

ImageDisk
A
Diskette Imaging
program
for
Soft-Sector
Formats

Version 1.18
Revised 8-Mar-2012

Copyright 2005-2012 Dave Dunfield
All rights reserved.

ImageDisk

TABLE OF CONTENTS

	Page
1. INTRODUCTION	1
1.1 License	2
1.2 System Requirements	3
1.3 Command Line Options	4
2. MAIN MENU COMMANDS	6
2.1 (S)ettings	6
2.2 (R)ead	10
2.3 (W)rite	10
2.4 (F)ormat	11
2.5 (E)rase	11
2.6 (A)lign/test	11
2.7 (C)lean	12
2.8 (T)est RPM	12
2.9 Track e(X)clusion map	13
3. FLOPPY DRIVE TYPES	14
3.1 5.25" Double Density	14
3.2 5.25" Quad Density	14
3.3 5.25" High Density	15
3.4 3.5" Double Density	15
3.5 3.5" High Density	16
3.6 8" Floppy Drive	16
3.7 writing 40 track disks on 80 track drives.	17
3.8 Single-density and High-density drives	17
3.9 Easy/Cheap 8" drive adapter	18
4. TIPS AND TRICKS	21
4.1 PC floppy controllers are DIFFERENT!	21
4.2 Slowing the drive	21
4.3 Identifying an image	22
4.4 writing a 250kbps image on a HD drive	22
4.5 writing a 300kbps image on a DD drive	22
4.6 Reading DD/QD images on HD drives	22
5. MISC. Topics	23
5.1 PC Diskette Drive Types	23
5.2 DD Transfer Rate (250kbps <> 300kbps)	23
5.3 Double Stepping	23
5.4 Gap Lengths	24
5.5 Differences from Teledisk	24
5.6 256 Tracks / 255 Sectors?	24

	Page
5.7 Aligning a drive	25
6. IMAGE FILE FORMAT	26
6.1 Mode value	26
6.2 Sector size	26
6.3 Head value	27
6.4 Sector numbering map	27
6.5 Sector Cylinder Map	27
6.6 Sector Head map	27
6.7 Sector Data Records	28
7. IMDA: ImageDisk Analyzer	29
7.1 Introduction	29
7.2 Command line options	29
8. IMDU: ImageDisk Utility	29
8.1 Introduction	29
8.2 Command line options	30
9. IMDV: ImageDisk Viewer	31
9.1 Introduction	31
9.2 Command line options	31
9.3 Custom Character Sets	31
10. TESTFDC - Evaluate PC floppy controller	32
11. Converting from other image file formats	33
11.1 BIN2IMD: Binary to ImageDisk .IMD utility	33
11.2 TD02IMD: convert Teledisk .TD0 to .IMD files	36
11.3 DMK2IMD: convert .DMK images to ImageDisk .IMD	38
11.4 ANY2IMD : Convert ANY format to .IMD by floppy write/read	40

1. INTRODUCTION

ImageDisk is a program which reads entire diskette images into files and can later recreate exact copies of the diskettes in order to:

- Archive diskette images for long term storage.
- Make it possible to transfer diskettes via the internet.

Unlike DISKCOPY, ImageDisk is not limited to PC format disks and PC type diskette drives. It has been designed to handle any drive type and disk format possible with the PCs disk controller hardware.

ImageDisk was created to address the problem of preserving system and application diskettes for classic computers. Floppy disks are rarely used with modern PCs, however they are the "bread and butter" of classic systems. Without the system/application diskettes these important artifacts of our digital history will become silent.

Please see the file HELP.TXT for help with common problems.

ImageDisk was created by:

Dunfield Development Services
115 Manion Heights Cres.
Carp, Ontario Canada
K0A 1L0

<http://www.dunfield.com>

<= Visit my collection
of classic computers.

1.1 License

This ImageDisk package is Copyright 2005-2012 Dave Dunfield, who grants the following free license for non-commercial use. You may:

- Use the package for personal/hobby purposes for as long as you wish, on as many computers as you wish.
- Redistribute copies of the package provided none of the copyright material has been altered or omitted from the distributed version. Additions to the package must be clearly marked as such.
- Create/manipulate .IMD files with no restriction. The disk image file format specification is hereby placed into the public domain.

All rights and permissions not listed above are explicitly reserved by the author.

Use of this software is entirely at your own risk. The author makes no guarantee of the suitability of this software for any particular purpose.

Copyright law prohibits duplication of material so protected. It is your responsibility to determine the legality of any particular use of this software. This package is NOT intended as a means to bypass software copy protection.

By using this software, you accept all responsibility of any kind which may result from such use. You agree to indemnify and hold harmless the author and his agents from any action which may arise from your use of the software.

1.2 System Requirements

ImageDisk requires a DOS based PC with an AT type floppy controller. Although ImageDisk will run on a 4.77Mhz PC/XT type machine, some of the analysis functions are real-time intensive and best results will be obtained with a 486 class machine or better.

ImageDisk *MAY* run under versions of windows which provide direct access to hardware (win9x), however a pure DOS environment is highly recommended. TSRs or other drivers which interfere with the real-time operation of the system may cause problems with ImageDisk.

ImageDisk will NOT run under operating systems which prohibit access to the floppy disk controller hardware, such as: winNT/2000/xp.

1.2.1 Floppy Diskette Controller

ImageDisk assumes that you have an AT compatible floppy disk system. Ie: one on which the data rate can be set. If you use ImageDisk with a PC/XT type diskette controller, you MUST use the command line option LR=250 to lock the data rate at 250kbps. If you do not, then Image files will have incorrect transfer rate values encoded within them.

Without this option, ImageDisk might "try" a data rate of 500kbps or 300kbps and find that it works - because it is actually is fixed at 250kbps in hardware, and happily write an image file indicating the wrong value.

NOTE: A PC/XT controller will NOT be able to read HD or 8" disks. It is NOT recommended to use ImageDisk with a PC/XT type floppy disk system.

Also note: Many floppy controllers are limited designs which do not work well with non-PC formats. All PCs are NOT THE SAME!

1.3 Command Line Options

Use: IMD [[$\$$]CMDfile] [options]

/4	- Assume 4 drives/FDC	[2] *
/A	- select drive A	[A]
/B	- select drive B	[A]
/C	- select drive C	[A] *
/D	- select drive D	[A] *
/F	- perform Full analysis	[off]
/H	- inhibit initial Help screen	[enabled]
/K	- Keep bad sectors	[off]
/M	- force Monochrome video	[color]
/N	- force Non-deleted data	[off]
/S[addr]	- use Secondary FDC at [372]	[primary/3F2]
C=1-255	- set number of Cylinders	[80]
CP=[1-9]	- number of Cleaning Passes	[3]
D=path	- default read/write Directory	[current]
DS=0-1	- Double-Step (0=OFF, 1=ON)	[Original]
FF=00-FF	- Format Fill value	[E5]
GF=0-255	- Format Gap	[Calculated]
GW=0-255	- Write Gap	[Calculated]
IL=[1-99]	- InterLeave (blank=BestGuess)	[As read]
LR=250/300/500	- Lock data Rate	[off]
R=0-255	- select Retries	[5]
S=1-2	- set number of Sides	[Original]
SD=0-2000	- Delay after Step (ms)	[55]
SR=1-16	- Step Rate (ms)	[8]
HL=2-254	- Head Load time (ms)	[254]
HU=16-240	- Head Unload time (ms)	[240]
T5=250/300/500	- 500khz Translate	[500]
T3=250/300/500	- 300khz Translate	[300]
T2=250/300/500	- 250khz Translate	[250]
TG=M[,c]	- enable Manual TG43	[off]
TG=P1-3[,c][,wf,wr]]	- enable Parallel-port TG43	[off]
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]

'*' options require FDC supporting four physical drives (most don't)

Details of the meaning of most command line options can be found in "MAIN MENU COMMANDS / Settings" in the description of the corresponding Settings menu options.

