

mfm_read can analyze and/or read a MFM disk and write raw MFM transition data, decoded data, or emulation data to files.

mfm_util can read transition and convert to emulation or decoded data. It can also read emulation file data and convert to decoded data.

mfm_read and mfm_util both use similar command options.

--analyze -a [=cyl,head]

Analyze disk format. If cylinder and head is specified it will analyze the specified track for header format. No spaces can be in the optional parameters. --analyze=10,4

--begin_time -b #

The number of nanoseconds to delay from index to start reading track

--cylinders -c #

The number of cylinders.

--data_crc -j #h,#h,#h[,#h]

The CRC/ECC parameters for the sector data area. Initial value, polynomial, polynomial length, maximum ECC span.

--drive -d #

Drive number to select for reading. Only valid for read command. Drives are number 1 to 4.

--emulation_file -m filename

File name to write emulation bit data to. No file created if not specified

--extracted_data_file -e filename

File name to write decoded data to. No file created if not specified. If drive has metadata for the sectors it will be extracted to filename.metadata. Currently only Xerox 6085 has metadata

--format -f WD_1006 | OMTI_5510 | DEC_RQDX3 | Xebec_104786

The track format. Not all formats listed. Use --format help to list all currently supported formats. Some formats will also set header_crc, data_crc, sectors, and sector_length.

Parameters specified after format will override values.

--head_3bit -3

Selects header 3 bit head encoding used by WD 1003 controller. Default is 4 bit. This will not be detected by analyze. The wrong number of heads may be selected.

--header_crc -g #h,#h,#h[,#h]

The CRC/ECC parameters for the sector header. Initial value, polynomial, polynomial length, maximum ECC span.

--heads -h #

The number of heads.

--interleave -i # | #,#,#,#,...

The logical sector numbers from header in physical sector order or the interleave value

--note -n "string"

String is stored in header of transition and emulation file for information about image. mfm_util will display.

--retries -r #[,#]

Select number of retries on read errors. Only valid for read command. The optional second parameter specifies the number of retries before seeking off track and back. Default 50,4.

--sector_length -l #

The sector data area length in bytes. Default is 512.

--sectors -s #[,#]

The number of sectors per track, lowest sector number.

--transitions_file -t filename

File name to write raw MFM transitions to. No file created if not specified. Only valid for read

command.

--quiet -q #h
Bit mask to select which messages don't print. 0 is print all messages. Default is 1 (no debug messages). Higher bits are more important messages in general.

--unbuffered_seek -u
Use unbuffered/ST506 seeks. Default is buffered/ST412.

--version -v
Print program version number.

Long options can be abbreviated to the shortest unique name. Option values can't have spaces unless quoted as a string.

is a number. #h is a number which may be decimal, octal if starts with a 0, or hex starting with 0x. The Cyclic Redundancy Check (CRC) / Error Correcting Code (ECC) parameters consists of a initial value which the CRC register is set to before starting the CRC, the CRC bit length, a CRC polynomial, and a maximum ECC span. The ECC span should be zero if ECC correction should not be used.

It is advisable to generate a transition file when reading important disks since it may be possible to reprocess in the case of errors either in reading or due to errors in these programs.

32 bit or longer polynomials may be usable for ECC even if the original controller only used them for CRC. The quality of the polynomial chosen determines the miss-correction probability. Most 32 bit polynomials specify 5 bit correction though some say they can be used for up to 11 bit correction.

Emulation file is for use by the mfm_emu program to emulate a disk drive. For mfm_read it is an output. For mfm_util it is an output if a transitions file is specified otherwise it is an input.

If extract file is specified mfm_read will attempt to decode the track data. If the disk format is known an extract file should be specified even if one is not needed since mfm_read will then reread tracks that have errors until it exceeds the error count or gets a successful read of the track. The extract format is the sector data from the drive written in logical sector order. If controllers use alternate cylinders the data for those cylinders is not moved to the logical cylinder location.

Begin_time is for drives that have a sector straddle the start of the index pulse. For reading to work properly all the data must be read consecutively. Set this parameter to the time in nanoseconds the first physical sector is delayed from the index pulse. It is needed for Corvus model H and NorthStar Advantage drives.

NOTE: Some computers use different format on different tracks. Analyze can only handle one format at a time. This may cause it to determine the wrong number of cylinders or other parameters. The published specification for the drive should be checked against what analyze determine and the command arguments adjusted as needed.

For retrying read errors on some drives seeking helps to recover the error since the heads end up at a slightly different location. On a drive that is unhappy the extra seeks bad be bad. It may be advisable to do the first read with -retries 50,51 to prevent any seeks and then go back to the default or adjust the parameters to try to recover more data.

The Adaptec format uses logical block addresses (LBA) instead of cylinder head sector (CHS) in the

header. This program does not have enough information to determine the CHS for the LBA in the header so if a seek error occurs it can not recover. Messages like Missing LBA address # to # where the second number is less than the first or large range may indicate a seek error. Some disks have a large jump near the end of the disk due to the spare and bad sector handling.

Examples:

To read an unknown format drive

```
mfm_read --analyze --transitions_file raw_data --extracted_data_file extracted_data --note "Drive from TI Professional computer read November 7 2014"
```

Analyze is conservative on maximum ECC burst length for correction. You may rerun with the command line printed and change the header or data_crc last parameter to increase ECC span with higher probability of miss correction. If you expected sector doesn't match sector found on some of the tracks the drive may have different interleave on some tracks. Try rerunning with the command parameters printed except leave off --interleave.

