K	38			
DUMP .COM	2K	5			
ED .COM	8K	53			
F .COM	2K	8			
HEXDUMP .COM	2K	12			

```

LOAD      .COM      2K      16
PIP       .COM      8K      58
QD31BIOS.ASM    22K     169
RX        .COM      2K       7
STAT      .COM      6K      42
SUBMIT    .COM      2K      11
SYSGEN    .COM      2K      10
TIME      .COM      2K       1
TX        .COM      2K      13
TX/RX     .DES      6K      44
XDIR      .COM      4K      24
XSUB      .COM      2K       7
  22 Files occupying   114K,      234K Free.

```

```
$ cpmfs -f sup3 -D CUTILSR.RAW
```

Name	Bytes	Recs	Attr	update	create
.DTA	0K	0	R	23-Dec-1118555	09:46
.HEX	0K	0			
.PRN	0K	0			
64KTEST.COM	2K	4			
ASM	8K	64			
BOOT.DTA	2K	1			
CONFIGUR.COM	10K	72			
DATE	2K	5			
DDT	6K	38			
DU-V86	8K	59			
DU-V86.DOC	8K	63			
DUMP	2K	5			
ED	8K	53			
FILT	4K	23			
FORMT40	6K	47			
HEXDUMP	2K	12			
KERMIT	14K	110			
LOAD	2K	16			
MEX	26K	202			
MEX.HLP	50K	390			
MLOAD	4K	22			
MOVIT	16K	124			
NSWEEP	12K	86			
NULU	16K	120			
PIP	8K	58			
QD31BIOS.AQM	12K	93			
RX	2K	7			
SAVEINI.MEX	2K	2			
SCROLL	2K	3			
SQ	6K	47			
STAT	6K	42			
SUBMIT	2K	11			
TEST	2K	2			
TIME	2K	1			
TX	2K	13			
TX/RX	6K	44			
USQ	2K	8			
WATERLOO.PHN	2K	4			
XDIR	4K	24			

```

XSUB      .CUM      2K      /
      40 Files occupying      272K,      76K Free.

```

After extracting or inserting files, you will have to COMPLEMENT the data again to be ready for the actual Computer.

If you wish to create another *.IMD file, BIN2IMD can process the .RAW (Sector Dump) file, and create an *.IMD file that IMD can write to a Floppy. (DSKTRANS and DSKCONV are libdisk utilities that can also write directly to floppy.)

Larry

Re: [HOWTO] - Access CP/M Floppy's via libdisk & cpmtools

by **Ikraemer**

Posted: **2023-07-03 12:27**

Post by Ikraemer » 2023-07-03 07:11

Re: [HOWTO] - Access CP/M Floppy's via libdisk & cpmtools

UPDATE 07-03-2023 – How to translate an IMD file from 300KHz to 250KHz.

I was sent the following IMD file: MBAS.IMD which was an image of a 360K DS/DD Floppy that was saved on a 1.2M Floppy Drive which was running at 360 RPM.

The IMDA utility shows:

```

$ IMDA MBAS.IMD > MBAS.TXT
IMageDisk Analyzer 1.18 / Mar 12 2012
IMD 1.18: 28/03/2019 17:48:01

```

MBASIC

```

Required cylinders: 40
Required heads      : 2
Data rate           : 300kbps
Est. maximum track: 6475 bytes

```

Possible drives/options to write MBAS.IMD :

```

5.25" QD 80-track
Double-step: ON
300 kbps -> 250 kbps

```

```

5.25" HD 80-track
Double-step: ON

```

```

3.5" DD 80-track  NOTE: *1
Double-step: OFF
300 kbps -> 250 kbps

```

```

3.5" HD 80-track  NOTE: *1
Double-step: OFF
300 kbps -> 250 kbps

```

*1 40 track image will use only first 1/2 of 80-track drive.

I needed to change the IMD file to a standard 360K Floppy so I could access the *.BAS files with cpmtools.

The Utility commands were:

```
$ IMDU MBAS.IMD MBAS250.IMD T3=250
```

```
$ IMDU MBAS250.IMD MBAS250.RAW /B /E /D >> MBAS250.TXT
```

Looking at the MBAS250.TXT file I see the file was translated to 250KHz.