1.3.1 Command Files

You can place commonly used command line options into a file with a .CMD extension, and activate all of the options in that file by specifying the file name on the command line. Options must occur one per line. Blank lines, and lines beginning with ';' (comment) are ignored when the .CMD file is processed.

```
eg: ; Sample command file for DS in HD drive
    ; 40 Cylinders, double-step, translate 250<>300 kbps
    C=40
    DS=1
    T3=250
    T2=300
```

Normally, IMD will look for a command file in the current directory, which allows you to easily keep command files specific to your current task.

You can use a '\$' prefix to cause IMD to look for the command file in the IMD home directory. This accomodates general command files that you wish to access from "anywhere".

When IMD starts, it looks for a command file names IMD.CMD in it's home directory and if found, it will process it before any other command files or command line options are performed. This provides a way to customize the default settings of IMD to your preferences.

2. MAIN MENU COMMANDS

The following commands are available from the ImageDisk main menu, by pressing the bracketed letter in the command name:

2.1 (S)ettings

The Settings menu allows you to change various ImageDisk operational settings. Use the Up/Down arrow keys to select one of the following settings, and press SPACE to change it's value.

2.1.1 Drive

Controls which physical floppy diskette (A or B) will be accessed by ImageDisk.

2.1.2 cylinders

Sets the maximum number of cylinders which ImageDisk will try to access.

In general: CYLINDERS should be set to the maximum number of tracks which can be stepped on your drive.

During Reading, ImageDisk will not attempt to analyze or read past this setting. Note that reading will still end if a track is found with no data before then. (hint: If you need to skip blank tracks, use the exclusion map)

During writing, this setting has no effect. ImageDisk will attempt to recreate all cylinders that are defined within the image.

During Align/Test, this setting has no effect (stepping is controlled manually).

When cleaning the heads, this setting controls the range of the "scrubbing" motion throughout which the heads are moved.

2.1.3 Sides

Controls how ImageDisk treats the the disk as single or double sided during an analysis. Has effect only for Analysis, Reading and manual Formatting of a disk. During writing, ImageDisk will recreate both sides if present in an image.

Available settings are:

AS ORIGINAL : ImageDisk will examine the disk to determine if the second side contains data.

ONE : Useful to prevent ImageDisk from detecting Side1 data for previously used disks with valid but unrelated data on the second side. Also speeds up analysis of single sided disks.

TWO : Use this to force ImageDisk to try and read a second side.

2.1.4 Double-Step

Setting this option ON causes ImageDisk to step two physical tracks for each logical track on the disk. This is necessary when reading 48tpi (40 track) 5.25" disks on 96tpi (80 track) drives. QD/HD drives have twice as many tracks within the same diskette area as DD drives.

Setting the option OFF causes ImageDisk to step a single physical track for each logical track in the image.

The setting AS ORIGINAL means that ImageDisk will examine the disk and determine if double-stepping is required. This is available for READING only - when WRITING a disk, you MUST set this option to either ON or OFF. (The correct setting is determined by the type of drive you are using and the disk image you are writing - in general, if the disk image has 40 tracks in it, and you are using a QD/HD floppy drive, you will need to turn this option ON - you may also need to select a data rate translation).

2.1.5 R/W gap & Format gap

The GAP lengths cannot be read from the disk, and using the wrong length can result in an unreadable copy. Unfortunately, determining the correct gap length can be "magic science".

CALCULATED will use suggested values from the NEC databook if it exists for the disk format being used, otherwise an estimate is made based on the total track data size, sector size and number of tracks/sector. This may not be correct, and may result in a bad disk, or one which cannot be written. I have therefore provided the option to force specific gap length settings.

If you have problems, try increasing/decreasing the calculated value. Note that in general, the R/W gap must be less than the FORMAT gap. FORMAT = total gap between sectors, R/W = gap after READ/WRITE - and should fall somewhere in the middle of the FORMAT gap.

Sometimes when a disk is made with an incorrect inter-sector gap size, it will read fine, however attempts to write the disk will corrupt it. This is because the inter-sector gap is too small, and the partial gap written at the end of a sector overflows into the next sector.

If you find this is happening, and you are unable to determine good gap size values, another solution is to simply copy the entire disk on the target system - this should restore all of the gaps to the correct sizes as determined by your systems disk driver.

2.1.6 Format Fill

This is the byte value used to fill freshly formatted sectors. Normally, this gets replaced when the sector data is written, however any sectors marked as "unavailable" in the image will remain filled with this value when it is written to a disk.

This setting also establishes the sector content for disks which are low-level formatted (only) with the Format command.

2.1.7 Full Analysis

To save time, normally when ImageDisk has analyzed a track, it performs only a "quick check" when moving to the next track - If the transfer rate, density and sector size are the same, it assumes the track will be the same as the last and proceeds to read the sectors. It will perform a full analysis only if the parameters change, or it encounters read errors on the sector data.

This option causes ImageDisk to perform a full analysis on each track - This increases the time needed to read a disk, but may be required for disks with missing or oddly-numbered sectors on some tracks, or sector head/cylinder values which do not match the physical track position on the disk.

2.1.8 Interleave

This controls how ImageDisk determines the Interleave factor of the disk being read or written. ImageDisk tries to discover Interleave factor by reading the sectors ids passing under the head after an index pulse, however for disks which are difficult to read, this is not reliable. The following options are available:

AS READ : Sectors are read/written in exactly the order that was encountered on the original disk/image file.

BEST GUESS : Examines sector order and determines the interleave factor which most closely matches. Writes sectors from lowest with this calculated interleave.

1-99 : writes sectors from lowest with specified interleave

2.1.8.1 Interleave factor (skew) explained

Many systems offset the physical sectors written to a track, so that extra sectors occur between sequentially numbered sectors, providing extra time for the CPU to process data between sequential reads. For example, consider a 9-sector disk:

1:1 Interleave:	1	2	3	4	5	6	7	8	9
2:1 Interleave:	1	6	2	7	3	8	4	9	5

The 1:1 interleave disk will require only one disk rotation to read all sectors, but only if the CPU is fast enough to be ready for a new sector IMMEDIATELY after reading the previous one. If the CPU is not that fast, it will take 9 revolutions. The 2:1 interleave disk takes two revolutions to read all sectors, but it allows an extra "sectors worth" of time between reads.

2.1.9 Retries

This controls the number of times that a disk operation will be retried.

After one set of retries, ImageDisk will normally re-analyze the disk to determine if the track format has changed. Retries=0 is a special case which prevents this from happening. This is useful if you want to quickly scan the disk to extract readable sectors only.

2.1.10 Keep bad sectors

This setting causes IMD to try and recover data from sectors which were read with data errors (bad CRC). When ON, IMD will record data from bad sectors along with a flag indicating that the sector is bad. When OFF, IMD will discard the sector and report it as unavailable.

NOTE: IMD will warn you if you recreate a disk containing bad sectors, however it is not reasonably possible to make the newly written sector appear bad on the disk. If you later read the disk the data will still be in error, however it will read as a good sector!

2.1.11 Data rate translations

Data rate translation allows you to try to recreate a disk on a different drive type from that on which it was originally read.

The most common/useful translations are:

250 -> 300	= Read on 5.25 DD, write on 5.25 HD **
300 -> 250	= Read on 5.25 HD **, write on 5.25 DD

** Normally you would also enable Double-Step on the HD drive.

You can see the transfer rates in an image with the IMDU utility.

2.2 (R)ead

Prompts for an image filename, and a comment record, then reads the disk as defined by current settings into the image file.

2.3 (W)rite

Prompts for an image filename, then writes the disk as defined by current settings from the image file.

2.4 (F)ormat

Performs a low-level format of a diskette.

Drive, Sides, Step, Format-fill and Interleave must be established via Settings.

Number of sectors/track, starting sector number, data rate, encoding (FM or MFM) and sector size will be prompted for.