To store raw transitions without decoding (change parameters to match your drive)

```
mfm_read --transitions_file raw_data --drive 1 --heads 6 --cylinders 640
```

To process previously read raw transitions data if analyze determined decoding parameters or they were manually specified when file created:

```
mfm_util --transitions_file raw_out2 --extracted_data_file /tmp/decoded_out
```

Otherwise specify parameters for your drive like:

```
mfm_util --sectors 17,0 --heads 6 --cylinders 640 --header_crc 0x2605fb9c,0x104c981,32,0  
--data_crc 0xd4d7ca20,0x104c981,32,0 --format OMTI_5510 --sector_length 512 --interleave  
0,3,6,9,12,15,1,4,7,10,13,16,2,5,8,11,14 --transitions_file raw_out2 --extracted_data_file  
/tmp/decoded_out
```

--emulation_file may be specified instead of --transitions_file to convert emulation file to extracted data.

Analysis typical messages. Explanation in italic:

AM33XX

File prucode0.bin open passed

Informational messages from PRU access routines.

Found drive at select 2

Informational. Select line drive responded to.

Returning to track 0

Informational. This operation may take a while.

Drive RPM 3594.4

Informational. Drive should be close to 3600 RPM

Matches count 34 for controller OMTI_5510

Header CRC: Polynomial 0x104c981 length 32 initial value 0x2605fb9c

Sector length 512

Data CRC: Polynomial 0x104c981 length 32 initial value 0xd4d7ca20

Number of heads 6 number of sectors 17 first sector 0

Informational. What we found about the disk format. Sometimes multiple formats will be shown. The code will retry on a different cylinder to try to determine correct format. The matches can either be false matches or some drives such as DEC RQDX3 use multiple formats. This code does not deal well with that. You can manually read with the different formats and put the data back together.

Interleave (not checked): 0 3 6 9 12 15 1 4 7 10 13 16 2 5 8 11 14

Informational. If it can't determine interleave the disk can still be read and decoded. Interleave is only checked if interleave is specified on the command line. Too many drives have tracks with different interleave which generated confusing messages.

Drive supports buffered seeks (ST412)

Informational.

No sectors readable from cylinder 640

Stopping end of disk search due to two unreadable tracks in a row

Number of cylinders 640, 33.4 MB

Informational. The method we used to determine the number of cylinders and size determined.

Command line to read disk:

```
--sectors 17,0 --heads 6 --cylinders 640 --header_crc 0x2605fb9c,0x104c981,32,0 --data_crc  
0xd4d7ca20,0x104c981,32,0 --format OMTI_5510 --sector_length 512 --retries 50 --drive 2  
--interleave 0,3,6,9,12,15,1,4,7,10,13,16,2,5,8,11,14
```

This is the options needed to decode the disk with mfm_read. For mfm_util remove --retries and --drive from options. Change header_crc and/or data_crc last parameter if you wish to use different ECC maximum span. Remove interleave if you get expected sector doesn't match sector found errors.

It is recommended to verify results of analysis since it can make mistakes.

Decoding messages:

AM33XX

File prucode0.bin open passed

Informational messages from PRU access routines in mfm_read.

Returning to track 0

Informational. This operation may take a while. Only mfm_read.

Retries failed cyl 22 head 0

Unable to recover all data from the specified track. Only mfm_read.

Bad sectors on cylinder 22 head 0: 3 15H

Indicates that for the specified track that data of sector 3 has uncorrected errors and the header for sector 15 has errors. If the header has an error the data will be all zero.

ECC Corrections on cylinder 56 head 5: 1(2) 13(4) 15(1H)

Informational. Indicates that for the specified track that data of sectors 1 and 13 were corrected and the header of sector 15 corrected. The number in the parenthesis is the length of the bit pattern corrected. Unless the ECC correction had a false correction the data is good.

Cyl 59 head 2 Missed sector between 12(4) and 1(6)

Informational. Indicates that the sectors found didn't match the interleave specified. The first number is the sector header number and the number in parenthesis the physical sector starting with 0. A bad sector message will be printed if the data is bad. Some disks vary the interleave. Try again without the interleave option.

Missing LBA address 2576 to 2579

Indicates some LBA addresses in the single value or range specified were not found. The data for those sectors was not recovered. Some drives have an intentional jump in the LBA address near the end of the disk when sectors marked bad or spare reported is non zero.

Good data after 20 Retries cyl 219 head 0

Informational. Indicates we recovered good data by retrying the read.

Found cyl 0 to 639, head 0 to 5, sector 0 to 16

Informational. Should match what was specified

Expected 65280 sectors got 65276 good sectors, 0 bad header, 4 bad data

0 sectors marked bad or spare

11 sectors corrected with ECC. Max bits in burst corrected 5

Summary of errors during the read. We have 4 sectors with bad data. Sectors marked bad or spare are one the drive format has either marked bad to not use or reserving for spare sectors.

Errors:

Command 4 fault 300 status 1002c

Not Write fault

Not Seek complete

Not Index

Ready

Drive selected

Not Track 0

This indicates a command failed. See cmd.h for definitions. Status is the drive status bits which are decoded in text below. This error was the drive did not give seek complete in the expected time. These are normally fatal and the program will exit. Some are recovered from during analysis and are informational.