IMaGeDisk Utility 1.18 / Mar 07 2012

IMD 1.18: 28/03/2019 17:48:01

MBASIC

Assuming 1:1 for Binary output

```
0/0 250 kbps SD 16x128 = 2048 = 0x0800 = Track 0 & Track 1 = 4096 = 0
    1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
    D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D
HD:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0/1  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D
1/0 250 kbps DD 30x128 = 3840 = 0x0F00 = all remaining tracks = 29952
    14 27 10 23 6 19 2 15 28 11 24 7 20 3 16 29 12 25 8 21 4 17
    D  DFFD DFFD D  D  D  DFFD DFFD D  D  D  DFFD DFFD DFFD D
HD:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
1/1 DE5DE5DE5DE5DE5DE5D DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5
2/0 D  D  D  D00D  D  DE5D  D  D  D00D  D00D  D  D  D  D  D  D00D  D
HD:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
2/1 D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D
3/0 D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D
HD:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
3/1 D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D  D
...
...
...
36/0 DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE
HD:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
36/1 DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE
37/0 DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE
HD:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
37/1 DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE
38/0 DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE
HD:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
38/1 DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE
39/0 DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE
HD:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
39/1 DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE5DE
80 tracks(40/40), 2372 sectors (1992 Compressed)
Floppy Definitions were:
# 22DISK
BEGIN OKT3 Oktel CP/M - DSDD 5" 128 x 30
DENSITY MFM, LOW
CYLINDERS 40 SIDES 2 SECTORS 30,128
SIDE1 0 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,
SIDE2 0 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,
ORDER CYLINDERS
BSH 4 BLM 15 EXM 1 DSM 143 DRM 63 AL0 0C0H AL1 0 OFS 1
FNN
```

```

# cpmtools
# OKT3 Oktel CP/M - DSDD 5" 128 x 30
diskdef okt3
    seclen 128
    tracks 80
    sectrk 30
    blocksize 2048
    maxdir 64
    skew 1
    offset 7936
    boottrk 0
    os 2.2
end

# libdsk
[okt3]
description = OKT3 Oktel CP/M - DSDD 5" 128 x 30
sides = outback
cylinders = 80
heads = 2
seclen = 128
sectors = 30
seclen = 1
datarate = DD

# Flashfloppy/GOTEK
[okt3]
cyls = 80
heads = 2
secs = 30
interleave = 1
bps = 128
id = 1
#rpm = 360
rpm = 300
rate = 250
mode = mfm
iam = no
cpmtools was used to get a directory listing of the files in the .RAW (sector dump)
image.
$ cpmfs -f okt3 -D MBAS250.RAW
$ cpmfs -f okt3 -T rawob,okt3 -D MBAS250.RAW

```

The Directory listing was sane.

