

dmraid(8) - Linux man page

Name

dmraid - discover, configure and activate software (ATA)RAID

Synopsis

dmraid {-a|--activate} {y|n|yes|no}
[-d|--debug]... [-v|--verbose]... [-i|--ignorelocking]
[-f|--format FORMAT[,FORMAT...]]
[{-P|--partchar} CHAR]
[-p|--no_partitions]
[-Z|--rm_partitions]
[--separator SEPARATOR]
[-t|--test]
[RAID-set...]

dmraid {-b|--block_devices}
[-c|--display_columns][FIELD[,FIELD...]]...
[-d|--debug]... [-v|--verbose]...
[--separator SEPARATOR]
[device-path...]

dmraid {-h|--help}

dmraid {-l|--list_formats}
[-d|--debug]... [-v|--verbose]...

dmraid {-n|--native_log}
[-d|--debug]... [-v|--verbose]... [-i|--ignorelocking]
[-f|--format FORMAT[,FORMAT...]]
[--separator SEPARATOR]
[device-path...]

dmraid {-R| --rebuild}
RAID-set
[device-path]

dmraid {-x| --remove}
[RAID-set]

dmraid -f FORMAT-handler
{-C| --create} set --type raidlevel

```
[--size=setsize --strip stridesize]
--disk "device-path, device-path [, device-path ...]"
```

```
dmraid [ -f|--format FORMAT-handler] -S|--spare [RAID-set] -M|--media "device-path"
dmraid {-r|--raid_devices}
[-c|--display_columns][FIELD[,FIELD...]]...
[-d|--debug]... [-v|--verbose]... [-i|--ignorelocking]
[-D|--dump_metadata]
[-f|--format FORMAT[,FORMAT...]]
[--separator SEPARATOR]
[device-path...]
```

```
dmraid {-r|--raid_devices}
[-d|--debug]... [-v|--verbose]... [-i|--ignorelocking]
[-E|--erase_metadata]
[-f|--format FORMAT[,FORMAT...]]
[--separator SEPARATOR]
[device-path...]
```

```
dmraid {-s|--sets}...[a|i|active|inactive]
[-c|--display_columns][FIELD[,FIELD...]]...
[-d|--debug]... [-v|--verbose]... [-i|--ignorelocking]
[-f|--format FORMAT[,FORMAT...]]
[-g|--display_group]
[--separator SEPARATOR]
[RAID-set...]
```

```
dmraid {-V/--version}
```

Description

dmraid discovers block and software RAID devices (eg, ATARAID) by using multiple different metadata format handlers which support various formats (eg, Highpoint 37x series). It offers activating RAID sets made up by 2 or more discovered RAID devices, display properties of devices and sets (see option **-l** for supported metadata formats). Block device access to activated RAID sets occurs via device-mapper nodes `/dev/mapper/RaidSetName`. RaidSetName starts with the format name (see **-l** option) which can be used to access all RAID sets of a specific format easily with certain options (eg, **-a** below).

Options

```
-a, --activate {y|n} [RAID set...]
```

Activates or deactivates all or particular software RAID set. In case metadata format handlers are chosen with **-f**, only RAID sets with such **format(s)** can be activated or

deactivated. Useful if devices have multiple metadata signatures. When activating RAID sets, **-p** disables the activation of partitions on them, and **-Z** will make dmraid tell the kernel to remove the partitions from the disks underlying the set, ie if sda is part of the set, remove sda1, sda2, etc. This prevents applications from directly accessing the disks bypassing dmraid. RAID set names given on command line don't need to be fully specified (eg, "dmraid -ay sil" would activate all discovered Silicon Image Medley RAID sets).

{-b|--block_devices} [device-path...]

List all or particular discovered block devices with their properties (size, serial number). Add **-c** to display block device names only and **-cc** for CSV column output of block device properties. See description of **-c** below for FIELD identifiers.

[-d|--debug]...

Enable debugging output. Option can be given multiple times increasing the debug output level.

[-c|--display_columns][FIELD[,FIELD...]]...

Display properties of block devices, RAID sets and devices in **column(s)**. Optional list specifying which FIELDS to display.

For **-b**:

d[evpath]|p[ath], sec[tors]|si[ze], ser[ialnumber].

For **-r**:

de[vpath]|p[ath], f[ormat], r[aidname], t[ype], st[atus], se[ctors]|si[ze], da[taoffset]|o[ffset].

For **-s**:

f[ormat], r[aidname], t[ype], sta[tus], str[ide], se[ctors]|si[ze], su[bsets], d[evices], sp[ares].

[-f|--format FORMAT[,FORMAT...]]

Use metadata format **handler(s)** to discover RAID devices. See **-l** for a list of supported format handler names. This is useful to select particular formats in case multiple metadata signatures are found on a device. A comma seperated list of format names can be specified which may not contain white space.

