NAME

fdcontrol - display and modify floppy disk parameters

SYNOPSIS

fdcontrol [**-F**] [**-d** *dbg*] [**-f** *fmt*] [**-s** *fmtstr*] [**-v**] *device*

DESCRIPTION

The **fdcontrol** utility allows the modification of the run-time behavior of the fdc(4) driver for the device specified by *device*.

Commands are implemented to query the current device density settings as well as the underlying device hardware as registered with the driver, to manipulate debugging levels, and to adjust the device density settings. All the operations that manipulate the kernel settings are restricted to the superuser (by the device driver), while all inquiry requests only require read access to *device*.

The *device* argument should always be given as a full path name, e.g. */dev/fd0*.

Inquiry Commands

Running the **fdcontrol** utility without any of the optional flags will report the drive type that is registered with the device driver. In the shortest form, a single string describing the drive type will be returned. Possible values are: "360K", "1.2M", "720K", "1.44M", "2.88M", or "unknown". This information is primarily intended to be easily parsable by scripts.

In order to add some descriptive text that makes the output better human readable, the flag **-v** can be added.

Specifying flag **-F** will report the device's density settings in a form that is suitable as input to the **-s** *fmtstr* option (see below). Again, together with **-v**, some more text will be returned, including the total capacity of the density settings in kilobytes.

Debug Control

The fdc(4) control utilities support two different options how to specify device density settings. The first form uses **-f** *fmt* to specify the format of the medium in kilobytes. Depending on the underlying drive type, the value is compared against a table of known commonly used device density settings for that drive, and if a match is found, those settings will be used. Currently, the following values for the respective drive types are acceptable:

2.88M and 1.44M drives:

KB	sect	rac	secsize	ncyls	speed	heads	flags
1721	21	2	(512)	82	500	2	MFM

147618	2 (512)82	500	2	MFM
144018	2 (512)80	500	2	MFM
120015	2 (512)80	500	2	MFM
82010	2 (512)82	250	2	MFM
80010	2 (512)80	250	2	MFM
7209	2 (512)80	250	2	MFM

1.2M drives:

KB	**sectrac**	**secsize**	**ncyls**	**speed**	**heads**	**flags**
120015	2	(512)	80	500	2	MFM
12328	3	(1024)	77	500	2	MFM
147618	2	(512)	82	500	2	MFM
144018	2	(512)	80	500	2	MFM
120015	2	(512)	80	500	2	MFM
82010	2	(512)	82	300	2	MFM
80010	2	(512)	80	300	2	MFM
7209	2	(512)	80	300	2	MFM
3609	2	(512)	40	300	2	MFM,2STEP
6408	2	(512)	80	300	2	MFM

720K drives:

KB	**sectrac**	**secsize**	**ncyls**	**speed**	**heads**	**flags**
7209	2	(512)80	250	2	MFM	

360K drives:

KB	**sectrac**	**secsize**	**ncyls**	**speed**	**heads**	**flags**
3609	2	(512)40	250	2	MFM	

The second form to specify a device density uses **-s** *fmtstr* to explicitly specify each parameter in detail. The argument *fmtstr* is a comma-separated list of values of the form:

sectrac,secsize,datalen,gap,ncyls,speed,heads,f_gap,f_inter,offs2,flags

The meaning of the parameters is:

sectrac The number of sectors per track.

secsize The sector size code, 0 = 128 bytes (or less), 1 = 256 bytes, 2 = 512 bytes, 3 = 1024 bytes.

datalen

The actual sector size if the size code is 0, or the (ignored) value 0xFF for larger size codes.

gap The length of the gap 3 parameter for read/write operations.

ncyls The number of cylinders.

speed The transfer speed in kilobytes per second. Can be 250, 300, 500, or 1000, but each drive type only supports a subset of these values.

heads The number of heads.

f_gap The length of the gap 3 when formatting media.

f_inter The sector interleave to be applied when formatting. 0 means no interleave, 1 means 1:1 etc.

offs2 The offset of the sector numbers on side 2 (i.e., head number 1). Normally, sector numbering on both sides starts with 1.

flags A list from one of the following flag values:

+mfm Use MFM encoding.
-mfm Use FM (single-density) encoding.
+2step Use 2 steps per each cylinder (for accessing 40-cylinder media in 80-cylinder drives).
-2step Do not use 2 steps per cylinder, i.e., access each physical cylinder of the drive.
+perpend Use perpendicular recording (for 2.88 MB media, currently not supported).
-perpend Use longitudinal recording.

For any missing parameter, the current value will be used, so only actual changes need to be specified. Thus to turn off a flag bit (like **+mfm** which is the default for all drive types), the form with a leading minus sign must explicitly be used.

EXAMPLES

A simple inquiry about the drive type:

```
$ fdcontrol /dev/fd0
1.44M
```

Same as above, but with verbose output. Note that the result is about the *drive type*, as opposed to a *device density*, so it is independent from the actual subdevice being used for *device*.

```
$ fdcontrol -v /dev/fd0
/dev/fd0: 1.44M drive (3.5" high-density)
```

Inquiry about the density settings:

```
$ fdcontrol -F /dev/fd0
18,512,0xff,0x1b,80,500,2,0x6c,1,0,+mfm
```

The verbose flag makes this human readable:

```
/dev/fd0: 1440 KB media type
        Format:         18,512,0xff,0x1b,80,500,2,0x6c,1,0,+mfm
        Sector size:    512
        Sectors/track:  18
        Heads/cylinder: 2
        Cylinders/disk: 80
        Transfer rate:  500 kbps
        Sector gap:     27
        Format gap:     108
        Interleave:     1
        Side offset:    0
        Flags           <MFM>
```

As indicated, trailing commas in the parameter list may be omitted.

In order to access archaic 160 KB single-density (FM encoded) 5.25 media in a modern 1.2M drive, something like the following definition would be needed. (Note that not all controller hardware is actually capable of handling FM encoding at all.)

```
# fdcontrol -s 16,128,0x80,0x2,40,300,,0x10,,,-mfm,+2step /dev/fd1.1
```

It is still possible to hook up 8" drives to most modern floppy controllers, given the right cable magic. (On PC hardware, tell the BIOS that it is a 5.25" drive.) The classical 128/26/2/77 format can be read with this entry

```
fdcontrol -s  26,128,0x80,0x2,77,500,2,0x10,,,-mfm /dev/fd0
```

SEE ALSO

fdc(4)

HISTORY

The **fdcontrol** utility appeared in FreeBSD 2.0, and was vastly overhauled in FreeBSD 5.0.

AUTHORS

The program and this man page was contributed by Jörg Wunsch, Dresden.