

# Forum:[Hack Forums](#)

Thread: [\[Noob Friendly\] Understanding /dev](#)

*XMR - October 1, 2015, 1:40 pm*

This thread is meant for mainly new Linux users to understand what is inside the /dev folder and what uses they each have. I will briefly explain some of the files in /dev and what each file can be used for, and whether it should be used at all. The /dev folder (dev short for device) contains special device files that should generally be left alone, but it is useful to have some understanding on what they're doing there. Again, this is just a shitty guide to some files in /dev.

Constructive criticism welcome, but don't be an asshole.

Please don't make any stupid ass posts like "Why would people need this?" either  
If something here is incorrect or difficult to understand, please tell me!

These files can be read from and written to using the dd command, like so:

dd if=<input file> of=<output file> bs=<block size> count=<blocks to write>

If count is not specified, dd will continue reading/writing until end of file.

/dev/block/ - Contains links to devices that handle data in blocks.

/dev/char/ - Contains links to devices that handle data character by character rather than in blocks.

/dev/console - System console (what you see when booting your system), view by running the xconsole command.

/dev/cdrom | /dev/cdrw | /dev/dvd | /dev/dvdrw - Links to your CD/DVD drive at /dev/sr0. Generally shouldn't be tampered with.

/dev/disk/ - Contains links to your hard disks and USB drives.

/dev/fb\* - Frame buffers, more info on that [here](#). Basically a memory buffer for GUI frames.

/dev/full - When written to, will return a "no space left on device" or "device full" type error. When read from, will produce a continuous stream of zero value bytes.

Attempting to write 1 byte to /dev/full. Notice the error and bytes copied.

Code root@arccotangent-kali:/dev# dd if=/dev/zero of=/dev/full bs=1 count=1

dd: error writing '/dev/full': No space left on device

1+0 records in

0+0 records out

0 bytes (0 B) copied, 0.000655918 s, 0.0 kB/s

/dev/full hexdump output. Note the ^C interruption at the end.

Code root@arccotangent-kali:/dev# hexdump /dev/full

00000000 0000 0000 0000 0000 0000 0000 0000 0000 0000

\*

^C

/dev/mem - All data stored in RAM. You can view memory by running memdump OR memdump | hexdump or hexdump /dev/mem. You can also save RAM to a file for later viewing with dd  
if=/dev/mem of=<output file> bs=2M.

/dev/null - When written to, will destroy all data written to it. Useful for trashing data. When read from, will produce no output.

/dev/random - A non-deterministic random byte generator. Non-deterministic means it is impossible to predict its output based on its current state. When read from, will produce a small, non-continuous amount of random bytes. Random bytes are generated from environmental noise collected from device drivers and sources such as the time between keystrokes.

Some /dev/random output. This command ran for about 7 seconds before termination. All bytes were printed at once.

```
Code root@arccotangent-kali:/dev# hexdump /dev/random
00000000 55d3 5062 b704 b914 439a db09 55e4 d9d5
00000010 3055 b233 1ade ecca c2b3 7b71 6397 c203
00000020 706a 4569 b3e2 cb12 8b0d 33ff f065 d3db
00000030 db34 0f00 5dc6 5a3f 3836 cd3e 87e5 12db
00000040 4e83 4db1 ab75 749c 4f9c 9beb dd25 e1f6
00000050 84e4 7cc9 17a0 6edb d695 8a33 c94e c651
00000060 5db4 af68 5036 40a2 2bde 4715 3623 bcc2
00000070 8e87 4cc2 ad5c 6fa7 98e5 4f4a b233 69ab
00000080 c15d 7420 ed85 e432 3bb7 9839 aff7 4abe
00000090 3242 bdf7 370b 5a9c 3e2d 459b c87b 2f7c
^C
```

/dev/sd\*\* - Disks sorted by letters (eg. /dev/sda, /dev/sdb), and each partition on said disks sorted by numbers (eg. /dev/sda0, /dev/sda1, /dev/sda2). These files allow raw I/O to disks, and as such, are often used to create bootable USB drives. Generally shouldn't be tampered with unless creating bootable drives. WARNING: Writing data to the wrong disk will corrupt data on it and may prevent your system from booting properly, if at all.

/dev/stdout | /dev/stdin | /dev/stderr - Standard output, input, and error streams, respectively.

/dev/tty\* - Terminals. You should know what a terminal is if you use Linux.

/dev/urandom - Similar to /dev/random. When read from, will produce a continuous stream of random bytes with potentially less entropy. To save some random bytes to a file, run `dd if=/dev/urandom of=<output file> bs=<byte count> count=1`.

Some /dev/urandom output, trimmed. This command ran for about 7 seconds before termination. Output was an continuous stream of random data.

```
Code root@arccotangent-kali:/dev# hexdump /dev/urandom
00000000 51a2 fbd3 cfd5 4d37 fe5c c7cc 4a91 aea9
00000010 1959 fb07 e8e2 bb8d 0bad aecf 9398 b61f
00000020 0e54 9faa fbdb 6bbc e0dd 73fc e180 36a4
00000030 edab 82ae 9015 b6d4 709b 9cfa 1932 5bec
00000040 3e72 258c cf5e 192c d97d b0d5 f562 785f
00000050 e14a 24e5 9329 ab81 9f88 2dbd 50ce 3a7e
...
030fa00 8e20 1982 d956 dcbf 5106 a20b abef a158
030fa10 d91b 519c 9962 f7e0 200d 5923 15b4 e2ae
030fa20 0b88 0bc0 32a8 3436 41c4 85ef e42f 77a4
030fa30 ec46 87c5 73aa 1eea b85b 9d5b 9196 dfc4
030fa40 679f 27b7 5ed4 b72a 2a6d 093b 776f cdd1
030fa50 2015 d004 3f56 9e3d 0b94 0b75 e558 74ad
```

/dev/video\* - Video devices such as webcams.

/dev/watchdog - When read from, will reboot your system. Can only be read by root.

/dev/zero - When written to, will destroy all data written to it like /dev/null. When read from, will produce a continuous stream of zero value bytes. Useful if you need a file of arbitrary size but don't care about its contents. Copying from /dev/zero is much faster than others, as seen below.

/dev/zero output. Same as /dev/full, and ran for about 3 seconds before termination.

Code root@arccotangent-kali:/dev# hexdump /dev/zero

00000000 0000 0000 0000 0000 0000 0000 0000 0000 0000

\*

^C

Copying from /dev/zero vs /dev/urandom to /dev/null. Note the write speed difference.

Code root@arccotangent-kali:/dev# dd if=/dev/zero of=/dev/null bs=32M count=1

1+0 records in

1+0 records out

33554432 bytes (34 MB) copied, 0.0834846 s, 402 MB/sCode root@arccotangent-kali:/dev# dd

if=/dev/urandom of=/dev/null bs=32M count=1

0+1 records in

0+1 records out

33554431 bytes (34 MB) copied, 9.36582 s, 3.6 MB/s

tuki - October 2, 2015, 3:22 pm

Sweet overview.

Since you mentioned /dev/block you should also add /dev/char.