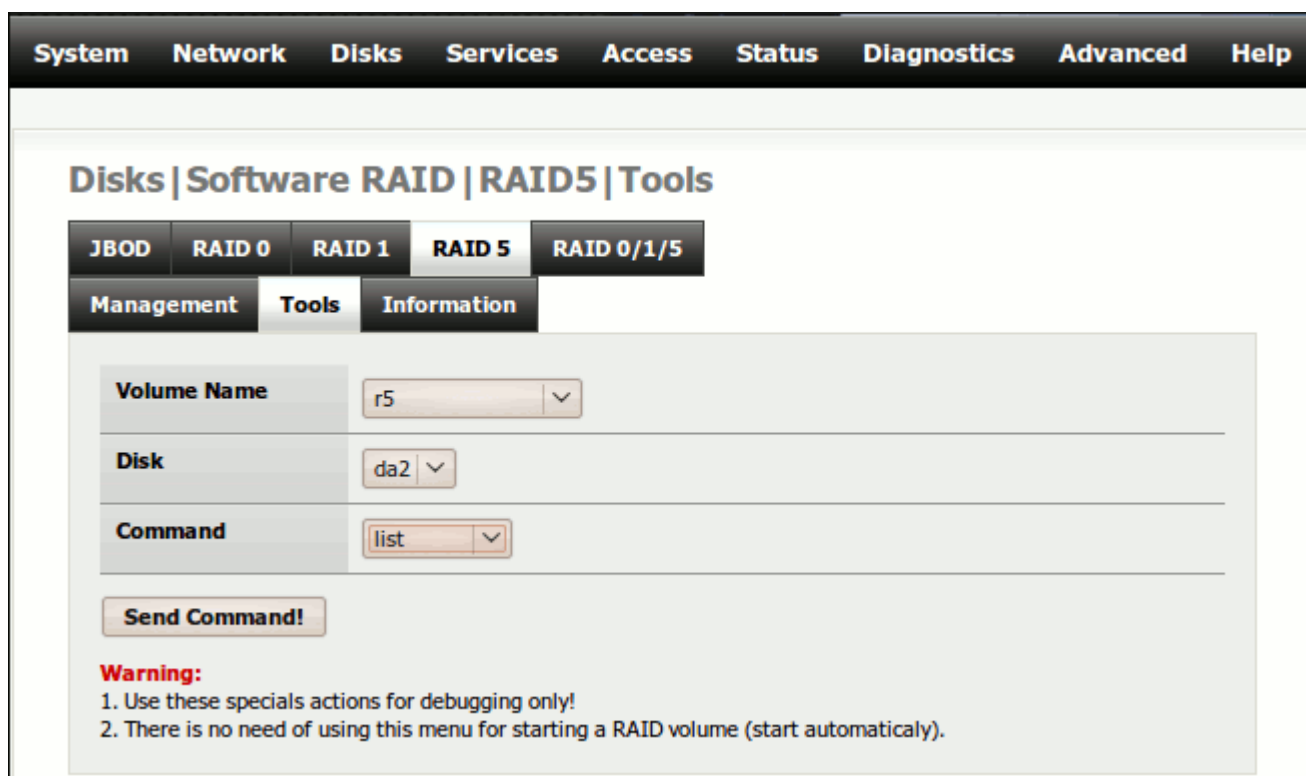


Disks Software RAID RAID5 Tools



The screenshot shows a web interface for managing RAID5 arrays. At the top is a navigation bar with links: System, Network, Disks, Services, Access, Status, Diagnostics, Advanced, and Help. Below this is a header section titled "Disks | Software RAID | RAID5 | Tools". Under the header, there are two rows of tabs. The first row contains "JBOD", "RAID 0", "RAID 1", "RAID 5" (which is selected), and "RAID 0/1/5". The second row contains "Management", "Tools" (which is selected), and "Information". The main content area has three dropdown menus: "Volume Name" with "r5" selected, "Disk" with "da2" selected, and "Command" with "list" selected. Below these is a "Send Command!" button. A "Warning:" section follows, containing two points: "1. Use these specials actions for debugging only!" and "2. There is no need of using this menu for starting a RAID volume (start automaticaly)."

This Tab is used for the troubleshooting and management of SoftRAID5 arrays created with [Geom_raid5](#).

Volume Name: Allows you to select an existing RAID5 volume (array, provider) to manage or troubleshoot.

Disk: Allows you to select a disk (consumer, device, array member) to manage or troubleshoot.

Command: Select the graid5 command to execute on the volume and/or disk chosen above. Press the "**SEND COMMAND!**" to execute the command.

Note - /sbin/graid5 is the utility executing the commands you choose.

As of today (2010/09/20) it appears graid5 is not part of official FreeBSD distributions so the man page is not easy to find & access. This is the current [graid5 home page](#) where the latest versions and man files reside.