{-h|--help}

Display help text.

{-i|--ignorelocking}

Don't take out any locks. Useful in early boot where no read/write access to /var is available.

{-l|--list_formats}

List all available metadata format handlers with their names and descriptions. Supported RAID levels are listed in parenthesis:

S: Span (concatination)

0: RAID0 (stripe)

1: RAID1 (mirror)

10: RAID10 (mirror on top of stripes)

01: RAID10 (stripe on top of mirrors) Note: Intel OROM displays this as RAID10

{-n|--native_log} [device-path...]

Display metadata in native, vendor-specific format. In case a metadata format handler is chosen with **-f** only RAID devices with such format will be displayed in native format. If **device-path(s)** is/are given on the command line, native metadata output is restricted to those listed.

[{-P|--partchar} CHAR]

Use CHAR as the separator between the device name and the partition number.

{-R|--rebuild} RAID-set [device-path]

Rebuild raid array after a drive has failed and a new drive is added. For Intel chipset based systems, there are two methods in which a new drive is added to the system.

1. Using OROM to identify a new drive During system reboot, enter OROM and mark the new drive as the rebuild drive.

After booting to the OS, use the dmraid command to rebuild.

Example: dmraid -R raid_set

2. Using dmraid to identify a new drive Boot to the OS and use the dmraid command with the new drive as the second parameter.

Example: dmraid -R raid_set /dev/sdc

3. Using hot spare drive Mark a drive as hot spare using the "dmraid -f isw -S" command. Then use the dmraid command to start the rebuild.

Example: dmraid -R raid_set

{-x|--remove} [RAID-set]

Delete one or all existing software RAID devices from the metadata.

-f FORMAT-handler {-C|--create} --type raidlevel [--size=setsize --strip stripsize] --disk "device-path, device-path [,device-path]"

Delete one or all existing Configure a software RAID device and store the configuration data in a group of hard drive devices consisting of this array. This command requires the following options:

-f FORMAT-handler

metadata format (see "dmraid -l")

--type digit[digit...]

specify the raid level of the software RAID set.

0: raid0

1: raid1

5: raid5

01: raid01 (isw raid10)

--size: [digits[k|K|m|M|g|G][b|B]]

specify the size of the RAID set. The number is an integer followed by [kKmMgG] and/or [bB].

b: byte (default)

B: block (512 bytes)

K or K: on the base of 1024

m or M: on the base of 1024*1024

g or G: on the base of 1024*1024*1024

If this option is missing, it's set to the default value pre-configured by the vendor. Note that different vendors may apply different constraints on the granularity of the size or the minimal value.

--strip: [digits[k|K|m|M|g|G][b|B]]

specify the strip size of a RAID1, RAID5, and RAID10 RAID set (as above)

--disk: device-path[{,| }device-path...]

specify the array of the hard drives, e.g. /dev/sda.

-f FORMAT-handler -S -M device-path

-S -M device-path

This command adds hot spare support for one or more RAID sets.

1. When used with a format handler, which supports hot spare sets (e.g. isw), a hot spare is marked to be used when rebuilding any RAID set of that format. 2. When used when specifying a RAID set, the drive is added to that RAID set and will be used only to rebuild that set. Note: If the specified name does not match an existing RAID-set, a set with the new name will be created.

{-r|--raid_devices} [device-path...]

List all discovered RAID devices with format, RAID level, sectors used and data offset into the device. In case a metadata format handler is chosen with **-f**, only RAID devices with such format can be discovered. Useful if devices have multiple metadata signatures. If **-D** is added to **-r** the RAID metadata gets dumped into a subdirectory named dmraid.format_name (eg. format_name = isw) in files named devicename.dat. The byte offset where the metadata is located on the device is written into files named devicename.offset and the size of the device in sectors into files named devicename.size.

If **-E** is added to **-r** the RAID metadata on the devices gets conditionally erased. Useful to erase old metadata after new one of different type has been stored on a device in order to avoid discovering both. If you enter **-E** option **-D** will be enforced in order to have a fallback in case the wrong metadata got erased. Manual copying back onto the device is needed to recover from erasing the wrong metadata using the

dumped files devicename_formatname.dat and devicename_formatname.offset. Eg, to restore all *.dat files in the working directory to the respective devices:

```
for f in *.dat
do
dd if=$f of=/dev/${f%%.dat} \

seek='cat ${f%%.dat}.offset' bs=1

done
```

If **device-path(s)** is/are given on the command line, the above actions are restricted to those listed. Add **-c** to display RAID device names only and **-cc** for CSV column output of RAID device properties. See description of **-c** above for FIELD identifiers.

--separator SEPARATOR

Use SEPARATOR as a delimiter for all options taking or displaying lists.

-s... [a|i] [RAID-set...]