NOTE: This is a LOW LEVEL format only. ImageDisk does not know how to establish a file system (high level format) for any operating system. If you want to automatically create high-level formatted disks for a certain system the easiest thing to do is to Read a freshly formatted disk and create copies from that image.

2.5 (E)rase

Completely erases the disk to make it unreadable.

Drive, Sides and Step must be established via Settings.

The PC controller has no documented capability to erase a disk. This works by formatting the disk with a extra-long sector at a low data- rate, which overflows the track, wraps and overwrites the ID field, causing the track to contain no valid sectors.

This may not work on some 8" drives and/or some floppy controllers. If this command does not work, you may be able to achieve a similar result by experimenting with F)formatting large sectors.

2.6 (A)lign/test

Performs diagnostics by continuously reading sector IDs passing under the drive head. For each sector ID, outputs:

```
<cylinder> <head> <sector> <#match> <#fail>
```

#match and #fail are updated after each full revolution and show the number of ID's on the track which MATCH the physical cylinder, and the number of ID's which DO NOT MATCH the physical cylinder.

At each revolution, a tone also sounds which is higher in pitch when more matching ID's are found. This allows you to adjust drive alignment by ear.

The following indicators are updated in real-time at the top of the screen:

```
F=fault          R = ready          D = double-sided  
W=write-protect Z = track-0
```

2.6.1 Alignment/test Commands

The following command characters can be entered during Align/Test:

- A - re-Analyze track
- B - toggle Beep on/off
- D - read Data from all sectors on track
- F - Format diskette track
- H - toggle Head 0/1
- I - write \$CnHn.IMD with track data (must do 'D' first)
- P - set format/write Parameters
- R - recalibrate to track zero/Reseek current track
- S - toggle single/double Stepping
- W - Write data to all sectors on track
- Z - recalibrate to track Zero (and stay there)
- 0-9 - seek to track 'n'*10 (00 to 90)
- +/- - seek +/- one track from current position
- X - exit disk alignment/test function.

2.7 (C)lean

Although I do not generally recommend the use of "Cleaning Disks", sometimes you don't have time to remove and disassemble the drive and these are a required immediate solution to dirty heads.

When prompted, insert a "wet" cleaning disk and press ENTER.

This function will "scrub" the disk head back and forth in a zigzag pattern.

NOTE: Make sure that CYLINDERS under Settings is set to match the number of physical cylinders on your drive.

2.8 (T)est RPM

This function calculates the revolutions per minute of the diskette drive. Note that this is a simple function which employs only the standard DOS 55ms timer tick - The results will become more accurate the longer you allow the test to run.

Note that the result is cumulative, and will not accurately reflect changes to the drive speed. If you are adjusting the drive, press SPACE to restart the calculation after each adjustment.

2.9 Track e(X)clusion map

Sometimes it is desirable to exclude certain tracks when reading an image (Unreadable, Unformatted, incompatible with your controller).

This function allows you to select tracks to exclude from the image when reading/writing. Note that tracks not occurring in an image will NOT be formatted or written when the disk is recreated.

Use the arrow keys to position the cursor on the track you wish to exclude, and then press SPACE to toggle between:

- = Exclude side 0 only
- = Exclude side 1 only
- = Exclude both sides (entire track)

3. FLOPPY DRIVE TYPES

There are many drive types that can be connected to a PC. Aside from physical media, the characteristics which affect how a disk is read or written are: Number of Cylinders, Rotation speed and Transfer rate.

Drive type	Cyls	Rpm	Rate
5.25" DD	40	300	250
5.25" QD	80	300	250
5.25" HD	80	360	500/300
3.5" DD	80	300	250
3.5" HD	80	300	500/250
8"	77	360	500

Understanding these parameters and how they relate to the different drive types is very important to using ImageDisk effectively.

3.1 5.25" Double Density

This is the standard 5.25" drive on most classic systems. It has 40 tracks, rotates at 300 RPM, and transfers at a rate of 250 kbps.

This gives it a normal formatted capacity of about 4600 bytes/track, which gives a total disk capacity of: $4600 \times 40(\text{Tracks}) \times 2(\text{heads}) = \text{about } 360\text{k bytes}$.

3.2 5.25" Quad Density

This drive is very similar to the 5.25" DD drive, the only difference being that it has 80 cylinders instead of 40. It rotates at 300 RPM, and has a data transfer rate of 250 kbps. This makes it electrically identical to the 3.5" DD drive.

This gives it a normal formatted capacity of about 4600 bytes/track, which gives a total disk capacity of: $4600 \times 80(\text{Tracks}) \times 2(\text{heads}) = \text{about } 720\text{k bytes}$.

The 5.25" QD drive is quite uncommon on PCs, however it does occur on some classic systems.

This drive can read 5.25" DD format disks by enabling Double-Step. Please read the note regarding track width when writing.

3.3 5.25" High Density

This drive has 80 cylinders, rotates at 360 RPM, and transfers at a rate of 500 or 300 kbps.

At 500kbps it has a formatted capacity of about 7600 bytes/track, which gives a total disk capacity of $7600 \times 80(\text{Tracks}) \times 2(\text{Heads}) = \text{about } 1.2\text{M bytes}$.

The 300kbps data rate is for compatibility with DD/QD drives. Due to the higher rotational speed (360 RPM), 300kbps results in the same bit density as 250kbps on the slower (300 RPM) drives.

At 300kbps it has a formatted capacity of about 4600 bytes/track, which gives a total disk capacity of $4600 \times 80(\text{Tracks}) \times 2(\text{Heads}) = \text{about } 720\text{k bytes}$.

This drive can read and write 5.25" QD format disks, however it will do so at a transfer rate of 300kbps. To create a true 5.25" QD image you must translate 300kbps to 250kbps when reading the disk, and 250kbps to 300kbps when writing the disk.

This drive can read 5.25" DD disks by enabling Double-Step, however it will do so at a transfer rate of 300kbps. To create a true 5.25" DD image you must translate 300kbps to 250kbps when reading the disk, and 250kbps to 300kbps when writing the disk.

Please read the notes regarding track width when writing, and single density operation with 5.25" HD drives.

3.4 3.5" Double Density

This drive has 80 tracks, rotates at 300 RPM, and has a data transfer rate of 250kbps, which makes it electrically identical to the 5.25" QD drive.

This gives it a normal formatted capacity of about 4600 bytes/track, which gives a total disk capacity of: $4600 \times 80(\text{Tracks}) \times 2(\text{heads}) = \text{about } 720\text{k bytes}$.

The 3.5" DD drive is quite uncommon on PCs, however it does occur on some classic systems and a few early PS/2 models.

3.5 3.5" High Density

This drive has 80 tracks, rotates at 300 RPM, and has a data transfer rate of 500kbps or 250kbps.

At 500kbps it has a formatted capacity of about 9200 bytes/track, which gives a total disk capacity of $9200 \times 80(\text{Tracks}) \times 2(\text{Heads})$ = about 1.5M bytes.

The 250kbps data rate is for compatibility with DD drives. At this data rate, the drive functions exactly as a 3.5" DD drive.

At 250kbps it has a formatted capacity of about 4600 bytes/track, which gives a total disk capacity of $4600 \times 80(\text{Tracks}) \times 2(\text{Heads})$ = about 720k bytes.

3.6 8" Floppy Drive

The 8" drive is rarely found on a PC however it occurs on many classic systems. It has a different connector than PC drives, and you will need to make an adapter to connect it.

This drive has 77 tracks, rotates at 360 RPM, and has a data transfer rate of 500kbps. Aside from being three cylinders short, it is otherwise very similar to the 5.25" HD drive.

Unlike the other disk types, 8" diskettes are physically either single or double sided - the difference being the position of the index hole.

A single-sided disk has a normal formatted capacity of about 7600 bytes per track which gives a total disk capacity of approximately $7600 \times 77(\text{Tracks})$ = about 600k bytes.

Double sided disks have twice this or about 1.2M bytes.

