

About the Install

The NVRAID BIOS is actually a software raid. The method used to install here will work with other operating systems such as Windows with no configuration changes needed with Windows.

These instructions will work with many BIOS RAID (Fake RAID) devices. Though, the actual device names will vary.

The performance of dmraid is often less than that which can be obtained from linux software raid (mdraid). For instance, dmraid-raid1 only reads from a single disk, while mdraid-raid1 will load-balance reads between all disks

Make sure you follow the instructions in chapter 3 of the handbook for setting up your network, and in chapter 6 about copying over DNS information, as you may need to download files to complete this process.

In Linux 2.4 the Ataraid kernel framework provided support for software RAID assisted by the BIOS. For kernel 2.6 the device-mapper runtime can do the same kind of work, but it's configured in userspace however. Heinz Maulshagen introduced the dmraid tool to create RAID mappings using the kernel device-mapper subsystem. The controllers supported are (mostly cheap) Fake-RAID IDE / SATA controllers which have BIOS functions on it. Most common ones are: Promise Fasttrak controllers as well as HPT 37x, Intel and LSI. Also serial ata RAID controllers like Silicon Image Medley and Nvidia Nforce are supported by the program. Support for VIA Raid has been introduced in the latest version. While all of the controllers listed previously should work with this method, it was tested with an NVRAID on an nForce2 chipset-based board.

This howto does not cover everything in the installation process. It merely explains how to get a Gentoo installation started with the LiveCD, and then jumps in at the end to explain how to get GRUB installed and boot into Linux after the first restart. This howto has been tested with a stage1, stage3, and a stage 1/3 install, so I have no doubts that this howto will not have any incompatibilities with other installation methods.

Booting

Loading dmraid with the LiveCD

To install Gentoo in the first place, you must be able to have access to the raid. When you are at the Gentoo LiveCD autorun, make sure to add the „dodmraid“ option.

```
gentoo dodmraid
```

This will ensure that the dmraid drivers are loaded, as they are not loaded by default as of this writing (2006.0).

Testing dmraid

To verify that dmraid will work with your configuration, check the contents of the `/dev/mapper` directory.

```
ls /dev/mapper
```

If there is just a „control“ file, then dmraid was either not loaded, or dmraid will not work with your configuration. If you are using a 2006.1 LiveCD, try using a more recent LiveCD instead. If there is still only a „control“ file in **/dev/mapper**, try to activate the RAID set with the **dmraid -a y** command.

With Adaptec Hostraid / ICH5 SATA RAID this error message was produced:

```
ERROR: asr: Invalid RAID config table checksum (0xE11 vs. 0x805) on /dev/sdb
```

Is this really still necessary? sadly Yes, Ill testet with install-amd64-minimal-2008.0-r1.iso, the question is, how to repair a broken raid-1 with this bug}} One workaround is to initialize the raid using windows setup cd. Format/Partition the array and then hit reset when it begins to copy files over and boot with the gentoo live cd, delete the windows partition and create/format the Linux partition(s).

Partitioning

Suggested Layout

This howto assumes that your boot partition is the first partition on the raid. If you set your **/boot** partition as the first partition on the disk, you will likely save yourself the need of a floppy disk and a reboot later (Method 2 of installing GRUB mentioned later in this guide). This will not confuse Windows as long as you install Windows onto your second partition (or whichever partition you like). Here is a sample layout for dual-booting with Windows

- Partition 1: /boot
- Partition 2: NTFS (for Windows)
- Partition 3: /swap
- Partition 4: /

This partition layout is just an example. The **/boot** partition does not need to be the first partition (but you will need to use a slightly more difficult method to install GRUB if it is not the first).

Using fdisk to Partition

When you partition your drives, do not use **/dev/sda1**, etc. Although Gentoo knows that a raid is there, it also knows that there are two disks and is not afraid to treat them as such. Be sure to fdisk

your raid as a whole, by referring to its mapped location found in **/dev/mapper**.

```
fdisk /dev/mapper/nvidia_abiccada
```

Your raid name will be different.

Upon completing fdisk the partition table needs to be reloaded, so the kernel is aware of the changes.

Easy Method

```
reboot
```

Remember to include the dodmraid option when you start the LiveCD environment again.

"Fast" Method

Execute the following command:

```
dmraid -ay
```

Problem 1: the dmraid binary seems not to be on the gentoo live cd, so i copied it from another gentoo installation. Don't forget to run:

```
chmod 755 dmraid
```

Problem 2: if there already was one or more partition on the raid, dmraid will not correctly reload the partition table; see easy method.