Display properties of RAID sets. Multiple RAID set names can be given on the command line which don't need to be fully specified (eg, "dmraid -s hpt" would display all discovered Highpoint RAID sets). Enter **-s** twice to display RAID subsets too. Add **-c** to display names of RAID sets only, **-cc** for CSV column output of RAID set properties and **-ccc** for inclusion of block devices in the listing. Doesn't imply **-s -s** to show RAID subsets (implied for group sets, e.g. isw). Add **-g** to include information about group RAID sets (as with Intel Software RAID) in the listing. See description of **-c** above for FIELD identifiers. Note: Size is given in sectors (not bytes).

[-v|--verbose]...

Enable verbose runtime information output. Option can be given multiple times increasing the verbosity level.

Examples

"dmraid -l" lists all supported metadata formats with their names along with some descriptive information, eg:

```
hpt37x : (+) Highpoint HPT37X
hpt45x : (+) Highpoint HPT45X
isw : (+) Intel Software RAID
lsi : (0) LSI Logic MegaRAID
nvidia : (+) NVidia RAID
pdc : (+) Promise FastTrack
sil : (+) Silicon Image(tm) Medley(tm)
via : (+) VIA Software RAID
dos : (+) DOS partitions on SW RAIDs
(0): Discover, (+): Discover+Activate
```

"dmraid -ay" activates all software RAID sets discovered.

"dmraid -an" deactivates all active software RAID sets which are not open (eg, mounted

filesystem on them).

"dmraid -ay -f pdc" (pdc looked up from "dmraid -l") activates all software RAID sets with Promise format discovered and ignores all other supported formats.

"dmraid -r" discovers all software RAID devices supported on your system, eg:

```
/dev/dm-46: hpt45x, "hpt45x_chidjhaiaa-0", striped, ok, 320172928 sectors, data@ 0
/dev/dm-50: hpt45x, "hpt45x_chidjhaiaa-0", striped, ok, 320172928 sectors, data@ 0
/dev/dm-54: hpt45x, "hpt45x_chidjhaiaa-1", striped, ok, 320172928 sectors, data@ 0
/dev/dm-58: hpt45x, "hpt45x_chidjhaiaa-1", striped, ok, 320172928 sectors, data@ 0
```

"dmraid -s -s hpt45x_chidjhaiaa" displays properties of set "hpt45x_chidjhaiaa", eg:

*** Superset

name : hpt45x_chidjhaiaa

size : 640345856

stride : 128

type : raid10

status : ok

subsets: 2

dev : 4

spare : 0

---> Subset

name : hpt45x_chidjhaiaa-0

size : 640345856

stride : 128

type : stripe

status : ok

subsets: 0

dev : 2

spare : 0

---> Subset

name : hpt45x_chidjhaiaa-1

size : 640345856

stride : 128

type : stripe

status : ok

subsets: 0

dev : 2

spare : 0

"dmraid -s -ccs hpt45" displays properties in column format of all sets and subsets with hpt45* format, eg:

hpt45x_chidjhaiaa,640345856,128,raid10,ok,4,0

hpt45x_chidjhaiaa-a,640345856,128,stripe,ok,2,0

hpt45x_chidjhaiaa-b,640345856,128,stripe,ok,2,0

"dmraid -r --sep : -cpath:size" display paths and sizes in sectors for RAID devices in

column format using ':' as a delimiter, eg:

```
/dev/dm-8:320173055  
/dev/dm-12:320173055  
/dev/dm-22:320173055  
/dev/dm-26:320173055  
/dev/dm-30:586114703  
/dev/dm-34:586114703  
/dev/dm-38:586114703  
/dev/dm-42:586114703  
/dev/dm-46:156301487  
/dev/dm-50:156301487  
/dev/dm-54:390624896  
/dev/dm-58:390624896  
/dev/dm-62:390624896  
/dev/dm-66:390624896
```

"dmraid -f isw -C Raid0 --type 0 --strip 8k --size 20g --disk "/dev/sdb /dev/sdc"" creates an ISW volume with a name of "Raid0", 20Gig bytes in total, and 8kilo bytes strip size on two disks.

"dmraid -f isw -C Test0 --type 0 --disk "/dev/sdd /dev/sde"" creates an ISW volume with the default size and strip size.

"dmraid -f isw -C Test10 --type 01 --strip 128B --disk "/dev/sda /dev/sdb /dev/sdc /dev/sdd" creates a stacked RAID device, RAID10 (isw format), with a name of "Test10", 128 blocks (512bytes) strip size , and the default volume size on 4 disks.

"dmraid -f isw -S -M /dev/sde" marks the device /dev/sde as a hot spare for rebuild

"dmraid -R isw_djaggchdde_RAID1 /dev/sde" starts rebuild of the RAID volume on device /dev/sde

Diagnostics

dmraid returns an exit code of 0 for success or 1 for error.

Author

Heinz Mauelshagen <Mauelshagen@RedHat.com>