Some 8" drives may have a TG43 signal issue when used on a PC floppy controller.

3.7 writing 40 track disks on 80 track drives.

An 80 track drive has a narrower track width than a 40 track drive, which means that the track laid down by the head is thinner than that laid down by the head in a 40 track drive.

When writing a disk in an 80 track drive which has previously been written on with a 40 track drive, some portion of the original 40 track data will remain at the edges of the new thinner track.

If you then attempt to read the disk in a 40 track drive, the head will see two conflicting signals which may cause read errors.

It is best to use blank or bulk-erased disks when writing a 40-track image to an 80-track drive.

3.8 Single-density and High-density drives

I have encountered a some PCs which are able to perform single density operations at 250 kbps (DD drive), but NOT at 300 kbps (HD drive). If you are using an HD drive and are unable to read or write single-density disks, try a DD drive.

I have a note on my web site describing a modification which allows an HD drive to operate at 300 rpm and 250 kbps data rate.

3.9 Easy/Cheap 8" drive adapter

Begin with an ISA bus connector from an old PC - this is the correct spacing for the 50 pin 8" drive, but too long. Cut down to 50 pins.

Salvage a 34 pin edge connector from a dead 5.25" drive - look for one which has a "ground plane" connecting the odd numbered pins. Leave about 1/2 inch of board under the connector, remove components and carefully cut tracks to insure that the even number pins are isolated from the remainder of the board.

Scrape away the solder mask to bare the ground plane. Position the "board" between the rows of pins on the 50 pin connector, centered from each end with the ground plane against the odd numbered pins of the 50 pin connector. Solder the odd pins to the ground plane and add wire extensions to insure that all 25 pins are joined together.

Using 30 gauge "wire-wrap" wire, and being careful to only solder to the very inner end of the "fingers" on the 34 pin edge connector, connect the even numbered pins according to this pinout:

PC(34p)	8"(50p)	Description	
	2	TG43	(see Note)
8	20	Index	
12	26	DS1(PC) -> DS0(8")	
16	18	Motor ON/Head Load	All odd
18	34	Direction (Note)	numbered
20	36	Step	pins are
22	38	Write Data	ground.
24	40	Write Gate	
26	42	Track 0 detect	
28	44	Write Protect	
30	46	Read Data	
32	14	Side1 Select	
34	12	Ready	

You will now have an adapter which can be plugged onto an 8" drive and allows you to connect to a standard 34 pin 5.25" PC cable.

NOTE: This pinout connects DS1 on the PC (the standard for both drive A or B with a twisted cable) to DS0 on the 8" drive. You can leave the 8" drive jumpered as Drive-0, and connect it as either A or B at the appropriate position on the PC cable.

3.9.1 Track > 43

Some 8" drives require a signal called "TG43" which indicates that the drive is positioned to a cylinder greater than 43. This is used to reduce the write current on inner cylinders.

The diskette controller in the IBM PC does not provide this signal, which causes write problems on the inner tracks of some 8" Drives.

There are three possible solutions:

3.9.1.1 Operating without TG43 signal

Some drives do not require TG43. Examine the control board on your drive. If Pin 2 of the 50 pin interface connector is not connected, then your drive does not require TG43. If it is connected, the drive MAY require TG43 - it may also be a jumper selectable option, please consult your drive documentation.

Since TG43 is used only during WRITE operation, you will be able to read 8" disks. Writing may fail or be unreliable on inner tracks.

You can also try tying TG43 permanently LOW or HIGH ... One or the other may allow you to create disks which although not correct, MAY be readable and can be copied on the destination computer to create correctly written disks.

3.9.1.2 TG43 via modified drive cable

The floppy control chip in the PC provides TG43 and DIRECTION on the same signal. When moving the head this signal means DIRECTION. When writing the signal means TG43 (reduce write current).

IF your disk controller does not "remove" TG43 from pin 18 of the interface during write, and IF your drive is compatible with the multiplexed signals, then you MAY be able to provide TG43 to your drive just by connecting both Pins 2 (TG43) and 34 (DIRECTION) of the 50 pin 8" interface to pin 18 (DIRECTION) of the PC interface.

Note that although this works with some drives, it does not meet the floppy interface standard which requires these signals to be driven separately. Some drives will not work in this configuration.

3.9.1.3 TG43 via parallel port

If necessary, the TG43 signal required by some 8" drives can be generated with the parallel port, using this command line option:

```
TG=Pport[,Cylinder[,wf,wr]]
```

port	= Parallel port 1-3 or hex address	[None]
Cylinder	= Cylinder above which to reduce WC	[43]
wf	= Value to write for full WC	[F0]
wr	= Value to write for reduced WC	[0F]

3.9.1.4 Manual TG43 control

If necessary, the TG43 signal required by some 8" drives can be controlled manually (via toggle switch). You can set IMD to pause for manual TG43 control when writing with the command line option:

```
TG=M[,Cylinder]
```

Cylinder	= Cylinder at which to pause write	[43]
----------	------------------------------------	------

4. TIPS AND TRICKS

4.1 PC floppy controllers are DIFFERENT!

The floppy disk controller varies widely from one PC to another, and MOST PCs have limited controllers designed only for the formats used on PC disks. Many controllers cannot handle single-density, or lock correctly with the small gaps used in some non-PC formats.

In developing ImageDisk, I have tested with dozens PCs, and found that only a few of them could read and write all the mixed density formats that I was able to make work. Some could read all the formats (with a high number of recoverable errors) but could not write them all. The remainder could not even read all the formats.

What this means is that you may have to try a number of different types of PCs before you find one that makes a good ImageDisk system.

The ImageDisk package includes a floppy disk controller evaluation program called TESTFDC - This program will check out your PC floppy controller and report on it's capabilities. I maintain a listing of test results for various mainboards and floppy controllers on my classic computer website.

4.2 Slowing the drive

The 765 FDC used in the IBM PC has limitations which are stricter than the WD 17xx/27xx controllers used in many vintage systems. One such limitation is that the 765 cannot handle the smaller gap times used on some disks, and can have trouble reading them.

A useful trick is to slow down the drive to around 290 RPM - this often allows enough extra time that the 765 controller can handle the disk.

Many disk drives have a potentiometer on the motor control circuit board which allows adjustment to the drive speed. You can use the Test RPM function to check your drive speed.

4.3 Identifying an image

If you receive a .IMD image and do not know what it contains, or what type of drive it should be created on, you can use the IMDU utility to learn more about the image:

- Hopefully the COMMENT shown at the top of the listing will tell you what is contained in the image.
- The summary near the end will show the number of tracks of side0 and side1 information. If the disk contains only side0 tracks, then it is single-sided.
- Look at the total number of cylinders for each side, normally:
 - 40 = 5.25" DD
 - 80 = 5.25" QD, 5.25" HD, 3.5" DD or 3.5"HD
 - 77 = 8"
- Look at the transfer rate, normally:
 - 250kbps = 5.25" DD, 5.25" QD or 3.5" DD
 - 300kbps = 5.25" DD, 5.25" QD read on a 5.25" HD
 - 500kbps = 5.25" HD, 3.5" HD or 8"

4.4 Writing a 250kbps image on a HD drive

When an image which was recorded on a 5.25" DD or 5.25" QD drive at 250kbps is written on a 5.25" HD drive, it must be written at 300kbps to account for the higher rotational speed of the HD drive.

You must therefore use Settings to Translate 250kbps to 300kbps.

For images containing 40 tracks (DD), you will also have to enable Double-Stepping in order to write the image correctly.

See also the note on writing 40 track disks on 80 track drives.

4.5 Writing a 300kbps image on a DD drive

An image which was recorded at 300kbps on a 5.25" HD drive, can be written on a 5.25" DD or 5.25" QD drive by using Settings to translate 300kbps to 250kbps.

Note that a QD (80 track) image cannot be written to a DD (40 track) drive.