To make life easier we offer the reproduction below along with additional notes at the end. Any resulting errors or deviations from Arne Worner's original are due to subsequent poor editing on my part. - Al

The version of graid5 used in the latest NAS4Free is slightly different, but not much, there are different options in some of the commands and this article will be updated as soon as I can track down the most recent version.

GRAID5(8) BSD System Manager's Manual GRAID5(8)

****NAME****

== ==

****graid5**** – control utility for raid5 devices

=====

****SYNOPSIS****

== ==

```
**graid5 destroy** [**-fvy**] //name ...// **\ graid5 label** [**-hnSv**]  
[**-s **//stripeSize//] //name prov prov ...// **\ graid5 configure**  
[**-hnRS**] //name// **\ graid5 stop** [**-fv**] //name ...// **\ graid5  
insert** //name prov// **\ graid5 remove** //name prov// **\ graid5  
clear** [**-v**] //prov ...// **\ graid5 dump** //prov ...// **\ graid5  
list \ graid5 status \ graid5 load \ graid5 unload**
```

=====

****DESCRIPTION****

== ==

The ****graid5**** utility is used for setting up a RAID-5 on two or more disks. The RAID5'ed device can be configured using two different methods: **'manual'** or **'automatic'**. When using the **'manual'** method, no metadata are stored on the devices, so the RAID5 device has to be configured by hand every time it is needed. The **'automatic'** method uses on-disk metadata to detect devices. Once devices are labeled, they will be automatically detected and configured.

The first argument to ****graid5**** indicates an action to be performed:

=====

****label****

== ==

Set up a RAID5 device from the given devices with the specified **//name//**. This is the **'automatic'** method, where metadata are stored in every device's last sector. The kernel module **//geom_raid5.ko//** will be loaded if it is not loaded already.

Additional options include:

****-h****

Hardcode providers' names in metadata.

****-c****

CowOp mode: Complete-Only-Write-Operation -- dont write if not in status COMPLETE.

****-S****

SafeOp mode: read the whole stripe for every read and verify parity.

****-n****

never-hot-mode: A 2 disk graid5 device doesnt need the hot marker, if it is used as swap space. Furthermore this flags is useful, if a rebuild would be harmful even if a write request was pending.

****_s**** //stripesize//

Specify stripesize. Recommendation: MAXPHYS (currently 128KiB) == stripesize. The //stripesize// must be a power of 2 and a multiple of the largest sector size of all the providers.

=====

****configure****

== ==

Configure an existing graid5 device:

Options are:

****_h****

Trigger: hardcoded option.

****_a****

Reset error flag of all disks.

****_c****

CowOp mode: Complete-Only-Write-Operation -- dont write if not in status COMPLETE.

****_n****

Trigger: never-hot-mode option.

****_S****

Trigger: SafeOp-mode option.

****_R****

Trigger: start/stop re-sync.

=====

****stop****

== ==

Turn off an existing RAID5 device by its //name//. This command does not touch on-disk metadata!

Options are:

****_f****

Force destroy even if still busy.

****_y****

Do not do the Yo-Yo effect.

=====

****destroy****

== ==

Same as ****stop****.

=====

****clear****

== ==

Clear metadata on the given devices.

=====

****dump****

== ==

Dump metadata stored on the given devices.

=====

****list****

== ==

See `geom(8)`.

=====

****status****

== ==

See `geom(8)`.

=====

****load****

== ==

See `geom(8)`.

=====

****unload****

== ==

See `geom(8)`.

=====

****Additional options:****

== ==

****_f****

Force the removal of the specified striped device.

****_v****

Be more verbose.

=====

****SYSCTL VARIABLES****

== ==

The following `sysctl(8)` variables can be used to control the behavior of the ****RAID5**** GEOM class. The default value is shown next to each variable.

`//kern.geom.raid5.debug//: 0`

Debug level of the ****RAID5**** GEOM class. This can be set to a number between

0 and 3 inclusive. If set to 0 minimal debug information is printed, and if set to 3 the maximum amount of debug information is printed.

//kern.geom.raid5.mhm//: 0 (read-only)

Number of malloc hamster cache misses.

//kern.geom.raid5.mhh//: 0 (read-only)

Number of malloc hamster cache hits.

//kern.geom.raid5.maxmem//: 8000000 (tunable)

This variable can be set any time to any 32bit signed integer value. It is cropped appropriately (0..128MB) and interpreted as bytes.

//kern.geom.raid5.wqf//: 0 (read-only)

This value shows the number of write requests that were issued early due to a conflicting read request.

//kern.geom.raid5.wqp//: 0 (read-only)

This value shows the maximum number of pending write requests so far.

//kern.geom.raid5.blked1//: 0 (read-only)

This value shows the number of new write requests that could not be combined because the corresponding area already has an issued but incomplete write request.