If you're not seeing any partitions after this, (/dev/mapper/nvidia_abiccada1, 2 3 etc) and you get errors like „Arguments out of bounds“ when running

```
dmraid -ay
```

, it might be caused by the fact that there are spaces in the array name. Use a tool like Intel Matrix Storage Manager to rename your array to something without spaces*

Intel Storage Manager seems to be able to rename an array without any data loss. Depending on the tools you're using, this may vary.

Mounting and Installation

Mounting the RAID

After partitioning your raid, mount the partitions accordingly and use your favorite method to install

Gentoo. Remember to mount the drives as the mapped drive and not as `/dev/sda1`, etc.

```
mount /dev/mapper/nvidia_abiccada4 /mnt/gentoo
mkdir /mnt/gentoo/boot
mount /dev/mapper/nvidia_abiccada1 /mnt/gentoo/boot
```

`/proc` and `/dev` can be mounted in the usual manner:

```
mount -t proc none /mnt/gentoo/proc
mount -o bind /dev /mnt/gentoo/dev
```

Building the Kernel

This guide prefers building the kernel yourself over using genkernel. When you are building your kernel, remember to add support for the RAID (and compile as static (*), not as a module (M)):

```
RAID Support
Device Drivers
-> Block devices
  <*> Loopback device support
  <*> RAM disk support
    (16) Default number of RAM disks
    (8192) Default RAM disk size (kbytes)
    (1024) Default RAM disk block size (bytes)
  [*] Initial RAM filesystem and RAM disk (initramfs/initrd) support
-> SCSI device support
  <*> RAID Transport Class
  <*> SCSI device support
    -> SCSI low-level drivers
    -> Serial ATA (SATA) support
      <*> Choose the RAID chipset of your choice
  [*] Multiple devices driver support (RAID and LVM)
  <*> RAID support
    <*> Choose your RAID method (striping/mirroring/etc)
  <*> Device mapper support
    <*> (I marked everything under here, but only because I'm too lazy to
troubleshoot)
```

Make sure you include needed suboptions (ex: mirror target).

With Silicon Image 3112 it works fine if you build the device-mapper as module.

RAID support and RAID Transport Class are not really needed if you're really using fakeraid.

When you get to the point of installing GRUB, return to this guide.

Installing dmraid

Emerge dmraid (this may be masked or hard-masked, so be sure to unmask it).

If hard-masked:

```
echo "sys-fs/dmraid" >> /etc/portage/package.unmask
```

If masked (change `~x86` to your arch, `~amd64` for example):

```
echo "sys-fs/dmraid ~x86" >> /etc/portage/package.keywords
```

GRUB Setup and Configuration

This howto assumes that you are using the GRUB bootloader. Be sure that it is installed before continuing.

```
emerge -a grub
```

Grub by itself will not work with a raid0/raid1 configuration. You may be able to get your kernel to boot, but it will panic before you reach a prompt, preventing you from using Gentoo at all. The solution is to create a initial ram disk that will load the dmraid drivers and allow your system to boot.

Creating the Initial RAM Disk

Easy Solution (using genkernel)

You can just use genkernel to do all the work for you:

```
emerge -a genkernel
```

Run to make a dmraid kernel: `genkernel --dmraid all`

or (if kernel/initrd compiling crashes)

```
genkernel --dmraid initrd genkernel kernel
```

If you have trouble with the initrd compilation, you can resolve it with:

```
emerge -a libselinux LIBS="-lselinux -lsepolicy" genkernel --dmraid initrd
```

Compromise Solution (using genkernel only for dmraid and initrd)

If you do not want to go the hard way but still want a custom kernel, then emerge genkernel and

execute it as follows.

```
<_yamdwe_nowiki>0</_yamdwe_nowiki>
```

or for just the initrd:

```
<_yamdwe_nowiki>1</_yamdwe_nowiki>
```

With this, the Kernel and initramfs will be autom. installed, be sure to use the correct file names later in the grub config. This way you can skip the rest of this section and go to editing grub config.

Hard Solution (using a self-built kernel)

The hard solution assumes that you are building your own kernel and not using genkernel. This solution has been tested on gentoo-sources 2.6.18-r3.

This is really the hard way... the versions of all the following components are changing over and over, download paths are broken, moved or temporarily down.