4.6 Reading DD/QD images on HD drives

A 250kbps DD/QD image can be read on a 5.25" HD drive, however it will read at 300kbps due to the faster rotational speed of the HD drive.

If you plan to give this image to others, it is recommended that you use Settings to translate 300kbps to 250kbps before reading the disk. This will result in an image file showing the true 250kbps nature of the original disk.

5. MISC. Topics

5.1 PC Diskette Drive Types

ImageDisk does not know or care what types of drives are installed in your PC - it determines the correct transfer rate and type of stepping required by analysis of the disk.

When writing a disk on a different drive type than it was originally read on, you may need to set a data transfer rate translation. For example, if recreating a 5.25" DD disk on a 5.25" HD drive, you will need to translate 250kbps to 300kbps.

5.2 DD Transfer Rate (250kbps <> 300kbps)

ImageDisk determines the correct transfer rate for your drive when reading a disk, and that transfer rate will be correct when writing the disk on the same drive, however the transfer rate may NOT be correct if you write the disk on a different drive type than it was originally read on.

Specifically: Standard 5.25" DD drives operate at 250kbps, however HD 5.25" drives on a PC operate at 300kbps when operated in DD mode (this is because the drive rotates at 360rpm instead of 300 rpm - the actual flux density transition rate on the media will be the same).

A DD image read on a DD drive has a data rate of 250kbps. In order to write this to a HD drive, you need to write the data at 300kbps. Use settings to select: 250 kbps -> : 300 kbps

A DD image read on a HD drive has a data rate of 300kbps. In order to write this to a DD drive, you need to write the data at 250kbps. Use settings to select: 300 kbps -> : 250 kbps

When reading DD in a HD drive, I suggest setting: 300kbps -> 250kbps The image will now indicate the normal DD data rate of 250kbps (even though it was read on the HD drive), and will recreate correctly on a DD drive - HD users will have to set: 250 kbps -> | 300 kbps.

5.3 Double Stepping

Although ImageDisk can determine that Double-Stepping is required when reading a disk (48tpi disk in 96tpi drive), it does not know what types of drives you are using, and therefore cannot tell when Double-Stepping is required when writing a disk.

You must therefore set the Double-Stepping option correctly when writing an image back to a disk.

5.4 Gap Lengths

As far as I can tell, determining the correct inter-sector Gap length is mostly "Magic Science" ... The NEC databook provides tables of suggested gap lengths for common sector formats, however no material appears to exist documenting how to calculate the gap lengths for non-standard formats.

Please read the Notes on Gap Length under Settings.

5.5 Differences from Teledisk

Teledisk is a disk image program from a company called Sydex. Until now, it has been the best available means to archive and share disk images for classic computers, and it has become something of a de-facto standard for this purpose. Sydex no longer supports Teledisk, and is requesting that it be removed from distribution. If you are familiar with Teledisk - please note the following:

- ImageDisk does not try to do everything for you - if you don't know what you are doing, Teledisk will probably work better for you. ImageDisk has more user controllable settings, and can do a better job of extracting and recreating disk images when properly configured.
- ImageDISK does not quietly generate garbage when sectors occur on the disk but cannot be read - ImageDisk will report all read and write errors in the main screen log.
- ImageDisk marks within the image when a sector could not be read.
See: ImageDisk Utility.
- ImageDisk has a documented image file format, which means that you always have the option of exploring other means to recreate a disk This is a MAJOR problem with Teledisk, and the main reason that I created ImageDisk.

5.6 256 Tracks / 255 Sectors?

All too often, we find a product in which some arbitrary limit has been imposed based on "current standards", that causes problems with future enhancements to that standard - a good example of this is the floppy disk controller chips designed in the era of the 77 track 8" drive which generate an error if recalibrating the head to track zero fails after 77 steps (this does not work on an 80 track drive).

In designing ImageDisk, I have chosen to provide support for all of the capabilities of the 765 FDC. This means that as many as 256 tracks, and 255 sectors/track are configurable. It's not that I expect such floppy drives will exist, however it is less likely that ImageDisk will become "useless" because I chose an arbitrary limit.

5.7 Aligning a drive

To properly align a disk drive, you need a special alignment diskette and an oscilloscope, however with ImageDisks Align/Test function, you can usually get "close enough" without special tools.

NOTE: If you don't know what you are doing, you can damage your drive, your PC and possibly yourself - Attempt this only if you are familiar with and comfortable working on mechanical and electronic hardware.

When you execute the (A)lign/Test function, ImageDisk will enter a continuous read loop, providing a running display of the number of sectors read which match the currently seeked track ID, and the number read which don't match. There is also an audible "beep" which increases in pitch the more matching sectors are found.

How you physically adjust head alignment varies from one drive to another. For some you loosen the stepper motor and rotate it. For others, you loosen the whole head carrier assembly and there is either a worm-screw or a slotted cam which moves it back and forth.

To "best guess" align the drive, start with a diskette formatted on a known good drive. Seek to various tracks, and adjust the head position for the highest audible tones. Note the limits where the tone starts to drop, and assume "dead center" between those limits is the best alignment point for that track. Do this at several tracks over the range of head movement and find the best compromise for all positions. Confirm your results with several other known good diskettes from other sources.

You should also check the track-zero detect switch. During the Align/Test function, ImageDisk will show you the status of the track-0 sensor at the top of the screen. Make sure that it comes ON when you step to track 0, and goes OFF when you step to track 1. If not, you may have to adjust the sensor position.

Note that if you have either the alignment or track-0 sensor wrong, you may not be able to adjust the other correctly. If the drive is really bad to begin with, you may have to go back and forth between the two a few times in order to get to the point where you can make the correct adjustments.

6. IMAGE FILE FORMAT

The overall layout of an ImageDisk .IMD image file is:

```

IMD v.vv: dd/mm/yyyy hh:mm:ss           (ASCII Header)
Comment (ASCII only - unlimited size)    (NOTE: You can TYPE a .IMD)
1A byte - ASCII EOF character            (file to see header/comment)
- For each track on the disk:
  1 byte Mode value                      (0-5)
  1 byte Cylinder                        (0-n)
  1 byte Head                            (0-1)      (see Note)
  1 byte number of sectors in track      (1-n)
  1 byte sector size                     (0-6)
  sector numbering map                   * number of sectors
  sector cylinder map (optional)         * number of sectors
  sector head map (optional)            * number of sectors
  sector data records                   * number of sectors
<End of file>

```

6.1 Mode value

This value indicates the data transfer rate and density in which the original track was recorded:

00 = 500 kbps FM	\	Note:	kbps indicates transfer rate,
01 = 300 kbps FM	>		not the data rate, which is
02 = 250 kbps FM	/		1/2 for FM encoding.
03 = 500 kbps MFM			
04 = 300 kbps MFM			
05 = 250 kbps MFM			

6.2 Sector size

The Sector Size value indicates the actual size of the sector data occurring on the track:

```

00 = 128 bytes/sector
01 = 256 bytes/sector
02 = 512 bytes/sector
03 = 1024 bytes/sector
04 = 2048 bytes/sector
05 = 4096 bytes/sector
06 = 8192 bytes/sector

```

ImageDisk does not currently handle disks with differently sized sectors within the same track (the PC FDC cannot write such disks), however an extension to the .IMD file format has been suggested to allow these type of disks to be represented:

A sector size value of 0xFF indicates that a table of sector sizes occurs after the sector numbering/cylinder/head maps (immediately before the data records) which contains one 16-bit value (in little endian format) per sector which defines the actual size of that sector.

6.3 Head value

This value indicates the side of the disk on which this track occurs (0 or 1).

Since HEAD can only be 0 or 1, ImageDisk uses the upper bits of this byte to indicate the presence of optional items in the track data:

Bit 7 (0x80) = Sector Cylinder Map
Bit 6 (0x40) = Sector Head Map

6.4 Sector numbering map

The sector numbering map contains one byte entry containing the physical ID for each sector that occurs in the track.