//kern.geom.raid5.blked2//: 0 (read-only)

This value shows number of due write (2-phase) requests, that were blocked by another such request due to parity area conflict.

//kern.geom.raid5.dsk_ok//: 50 (read-only)

This value shows the healthiness of the underlying devices. 50 is perfect. 40 or lower triggers a soft-device-remove. 0 causes an error announced to the upper layer.

//kern.geom.raid5.veri_nice//: 100 (tunable)

This value (milli seconds) enforces a delay after a user-land read request for internal verify requests, which are certainly quite hindering for user-land requests, because they read all disks and in some cases even write a disk.

//kern.geom.raid5.veri_w//: 0 (read-only)

This value shows the number of parity-failures (during rebuild)

```
//kern.geom.raid5.veri//: 0 (read-only)
```

This value shows the number of parity checks (during rebuild).

```
//kern.geom.raid5.wreq2_cnt//: 0 (read-only)
```

Number of 2-phase writes (1. phase: read data&parity (or "other" data in case of three disks); 2. phase: write data&parity).

```
//kern.geom.raid5.wreq1_cnt//: 0 (read-only)
```

Number of 1-phase writes (sufficiently long chunks can be written in one phase).

```
//kern.geom.raid5.wreq_cnt//: 0 (read-only)
```

Write requests started by upper layer.

```
//kern.geom.raid5.rreq_cnt//: 0 (read-only)
```

Read requests started by upper layer.

```
//kern.geom.raid5.maxwql//: 0 (tunable)
```

This variable gives a hint for the maximum length of the write queue. Write requests are queued until they are long enough or old enough or until there are too many of them.

```
//kern.geom.raid5.wdt//: 10 (tunable)
```

This variable determines the maximum age of a write request before it is issued.

```
//kern.geom.raid5.tooc//: 3 (tunable)
```

This variable determines the time-out-on-create. The provider is not created before all consumers are present or the timeout is over.

```
=====
```

```
**EXIT STATUS**
```

```
== ==
```

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

```
=====
```

```
**EXAMPLES**
```

```
== ==
```

The following example shows how to set up a RAID5 device from four disks with a 128KB stripe size for automatic configuration, create a file system on it, and mount it:

```

graid5 label -v -s 131072 data /dev/da0 /dev/da1 /dev/da2 /dev/da3 \\\ newfs
/dev/raid5/data \\\ mount /dev/raid5/data /mnt \\\ [...] \\\ umount /mnt \\\
graid5 stop data \\\ graid5 unload
=====
**COMPATIBILITY**
== ==
The **graid5** interleave is in number of bytes, unlike ccdconfig(8) and
atacontrol(8) which use the number of sectors. A ccdconfig(8) //ileave// of
'128' is 64 KB (128 512B sectors). The same stripe interleave would be
specified as '65536' for **graid5**.
=====
**SEE ALSO**
== ==
geom(4), loader.conf(5), atacontrol(8), ccdconfig(8), geom(8), mount(8),
newfs(8), sysctl(8), umount(8), vinum(8)
=====
**HISTORY**
== ==
The **graid5** utility appeared in FreeBSD 5.3.
=====

BSD Dec 11, 2006 BSD

```

Additional Information:

There are several “cause phrase” in case of a meta data update:

1. **verify forced**: means that the user forced a parity block check.
2. **verify aborted**: means that the user switched manually from state REBUILDING to state COMPLETE (which might be a bad idea; e. g. if ALL disks were pre-initialized with zeroes, the rebuild isnt necessary)
3. **new configuration**: the configuration was changed by the user and should be written to disk as soon as possible.
4. **verify completed**: the verify run is done, so that we can switch from state REBUILDING to state COMPLETE. In state COMPLETE there is no message at default debug level (0) about CALM↔HOT transitions.
5. **store verify progress**: Every 3 minutes we store the verify progress, so that we do not lose too much work, if the box crashes (again) or is shutdown.
6. **store verify flag**: mark the device as “to be verified”
7. **valid disk count**: A disk was removed or added, which should be noted in the meta data as soon as possible.
8. state **hot**: the device is marked as “hot” (that means: write requests pending); this message is usually at debug level 1 and in case of a running verify process it is at debug level 0. In order to reduce the amount of meta data updates, a graid5 device stays HOT some time (.wdt seconds, which is by default 10 sec currently) after the write cache of graid5 has been emptied...
9. state **calm**: the device is marked as “calm” (all parity is synchronized); otherwise **see 8**.

From:
<https://www.xigmanas.com/wiki/> - **XigmaNAS**

Permanent link:
https://www.xigmanas.com/wiki/doku.php?id=documentation:setup_and_user_guide:disks_software_raid_raid5_tools

Last update: **2018/07/08 17:57**

