

Anaconda Fails When BIOS RAID Metadata Encountered

Finnbarr P. Murphy

(fpm@fpmurphy.com)

Anaconda (the RHEL, CentOS and Fedora installer) does not handle disks which were previously used in a BIOS or software RAID very well. Typically, it will refuse to install an OS on such disks. This is because it finds metadata relating to the RAID on the disk.

The term *BIOS RAID* (or *Fake RAID*) denote a system with a BIOS that is able to do basic RAID operations on an array of disks. Such a system has no actual RAID controller. Instead, an OS driver is required.

One metadata format commonly found on modern RAID systems (real as well as software-based) is that defined by the SNIA Common RAID Disk Data Format (DDF) specification. Both *mdadm* and *dmraid* support this format. This specification defines a standard data structure describing how data is formatted across the disks in a RAID group. This structure enables a certain level of interoperability between different RAID technology vendors.

From Section 2.2.1 of the specification:

The Anchor Header (see Section 5.5) for the DDF structure MUST be stored at the last logical block returned by either the ATA Identify Device or SCSI Read Capacity commands depending on the type of physical disk.

The DDF structure SHOULD be stored at the end of the physical disk next to the anchor header. Storing the DDF structure at the end of the physical disk allows the possibility of converting a single non-RAID physical disk to a RAID 1 configuration without shifting user data. Similarly, data on a member of a RAID 1 configuration with the DDF structure at the end can also be accessed using a non-RAID controller.

Another common metadata format is the Intel Matrix Storage Manager (IMSM) format. IMSM first appeared in the ICH6R (Southbridge) chip. As of 2014, Intel uses the term *Rapid Storage Technology* instead of *Matrix Storage Manager*. As with DDF, the metadata structure is stored at the end of the disk (search for the string "Intel Raid ISM Cfg Sig.")

The simplest solution is to reboot and add the *nodmraid* option to the kernel command line. Anaconda should then be able to use the disk.

If you wish to delete the metadata quickly, the quickest way is to zero out the last 512KB of data on the disk using the following command:

```
# dd if=/dev/zero of=$DEVICE bs=512 seek=$(( $(blockdev --getsz $DEVICE) - 1024 )) count=1024
```

Another way is to use the *dmraid* utility:

```
# dmraid -r -E $DEVICE
```

Replace \$DEVICE with the actual physical device, such as /dev/sda

For personal use only