Note that these entries may NOT be sequential. A disk which uses sector interleave will have a sector numbering map in which the sector numbers occur in non-sequential order.

If ImageDisk is unable to obtain all sector numbers in a single revolution of the disk, it will report "Unable to determine interleave" and rearrange the sector numbers into a simple sequential list.

6.5 Sector Cylinder Map

This is an optional field. Its presence is indicated by bit 7 being set in the Head value for the track.

When present, it means that the cylinder values written to the sectors do NOT match the physical cylinder of the track.

The Sector Cylinder Map has one entry for each sector, and contains the logical Cylinder ID for the corresponding sector in the Sector Numbering Map.

Reading a disk with non-standard cylinder ID's will require the use of the FULL ANALYSIS setting.

6.6 Sector Head map

This is an optional field. Its presence is indicated by bit 6 being set in the Head value for the track.

When present, it means that the head values written to the sectors do NOT match the physical head selection of the track.

The Sector Head Map has one entry for each sector, and contains the logical Head ID for the corresponding sector in the Sector Numbering Map.

Reading a disk with non-standard Head ID's may require the use of the FULL ANALYSIS setting.

6.7 Sector Data Records

For each sector ID occurring in the Sector Numbering Map, ImageDisk records a Sector Data Record - these records occur in the same order as the IDs in the Sector Numbering Map:

```
00      Sector data unavailable - could not be read
01 .... Normal data: (Sector Size) bytes follow
02 xx   Compressed: All bytes in sector have same value (xx)
03 .... Normal data with "Deleted-Data address mark"
04 xx   Compressed with "Deleted-Data address mark"
05 .... Normal data read with data error
06 xx   Compressed read with data error
07 .... Deleted data read with data error
08 xx   Compressed, Deleted read with data error
```

7. IMDA: ImageDisk Analyzer

7.1 Introduction

ImageDisk Analyzer (IMDA) is a utility which examines an ImageDisk .IMD file and suggests possible drive types and options which may be suitable to recreate a disk from that image.

NOTE: This is NOT intended as a replacement for "knowing what you are doing". Getting the right drive type and settings often requires knowledge about the drive type and format information of the original system. IMDA will make suggestions as to what drives/settings could be used to recreate an image, but it does not know of the resultant disk is suitable for that system or your particular use.

7.2 Command line options

Use: IMDA image[.IMD] [options]

opts: /Q - Quiet: inhibit informational output

8. IMDU: ImageDisk Utility

8.1 Introduction

ImageDisk Utility (IMDU) allows you to manipulate ImageDisk .IMD files. The functions which can be performed by IMDU are:

- Display track/sector detail.
- Compare two .IMD files.
- Merge two .IMD files, extracting missing tracks and sectors from one to complete another.
- Remove specific tracks/sides.
- Translate the data rate values.
- regenerate track data with a new interleave factor.
- Compress/Expand "all same" sectors.
- Convert .IMD file to raw binary .BIN format.
- Extract, Append and Replace the image comment record.

8.2 Command line options

Use: `IMDU input-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 data
        /M          - ignore Mode difference in merge/compare
        /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 read]
        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]

```

In general, IMDU works by reading the input-image, and writing the output-image, performing whatever adjustments are specified by the options.

If no adjustment options are made, then IMDU performs a compare of the two images. (Adjustment options are: /B, /C, /E, /NB, /ND, AC=, F=, IL=, RC=, T2=, T3=, T5=, X=, X0= and X1=).

If three image file names are provided, IMDU will merge the first two, combining missing data from one into the other and write the final combined image to the output file. Adjustment options may also be applied during the image merge.

The /B option causes IMDU to remove the comment and all formatting information, outputting what is essentially only the raw sector content of the disk. This is useful for many emulators and other disk image manipulation utilities which do not understand the embedded format information in a .IMD file.

When using /B, IMDU automatically outputs the sector data in order by logical sector number (ie: removes interleave), however if you wish the sector data to be output in some other order, you can use the IL= option to re-interleave the output order.

9. IMDV: ImageDisk Viewer

9.1 Introduction

ImageDisk viewer (IMDV) is a utility which lets you easily look through the content of an image file, and optionally make changes/patches to the content of individual sectors (not recommended unless you really know what you are doing).

IMDV is a low-level tool which presents the data content of sectors in the image file in hexadecimal dump format with either ASCII, EBCDIC or custom character-set display. IMDV does **NOT** understand the arrangement of files, directories or other features of logical volumes which may be contained in the image (these things are highly operating system dependant, and ImageDisk does not assume any particular operating system created the disks).

What IMDV does do is allow you to search through the disk image sectors, view them in HEX/ASCII/EBCDIC/Custom, search for specific strings, and also to extract and display all printable strings occurring from a set point in the image.

9.2 Command line options

Use: `IMDV image[.IMD] [options]`

Opts:	/C	- Case-sensitive searches	
	/E	- use EbcDic displays	
	/I	- ignore Interleave in sector ordering	
	/W	- allow write to file (edit)	
	C=file[.CHR]	- use Custom character-set	[none]
	E=file[.CHR]	- Edit custom character-set	[no]
	M=n	- Minimum scan-string length	[5]
	S=file[.TXT]	- extract scan-Strings to file	[no]
	W=n	- width of scan-string output	[80]

9.3 Custom Character Sets

IMDV runs on a PC, which uses the ASCII character set. It accomodate the viewing of images containing EBCDIC encoded characters, an internal EBCDIC->ASCII translation table is provided.

If you need to use a different character set, you can load your own translation table. This table consists of exactly 256 bytes, each byte containing the ASCII character which should be displayed when the character code corresponding to the position in the file is encountered. NOTE: This is a simple byte translation only. IMDV cannot accomodate character sets of more than 8-bits, nor can it accomodate character sets such as BAUDOT which use encoded shift sequences to select banks of characters.

A very simple character set editor is provided with the E=character-file option.

10. TESTFDC - Evaluate PC floppy controller

As noted in preceding sections, PC floppy controllers vary widely, and it can be tricky to find a system with all the capabilities you need to backup and restore some non-PC formats.

This TESTFDC utility evaluates the floppy disk controller in your system to determine how well it performs at single/double density at various data rates. It also tests MFM operation with 128 byte sectors. Many PCs cannot access these disk types. If you plan to use ImageDisk to archive and restore non-PC format disks, this program should help you find a PC that's up to the task.

Use: TESTFDC drive [type] [options]

drive: Physical floppy drive to test (A: - D:)

type: Override BIOS drive type:
360 = DD 40 track 1.2 = HD 80-track (5.25)
720 = DD 80 track 1.44 = HD 80-track (3.5)

opts: /D - skip standard Double-density test (should always work)
 /R - please Read this!
 R=file - Write final report to file.
 /X - exclude MFM/128 test

You will be prompted to insert low and/or high density media. TESTFDC only writes to the top 5 tracks of the disk, allowing you to use a single disk to boot and to test. TESTFDC should be run from a bare DOS boot (no windows).

NOTE: I am attempting to build a registry of mainboard and add-in floppy controllers and their capabilities. The latest version of this registry can be found with the TESTFDC program at:
<http://www.classiccmp.org/dunfield/img>

If you test a system/controller not already listed, or exhibiting results differing from those that I have on file - please use the R=file option to save the TESTFDC report(s) and send them to me along with:

- The manufacturer and model number of your mainboard or add-in controller
- The floppy controller chip used.

11. Converting from other image file formats

The ImageDisk package includes several utility programs which are useful in converting other disk image types to ImageDisk .IMD file format:

11.1 BIN2IMD: Binary to ImageDisk .IMD utility

The BIN2IMD utility converts raw binary disk images into ImageDisk .IMD format by adding Mode, Cylinder, Head, and Sector information to each track/side record.

BIN2IMD can be useful to convert raw disk images (such as those used by many emulators) into ImageDisk images. It can also be useful in converting images from other sources - even if those images contain embedded format information, it may be easier to convert them to raw binary format and then use BIN2IMD to generate a .IMD file than it is to try and convert directly to the .IMD format.

11.1.1 Command line options

Use: BIN2IMD input-file output-image [option-file] [options]