Gerte Hoogewerf created a script to create an image that will allow you to boot correctly. To start the process of creating the image, download the following files and mark the dmraidinitrd script as executable.

```
wget http://www.homelinux.nl/gen2dmraid/dmraidinitrd wget  
http://www.homelinux.nl/gen2dmraid/linuxrc chmod +x dmraidinitrd
```

Patched version of dmraidinitrd also available. MichaelDubner 19:40, 2 September 2006 (UTC) This has not been tested to work on <_yamdwe_nowiki>2</_yamdwe_nowiki>

Before running the script, you will have to make some changes to it. These changes will update the script as some of the download links are borked. It will also force a download of the correct version of these utilities, as there are compatibility issues with certain versions. In the segment

```
dmraidinitrd|<code> #changing this values might involve editing make_busybox()  
BUSYBOX_URI_PREFIX="http://www.busybox.net/downloads/"  
BUSYBOX_TARBALL="busybox-0.60.5.tar.bz2" BUSYBOX_DIR="busybox-0.60.5"</code> Change the  
values to the following: dmraidinitrd|<code>  
BUSYBOX_URI_PREFIX="http://www.busybox.net/downloads/legacy/" </code>
```

Note that using later versions of busybox will force you to use the patched script. They will also likely give you shell errors upon your first reboot.

```
In the segment dmraidinitrd|<code> #changing this values might involve editing  
make_devicemapper() DEVICEMAPPER_URI_PREFIX="ftp://sources.redhat.com/pub/dm/"  
DEVICEMAPPER_TARBALL="device-mapper.1.02.08.tgz" DEVICEMAPPER_DIR="device-  
mapper.1.02.08" </code>
```

```
Change the values to the following: dmraidinitrd|<code>  
DEVICEMAPPER_URI_PREFIX="ftp://sources.redhat.com/pub/dm/old/"  
DEVICEMAPPER_TARBALL="device-mapper.1.02.10.tgz" DEVICEMAPPER_DIR="device-  
mapper.1.02.10" </code>
```

In the segment `dmraidinitrd` `#changing this values might involve editing make_dmraid()`
`#please, don't use older versions of dmraid with this version of dmraidinitrd`
`DMRAID_URI_PREFIX="http://people.redhat.com/~heinzm/sw/dmraid/src/"`
`DMRAID_TARBALL="dmraid-1.0.0.rc8.tar.bz2" DMRAID_DIR="dmraid/1.0.0.rc8" </code>`

Change the values to the following: `dmraidinitrd`
`DMRAID_URI_PREFIX="http://people.redhat.com/~heinzm/sw/dmraid/src/"`
`DMRAID_TARBALL="dmraid-1.0.0.rc13.tar.bz2" DMRAID_DIR="dmraid/1.0.0.rc13" </code>`

Once you have the locations for the files done, go ahead and run the script. (Be sure that it has executable rights)

```
./dmraidinitrd linuxrc initrd
```

If the script fails for any reason, make sure the links are still active in the script. Sometimes the sites are rearranged (most notably: a file is moved into the /old/ folder). If you are building a SELinux box, make sure the correct libraries are installed:

```
emerge -a libselinux
```

If the script fails while complaining about dmraid compilation error then try adding „make \“ after the line „./configure ...“ in the „make_dmraid() { ... }“ segment.

This script will create a new ram drive image (initrd). Copy the initrd file and the linuxrc file to your /boot directory (or wherever the kernel images and grub reside)

```
cp linuxrc /boot cp initrd /boot
```

Editing grub.conf

the details at [HOWTO Gentoo Install on Bios \(Onboard\) RAID](#) are extremely useful.

Now that the ram image has been created, you need to let GRUB know how to work with it. Use the following segment from my grub.conf and make the adjustments to yours accordingly (this grub.conf assumes that your /boot is located on the first partition, and that your root is on your last partition).

```
/boot/grub.conf|<code> title Gentoo Linux root (hd0,0) kernel /boot/kernel-genkernel-x86-2.6.24-gentoo-r5 root=/dev/ram0 init=/linuxrc ramdisk=8192 real_root=/dev/mapper/nvidia_abiccada4 dodmraid video=vesafb:ywrap,pmipal,1280x1024-16@60 initrd /boot/initramfs-genkernel-x86-2.6.24-gentoo-r5 </code>
```

Note that the major changes are the root=/dev/ram0, the init=/linuxrc, and the real_root pointing to your root partition. Also note the initrd line at the bottom.

Installing GRUB

Boot-partition-is-first Method

This method of installing GRUB will work 99% of the time when your `/boot` partition is the first partition on the disk. If you have `/boot` on a partition other than the first, or if this method fails, proceed to Method 2. Do not run `grub-install`! It will not work properly, and will probably detect the `sda1`, etc. instead of your actual raid. Make also sure not to start grub inside of an chroot environment like you probably do if you install from the Gentoo LiveCD and do it like described in the installation manual of gentoo.org. It doesn't know about `/dev/mapper` so you would get some errors with grub. In this case just jump into another terminal with ALT+F2. Start grub and make sure that it's pointing to `/dev/null` as a device map.

