

NAME

graid3 - control utility for RAID3 devices

SYNOPSIS

graid3 label [-Fhnrvw] [-s *blocksize*] *name prov prov prov ...*

graid3 clear [-v] *prov ...*

graid3 configure [-adFFhnrRvwW] *name*

graid3 rebuild [-v] *name prov*

graid3 insert [-hv] [-n *number*] *name prov*

graid3 remove [-v] -n *number name*

graid3 stop [-fv] *name ...*

graid3 list

graid3 status

graid3 load

graid3 unload

DESCRIPTION

The **graid3** utility is used for RAID3 array configuration. After a device is created, all components are detected and configured automatically. All operations such as failure detection, stale component detection, rebuild of stale components, etc. are also done automatically. The **graid3** utility uses on-disk metadata (the provider's last sector) to store all needed information.

The first argument to **graid3** indicates an action to be performed:

label Create a RAID3 device. The last given component will contain parity data, whilst the others will all contain regular data. The number of components must be equal to 3, 5, 9, 17, etc. ($2^n + 1$).

Additional options include:

-F

Do not synchronize after a power failure or system crash. Assumes device is in consistent state.

-h

Hardcode providers' names in metadata.

-n

Turn off autosynchronization of stale components.

-r

Use parity component for reading in round-robin fashion. Without this option the parity component is not used at all for reading operations when the device is in a complete state. With this option specified random I/O read operations are even 40% faster, but sequential reads are slower. One cannot use this option if the **-w** option is also specified.

-s

Manually specify array block size. Block size will be set equal to least common multiple of all component's sector sizes and specified value. Note that array sector size calculated as multiple of block size and number of regular data components. Big values may decrease performance and compatibility, as all I/O requests have to be multiple of sector size.

-w

Use verify reading feature. When reading from a device in a complete state, also read data from the parity component and verify the data by comparing XORed regular data with parity data. If verification fails, an EIO error is returned and the value of the *kern.geom.raid3.stat.parity_mismatch* sysctl is increased. One cannot use this option if the **-r** option is also specified.

clear Clear metadata on the given providers.

configure Configure the given device.

Additional options include:

-a

Turn on autosynchronization of stale components.

-d

Do not hardcode providers' names in metadata.

-f

Synchronize device after a power failure or system crash.

-F

Do not synchronize after a power failure or system crash. Assumes device is in consistent state.

-h

Hardcode providers' names in metadata.

-n
Turn off autosynchronization of stale components.

-r
Turn on round-robin reading.

-R
Turn off round-robin reading.

-w
Turn on verify reading.

-W
Turn off verify reading.

rebuild Rebuild the given component forcibly. If autosynchronization was not turned off for the given device, this command should be unnecessary.

insert Add the given component to the existing array, if one of the components was removed previously with the **remove** command or if one component is missing and will not be connected again. If no number is given, new component will be added instead of first missed component.

Additional options include:

-h
Hardcode providers' names in metadata.

remove Remove the given component from the given array and clear metadata on it.

stop Stop the given arrays.

Additional options include:

-f
Stop the given array even if it is opened.

list See geom(8).

status See geom(8).

load See geom(8).

unload See geom(8).

Additional options include:

-v
Be more verbose.

EXIT STATUS

Exit status is 0 on success, and 1 if the command fails.

EXAMPLES

Use 3 disks to setup a RAID3 array (with the round-robin reading feature). Create a file system, mount it, then unmount it and stop device:

```
graid3 label -v -r data da0 da1 da2
newfs /dev/raid3/data
mount /dev/raid3/data /mnt
...
umount /mnt
graid3 stop data
graid3 unload
```

Create a RAID3 array, but do not use the automatic synchronization feature. Rebuild parity component:

```
graid3 label -n data da0 da1 da2
graid3 rebuild data da2
```

Replace one data disk with a brand new one:

```
graid3 remove -n 0 data
graid3 insert -n 0 data da5
```

SEE ALSO

geom(4), geom(8), gvinum(8), mount(8), newfs(8), umount(8)

HISTORY

The **graid3** utility appeared in FreeBSD 5.3.

AUTHORS

Pawel Jakub Dawidek <pjd@FreeBSD.org>

BUGS

There should be a section with an implementation description.

Documentation for `sysctls kern.geom.raid3.*` is missing.