```

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 cylinders
        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

```

The DM=, SS=, SM=, CM= and HM= options can all be applied to both sides, or one side only. Example:

```

SM=1-9 <= Both sides
SM0=1-9 <= Side0 only
SM1=1-9 <= Side1 only

```

It is therefore possible to create disks which are formatted differently on one side than the other (Most commonly used for formats which require differing sector numbering and/or head mapping on each side).

As a minimum, in order to generate an image, BIN2IMD needs at least the following options applied to all sides that are being generated:

N= (always applies to both sides), DM=, SS= and SM=.

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.

SM= specifies the Sector Numbering Map, which also defines the number of sectors occurring on a track. SM= may consist of any of the following elements:

number = Single sector-number
 number1-number2 = Series ranging from number1 to number2
 number.times = Single value occurring multiple times.

Multiple elements may be combined into a map by separating them with commas:

SM=1,2,3,4,5,6,7,8,9 <= 9 sectors from 1 to 9
 SM=1-9 <= Also 9 sectors from 1 to 9
 SM=9-1 <= 9 sectors from 9 to 1
 SM=1,2-8,9 <= 9 sectors from 1 to 9
 SM=1.9 <= Sector #1 occurring 9 times **

** This (1.9) is actually invalid since sector numbers may occur once within a track, however this format is useful in some of the other lists.

The sector mapping you specify MUST match the order in which the sector data occurs in the binary file. If you wish to create images with differing interleave etc. You must first create the image with the physical data ordering, and then use the IMDU utility to re-interleave it.

HM= specifies a non-standard Head Numbering Map (if not specified, all sectors are assumed to have the physical head number encoded). The number of HM= entries must match the number of sectors defined by SM=.

Here is an example where a disk is created which logically has 20 sectors on Side0, even though they are physically organized as sectors 1-10 on Side0, and 11-20 on Side1:

SM0=1-10 HM0=0.10 SM1=11-20 HM1=0.10

NOTE: The HM0=0.10 isn't strictly necessary, since the default head encoded for Side0 would be 0, however it has been included for it's descriptive value.

CM= specifies a non-standard cylinder encoding. Very rarely used, this option operates similarly to HM=.

The /1 and /2 options are rarely used. These options exist to inform BIN2IMD that a disk is single or double sided in cases where it may not be able to determine this from the parameters.

/2 can be used when both sides are the same, and no Side1 specific options have been specified - Use /2 to let BIN2EXE know that it should generate two sides (You could also just specify one Side1 specific option as the presence of such an option automatically sets double-sided).

/1 or /2 are also used in option files, when the number of sides to be encoded changes part way through the disk. An example of this might be a disk which has system tracks on Side0 only for the first two cylinders, and is double-sided after that. This is quite rare.

/V controls the output of verbose information detail about the diskette format being generated.

```

/V = Generate track format summaries only
/V0 = Generate no detail (Default - use this to turn
      verbose detail output OFF within an option file).
/V1 = Generate detail showing track format and map detail.

```

C= Specifies a comment to be included in the image. A single line of text can be specified. To accommodate the fact that you cannot put spaces in command line parameters, any '~' characters occurring in the comment text will be translated to spaces. You may also include the content of a file with 'C=@filename' - in this case the text file is included with no translation.

11.1.2 Mixed format disks - Command option files

BIN2IMD can create mixed format disks - those in which the format changes from one track to the next. To do this, you must specify the command options to specify the format into an "option" file (type: .B2I), with each set of parameters on a separate line preceded by the cylinder number where those parameters are to take effect.

You may also use a command option file to establish commonly used initial format option. If the first entry in the file is assigned for cylinder 0, then these options need not be specified on the command line.

Blank lines, and lines beginning with ';' are ignored as comments in the command option file.

```

;
; 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

```

11.2 TD02IMD: convert Teledisk .TD0 to .IMD files

Teledisk is a program which performs a similar function to ImageDisk, and for many years was one of the most popular imaging program which could handle non-PC formats. For this reason, it became the primary method by which many people preserved and exchanged system disks for classic computers.

Originally sold as a shareware product by a company called Sydex, all rights to Teledisk were later sold to NTI, who no longer supports it or offers it for sale. This puts Teledisk into a state of limbo, as you cannot legally obtain or use later NTI versions, and the Sydex shareware versions cannot be used beyond the evaluation period without a purchased license, which you currently cannot buy. This is one of several reasons I created ImageDisk as a replacement for Teledisk. I urge anyone creating or manipulating new diskette images to use ImageDisk instead of Teledisk.

TD02IMD reads a teledisk .TD0 image file, and reformats it into an ImageDisk .IMD file. Once you have done this, you can manipulate the disk and recreate it with ImageDisk and it's utilities exactly as you would an image that you recorded directly.

Note that the Teledisk file format is closed and completely undocumented. TD02IMD relies on information obtained by reverse engineering and may not be able to handle all Teledisk images. I recommend that you do not remove the .TD0 files until you have fully tested the resulting .IMD images on your target system.

Use: TD02IMD filename [options]

opts: /Dx,l,h - enable Debug options
 /F - don't assume whole disk is FM if file FM indicator set
 /L - don't convert 300kbps (LD in HD) to 250 (standard LD)
 /Q - Quiet: less informational output.
 /RF - remove Repeated (duplicate) sectors, keep First
 /RL - remove Repeated (duplicate) sectors, keep Last
 F=ACDIN - remove sectors based on status Flags
 A=deletedaddressmark C=crcerror D=dos N=nodata I=noid
 M=n - Maximum sector number (remove any higher)
 W=CDHIN - inhibit warnings:
 C: CRC error (bad sector)
 D: Duplicate sector numbers
 H: Head not 0 or 1
 I: No ID field - possible bogus sector
 N: No DATA field

The /RF, /RL, F= and M= options are useful to remove bogus sectors that TD sometimes inserts into a disk image when it has trouble reading the disk.

Early versions of Teledisk use a single bit in the disk header to indicate single-density, which suggests that it was "all or nothing" - later versions have a bit in the track header, allowing mixed single and double density tracks. I have made the assumption that if the global SD bit is turned on, the image is from an early Teledisk version and should be recreated entirely in single density, however I do not know this for certain. The /F option allows you to disable this assumption.

Debug options are mainly for my own use in resolving compatibility problems, 'x' is a byte mask in hex, consisting of the OR of these bits:

80 = Display disk header	02 = Internal sector info
40 = Display track headers	01 = Stop after one track
20 = Display sector headers	

'l' and 'h' are low and high track numbers. If specified, Debug output will only be produced for the indicated track range.

TD02IMD is a work in progress. If you encounter .TD0 files that it cannot convert, please notify me with the details so that I can look into it.

NOTE: The Teledisk file format allows for different sector sizes within a track - the PC 765 floppy controller cannot format disks in this manner, and ImageDisk does not currently support it (A future version may support this ability as a "read only" feature). Until such time as it does, TD02IMD will not convert such images.

11.3 DMK2IMD: convert .DMK images to ImageDisk .IMD

DMK is a popular disk image format for emulators, because it stores all track data including formatting fields, and has no compression options. This allows disks to be read/written by the emulator exactly as a physical disk would be, and in many cases can support copy protection and other features relating to non-standard physical formatting.

DMK2IMD reads a .DMK format image file and converts it to an ImageDisk .IMD format file.

Use: DMK2IMD filename[.DMK] [options]