- note* grub is not on the gentoo-am64 live cd, so ALT+F2'ing to another shell will NOT solve your problem of using GRUB outside of a chrooted environment. (grub is located at `/mnt/gentoo/sbin/grub`)

<_yamdwe_nowiki>3</_yamdwe_nowiki>

Now that you're inside of the grub environment, set up your raid devices and install grub!

```
grub> device (hd0,0) /dev/mapper/nvidia_abiccada1
grub> device (hd0) /dev/mapper/nvidia_abiccada
grub> root (hd0,0)
```

```
Filesystem is ext2fs, partition type 0x83
```

```
grub> setup (hd0)
```

```
Checking if "/boot/grub/stage1" exists... yes
Checking if "/boot/grub/stage2" exists... yes
Checking if "/boot/grub/e2fs_stage1_5" exists... yes
Running "embed /boot/grub/e2fs_stage1_5 (hd0)"... 16 sectors are embedded.
succeeded
Running "install /boot/grub/stage1 (hd0) (hd0)1+16 p
(hd0,0)/boot/grub/stage2 /boot/grub/menu.lst"...succeeded
Done.
```

```
grub> quit
```

Floppy Disk Method

If the previous method of installing GRUB failed, then you will need to create a GRUB boot disk and install GRUB natively. For this you will need a floppy disk. First you will need to install GRUB onto the floppy disk (note that this will erase all contents of the floppy disk!):

```
cd /boot/grub dd if=stage1 of=/dev/fd0 bs=512 count=1
```

```
1+0 records in
1+0 records out
```

<_yamdwe_nowiki>4</_yamdwe_nowiki>

```
153+1 records in
```

```
153+1 records out
```

Now reboot your computer and boot from your new GRUB boot disk. Once started, GRUB will show the command-line interface. First, set the GRUB's root device to the partition containing the boot directory:

```
grub> root (hd0,0)
```

If you are not sure which partition actually holds this directory, use the find command:

```
grub> find /boot/grub/stage1
```

This will search for the file name `/boot/grub/stage1` and show the devices which contain the file. Once you've set the root device correctly, run the setup command:

```
grub> setup (hd0)
```

```
Checking if "/boot/grub/stage1" exists... yes
Checking if "/boot/grub/stage2" exists... yes
Checking if "/boot/grub/e2fs_stage1_5" exists... yes
Running "embed /boot/grub/e2fs_stage1_5 (hd0)"... 16 sectors are embedded.
succeeded
Running "install /boot/grub/stage1 (hd0) (hd0)1+16 p
(hd0,0)/boot/grub/stage2 /boot/grub/menu.lst"...succeeded
Done.
```

You should be able to restart your computer now (or finish your installation and restart) and GRUB appear in its glory. Even more importantly, you should be able to boot into Linux and get a prompt.

Boot-partition-not-first Method

When your `/boot` partition is not the first partition, try following the steps. In some cases, you will have to specify „drive“ geometry. Run `fdisk` on the raid drive, and note C/H/S informations for later usage in grub.

```
<_yamdwe_nowiki>5</_yamdwe_nowiki>
```

We have to set the devices for grub manually, grub can't do this for us. We use the

```
--device-map
```

option and input nothing:

```
<_yamdwe_nowiki>6</_yamdwe_nowiki>
```

Now we are in Grub shell mode, we tell grub which „linux device“ becomes which „bios device“ on boot. If your boot partition is `/dev/mapper/your_raid_set1` then it becomes `(hd0,0)` on boot the last bios nr is the Linux partition nr -1.

First tell grub where your partitions are:

<__yamdwe_nowiki>7</__yamdwe_nowiki>

Partitions are identified in the order in which they are numbered on the partition table. In other words, if your boot partition is not labeled `your_raid_set1`, it will not be `(hd0,0)`.

If you have problems with your partitions in grub, tell grub what your drive geometry is now. But don't forget to set the devices first or grub doesn't know what you are talking about. **Replace C H S with cylinders, heads, and sectors.**

<__yamdwe_nowiki>8</__yamdwe_nowiki>

Now Grub knows where the boot partition is, and where to write the MBR. Next step is Grub install, type

```
root (hd0,
```

and press Tab. If you did nothing wrong grub should show you a list of possible partitions, if not you did something wrong.

```
grub> root (hd0,0) grub> setup (hd0) grub> quit
```

Acknowledgements

- [Gerte Hoogewerf](#), for creating the `initrd` script
- [aus9](#) of [Linux Questions Forums](#), for his constant and valuable help that made this possible
- [NeddySeagoon](#) of the [Gentoo Forums](#), for pointing me in the right direction.

From:
<https://www.deepdoc.at/dokuwiki/> - **DEEPDOC.AT - enjoy your brain**

Permanent link:
https://www.deepdoc.at/dokuwiki/doku.php?id=raidcontroller_unter_gentoo&rev=1306182925

Last update: 2025/11/29 22:06