Name	Bytes	Recs	Attr	update	create
CLEAR .BAS	2K	1			
MBASIC .COM	24K	190			
PIP .COM	8K	58			
UTIL1 .BAS	4K	29			
UTIL2 .BAS	4K	23			
UTILITY .BAS	4K	29			
6 Files occupying 46K, 252K Free.					
and UTIL1.BAS extracted sane.					

```
$ cpmcp -T OKT3 -I raw00,OKT3 MBASZ50.RAW 0:*.BAS .
UTIL1.BAS:
90 DIM A$(4),B$(4),L1(3),L2(3),M(4),R1(3),R2(3),R3(3),W(50),Z(4)
100 GOTO 340
110 A1=0
120 PRINT"INPUT ADJUSTMENTS TO BILL(+ OR - , '0' WHEN DONE)"
130 INPUT E
140 A1=A1+E
150 IF E=0 THEN 170
160 GOTO 130
170 GOSUB 250
180 GOSUB 290
190 RETURN
200 PRINT CHR$(27)CHR$(91)CHR$(72)CHR$(27)CHR$(91)CHR$(74)
210 RETURN
220 PRINT " ";A$(X);" BILL (CONT')"
230 PRINT
240 RETURN
250 FOR I=1 TO 3
260 PRINT
270 NEXT I
280 RETURN

290 PRINT "INPUT C TO CONTINUE:;
300 INPUT C$
310 GOSUB 200
320 RETURN
340 FOR I=1 TO 4
350 READ N(I)
360 NEXT I
370 FOR I=1 TO 3
380 READ A$(I),B$(I),M(I),Z(I),L1(I),L2(I),R1(I),R2(I),R3(I)
390 NEXT I
400 READ A$(4), M(4), Z(4)
410 GOSUB 200
420 PRINT " UTILITY BILL AUDIT"
430 GOSUB 250
440 PRINT " 1. ELECTRIC BILL"
450 PRINT " 2. GAS BILL"
460 PRINT " 3. WATER BILL"
470 PRINT " 4. PHONE BILL"
480 PRINT " 5. ALL OF THE ABOVE"
490 PRINT " 6. EXIT"
500 PRINT
510 PRINT
520 PRINT" CHOOSE AN OPTION ";
530 INPUT P
540 IF (P<1)+(P>6) THEN 530
550 ON P GOTO 1020,1050,1080,1110,1590,1690
560 GOSUB 200
570 PRINT" ";A$(X);" BILL"
580 PRINT
590 PRINT" PREVIOUS METER READING"
600 INPUT E1
610 PRINT
620 PRINT" PRESENT METER READING"
630 INPUT E2
640 PRINT
```

```

650 PRINT"INPUT DAYS IN THE BILLING PERIOD"
660 INPUT D
670 GOSUB 250
680 GOSUB 290
690 GOSUB 200
700 GOSUB 220
710 GOSUB 110
720 U=E2-E1
730 Y=U/D
740 IF U>L2(X) THEN 800
750 IF U>L1(X) THEN 780
760 T=M(X)+R1(X)*U
770 GOTO 810
780 T=M(X)+R1(X)*L1(X)+R2(X)*(U-L1(X))
790 GOTO 810
800 T=M(X)+R1(X)*L1(X)+R2(X)*(L2(X)-L1(X))+R3(X)*(U-L2(X))
810 T1=T*Z(X)
820 T2=T+T1
830 T3=T2+A1
840 GOSUB 200
850 GOSUB 220
860 PRINT "USE FOR THE PERIOD IS ";INT(U*100)/100;" ";B$(X)
870 PRINT
880 PRINT "USE/DAY IS ";INT(Y*100)/100;" ";B$(X);" OR $";
890 PRINT INT(T2/D*100)/100;"/DAY INCLUDING TAX"
900 PRINT
910 PRINT A$(X);" BILL:"
920 PRINT " W/OUT TAX      :$";INT(T*100)/100
930 PRINT " TAX IS        :$";INT(T1*100)/100
940 PRINT" ADJ'TS         :$";A1
950 PRINT
960 PRINT " *TOTAL*       :$";INT(T3*100)/100
970 IF N(X)=1 THEN 990
980 PRINT"SPLIT ";N(X); " WAYS:$";INT(T3/N(X)*100)/100
990 PRINT
1000 GOSUB 290
1010 RETURN
1020 X=1
1030 GOSUB 560
1040 GOTO 430
1050 X=2
1060 GOSUB 560
1070 GOTO 430
1080 X=3
1090 GOSUB 560
1100 GOTO 430
1110 GOSUB 200
1120 X=4
1130 PRINT "          ";A$(X);" BILL"
1140 PRINT
1150 GOSUB 110
1160 FOR K=1 TO N(X)
1170 I=1
1180 IF N(X)=1 THEN 1200
1190 PRINT"FOR PERSON #";K;" ";
1200 PRINT "INPUT CHARGE FOR EACH LONG DISTANCE CALL (INPUT '0' WHEN DO
1210 TNPIIT W(T)

```

```

1220 IF W(I)=0 THEN 1250
1230 I=I+1
1240 GOTO 1210
1250 GOSUB 200
1260 PRINT " PERSON #";K;" ,"
1270 FOR J=1 TO I-1
1280 PRINT "CALL #";J;" :$";W(J)
1290 PRINT
1300 PRINT "IS THIS CORRECT (Y/N)"
1310 INPUT C$
1320 IF C$="Y" THEN 1350
1330 PRINT "TYPE IN CORRECTION"
1340 INPUT W(J)
1350 PRINT
1360 NEXT J
1370 GOSUB 200
1380 GOSUB 220
1390 T=0
1400 FOR J=1 TO I-1
1410 T=T+W(J)
1420 NEXT J
1430 PRINT "SERVICE:  $";INT(M(X)/N(X)*100)/100
1440 PRINT
1450 PRINT "LD CALLS:  $";T
1460 PRINT
1470 PRINT "ADJ'TS :  $"INT(A1/N(X)*100)/100
1480 PRINT
1490 T1=T+INT(M(X)/N(X)*100)/100
1500 T2=INT(T1*Z(X)*100)/100
1510 PRINT "TOTAL TAX:  $";T2
1520 PRINT
1530 PRINT
1540 PRINT " TOTAL BILL:  $";T1+T2+INT(A1/N(X)*100)/100
1550 GOSUB 250
1560 GOSUB 290
1570 NEXT K
1580 GOTO 430
1590 FOR F=1 TO 3
1600 X=F
1610 GOSUB 560
1620 NEXT F
1630 GOTO 1110
1640 DATA 1,1,1,1
1650 DATA ELECTRIC ,KWH,3.50,0.05,350,1300,.06490,.06490,.06490
1660 DATA GAS,CCF,4.05,0,99999,99999,.49541,0,0
1670 DATA WATER,CCF,3.26,0,500,99999,.144,.160,0
1680 DATA PHONE,13.50,.03
1690 END

```

Larry