```

Opts:  /C[filename]   - insert Comment record
        /H          - force High density      (500kbps)
        /L          - force Low density       (250kbps)
        /S          - force Single-density image
        /UD         - allow User defined DAM - treat as Deleted
        /UN         - allow User defined DAM - treat as Normal
        /W          - continue after warning
        O=file[.IMD] - specify output file           [filename.IMD]

```

DMK has no capability to store a comment record. The /C option allows you to provide a .IMD comment. If a file is specified, it will be read and inserted into the comment record as ASCII text. If no file is specified, you will be prompted to enter the comment (press ^Z to exit).

DMK includes no information about the physical drive type or controller settings used to create the image. In particular, it does not contain any indication as to the data rate/density of the original disk. DMK2IMD makes a "guess" based on the size of the tracks, however you can override this and force /High or /Low density.

DMK requires special handling for single-density. If a disk is declared to be double-density, single-density data is recorded twice to preserve sector spacing - DMK has a bit in the header indicating single-density disks, however it's not obvious if this has been supported from the very beginning. I have provided the /S option to force DMK2IMD to assume a single-density disk (no double data read).

The PCs 765 controller cannot handle the "user defined" DAM (Data Address Marks) which occur in some DMK images - DMK2IMD will abort if it finds such an address mark, however the /UN and /UD options can be used to force it to treat the user defined DAM as either a Normal or Deleted DAM.

The /w option allows DMK2IMD to continue processing after certain errors which would otherwise be considered fatal:

Condition	Action taken with /w
Bad Sector offset	Sector not recorded
IAM not found	Sector not recorded
ID crc error	Sector not recorded
DAM not found	Sector recorded, no data
Data crc error	Sector recorded as bad

NOTE: The DMK format can represent disks with different sector sizes or densities within a single track - the PC 765 floppy controller cannot format disks in this manner, and ImageDisk does not currently support it (A future version may support these abilities as a "read only" feature). Until such time as it does, DMK2IMD will not convert such images.

11.4 ANY2IMD : Convert ANY format to .IMD by floppy write/read

If you have on other way to convert an image file, the "last resort" is write the image to a floppy disk using whatever tool was designed for that purpose, and then read it back with ImageDisk. This works, however it gets tedious because you need to run both the disk-creator and ImageDisk separately for each file you wish to convert, entering the filename twice (once for each program), and noting and re-entering the comment field.

ANY2IMD simplifies this process process by building a list of files of the foreign image type and automatically entering the names into the programs, as well as capturing the comment field displayed by the foreign imaging program and entering it into the .IMD file.

The whole process looks like this:

- Set up a .INI file for ANY2IMD and the foreign disk creator (see later)
- Set up a blank diskette in an appropriate drive, and enter the command: ANY2IMD *.typ <= Command to convert ALL .typ files
- The foreign disk-creation program runs
- ANY2IMD enters keystrokes necessary to start creating the disk, enters the filename, and display the file comment record (if any).
- ANY2IMD displays a "capture" box over the comment area. You can move or resize the box with the arrow keys
- F1 toggles moving (alcorners), and resizing (lower right corner). Press ENTER - ANY2IMD will capture the comment from the screen, and continue.
- Proceed to create the disk, then exit.
- ANY2IMD creates a "blank" .IMD file containing the comment captured from the foreign disk-creator program screen.
- ImageDisk runs...
- ANY2IMD enters the keystrokes to 'R'ead a disk, and enters the filename.
- The comment is read from the file created above and displayed. Press 'Y' to proceed.
- Edit the comment if desired, and proceed to make the disk.
- Exit ImageDisk - if there are more .typ files to process, the process will repeat.

This seems like a lot, however it's really quite simple, and goes much more quickly than manually entering the filenames and comments. You can also automate the process of selecting drives and options with the ANY2IMD options below:

11.4.1 ANY2IMD options:

ANY2IMD supports a number of options to control it's operation. While these can be entered on the command line, it is usually best to place these in a file called ANY2IMD.INI which will be processed automatically when ANY2IMD starts. It first looks for ANY2IMD.INI in the current directory, and then if not found, looks in the directory where ANY2IMD.COM is located. Options in the ANY2IMD.INI file must occur one per line. Blank lines, and lines beginning with ';' are ignored.

WE=filename [options]
RE=filename [options]

Specify the write/Read executable file.
You must enter a COMPLETE path - ANY2IMD does NOT scan your PATH.
- RE= will normally end with "IMD.COM".
Within the ANY2IMD.INI file you can also specify command line options (you cannot do so on the interactive command line).
The special sequence ~FN can be used to substitute the current filename into the command line options.

WS=string
RS=string

Specify a "script" of keys to be automatically entered with the write or read program runs. Note that embedded spaces can only be entered when this option occurs within the ANY2IMD.INI file.
ASCII characters are entered exactly as they occur on the string, along with special sequences identified by '~':

~~	= Single '~' character
~F1-~F0	= Function key 1-10
~S1-~S0	= Shift-Function key 1-10
~C1-~C10	= Control-Function key 1-10
~A1-~A10	= Alt-Function key 1-10
~EN	= Enter
~BS	= BackSpace
~TB	= Tab
~BT	= BackTab (Shift-Tab)
~ES	= ESCape
~IN	= INsert
~DL	= DeLete
~K+	= Keypad +
~K-	= Keypad -
~LA	= Left Arrow
~RA	= Right Arrow
~UA	= Up Arrow
~DA	= Down Arrow
~HO	= HOme
~EN	= ENd
~PU	= PageUp
~PD	= PageDown
~WAn;	= wait 'n' milliseconds [note]
~FN	= enter the Filename
~CC	= Capture the comment record
~ST	= Stop processing - press CTRL-END to continue

[note] waiting works only when the application being executed is polling for keys using INT 16h/AH=01h - If the application requests a key using INT 16h/AH=00h, it will be suspended until the next key, so waiting serves no purpose and is ignored (this is normally used to wait until an application is ready for input, but polls for other keys in the interm).

By carefully noting the exact keystrokes needed to prepare the disk creation program or Imagedisk for the operation you desire, you can use these options to automate those keystrokes.

CC=x,y,width,height

These options allow you to set the exact region of the screen with is captured for the image file comment. The capture area is a box with the X, Y coordinates giving the upper left-hand position (0,0=Top left), extending to the right and down by the width, Height values.

FD=x,y

Causes ANY2IMD to try and display the current filename at the indicated coordinates on the screen when the write/read program is running. Choose and area which does not interfere with normal screen display.

/P

Normally ANY2IMD will pause before running each write or Read command, allowing you to Proceed, Skip or Stop all commands - This is useful when you are debugging your script strings, however once you have the process working, the /P option can be used to disable the pause and have ANY2IMD simply run the commands as quickly as possible.

/Ifile

Specifies the .INI file that ANY2IMD will read for it's default options (Has no effect within the .INI file) - the default .INI filename is 'ANY2IMD'.

11.4.2 Sample ANY2IMD.INI file using Teledisk

```

;
; Load all .TD0 (Teledisk) files
;
;*.TD0

;
; Define Teledisk comment display/capture area
;
CC=4,10,73,9

;
; Teledisk executable file - no options
;
WE=c:\teledisk\teledisk.exe

;
; Write script: Use Teledisk to write image to floppy:
;
; ~RA      - Right Arrow:  move to 'Copy File to Disk'
; ~EN      - ENter:        proceed to next screen
; ~RA      - Right Arrow:  move to 'B:' (select drive B:)
; ~EN      - ENter:        proceed to filename input
; ~FN      - send FileName
; ~EN      - ENter:        proceed to comment display
; ~CC      - Capture Comment
; ~EN      - ENter:        Proceed to make disk
WS=~RA~EN~RA~EN~FN~EN~CC~EN

;
; ImageDisk exec - /H to skip help, /B for drive B:
;
RE=c:\imd\imd.com /H /B

;
; Read script: Use ImageDisk to Read floppy into image
;
; R        - 'R':         Activate Read function
; ~FN      - send FileName
; ~EN      - ENter:        proceed
; Y        - 'Y':         answer 'Y' to continue prompt
RS=R~FN~ENY

;
; Put filename display at lower left of screen
;
FD=0,24

```