

**NAME**

**mrsas** - LSI MegaRAID 6Gb/s and 12Gb/s SAS+SATA RAID controller driver

**SYNOPSIS**

To compile this driver into the kernel, place the following lines in your kernel configuration file:

```
device pci
device mrsas
```

Alternatively, to load the driver as a module at boot time, place the following line in loader.conf(5):

```
mrsas_load="YES"
```

**DESCRIPTION**

The **mrsas** driver will detect LSI's next generation (6Gb/s and 12Gb/s) PCI Express SAS/SATA RAID controllers. See the **HARDWARE** section for the supported devices list. A disk (virtual disk/physical disk) attached to the **mrsas** driver will be visible to the user through camcontrol(8) as `/dev/da?` device nodes. A simple management interface is also provided on a per-controller basis via the `/dev/mrsas?` device node.

The **mrsas** name is derived from the phrase "MegaRAID SAS HBA", which is substantially different than the old "MegaRAID" Driver mfi(4) which does not connect targets to the cam(4) layer and thus requires a new driver which attaches targets to the cam(4) layer. Older MegaRAID controllers are supported by mfi(4) and will not work with **mrsas**, but both the mfi(4) and **mrsas** drivers can detect and manage the LSI MegaRAID SAS 2208/2308/3008/3108 series of controllers.

The `device.hints(5)` option is provided to tune the **mrsas** driver's behavior for LSI MegaRAID SAS 2208/2308/3008/3108 controllers. By default, the mfi(4) driver will detect these controllers. See the **PRIORITY** section to know more about driver priority for MR-Fusion devices.

**mrsas** will provide a priority of (-30) (between `BUS_PROBE_DEFAULT` and `BUS_PROBE_LOW_PRIORITY`) at probe call for device id's 0x005B, 0x005D, and 0x005F so that **mrsas** does not take control of these devices without user intervention.

Solid-state drives (SSD) get ATA TRIM support with **mrsas** if underlying adapter allows it. This may require configuring SSD as Non-RAID drive rather than JBOD virtual mode.

**HARDWARE**

The **mrsas** driver supports the following hardware:

[ Thunderbolt 6Gb/s MR controller ]

- ⊕ LSI MegaRAID SAS 9265
- ⊕ LSI MegaRAID SAS 9266
- ⊕ LSI MegaRAID SAS 9267
- ⊕ LSI MegaRAID SAS 9270
- ⊕ LSI MegaRAID SAS 9271
- ⊕ LSI MegaRAID SAS 9272
- ⊕ LSI MegaRAID SAS 9285
- ⊕ LSI MegaRAID SAS 9286
- ⊕ DELL PERC H810
- ⊕ DELL PERC H710/P

[ Invader/Fury 12Gb/s MR controller ]

- ⊕ LSI MegaRAID SAS 9380
- ⊕ LSI MegaRAID SAS 9361
- ⊕ LSI MegaRAID SAS 9341
- ⊕ DELL PERC H830
- ⊕ DELL PERC H730/P
- ⊕ DELL PERC H330

## CONFIGURATION

To disable Online Controller Reset(OCR) for a specific **mrsas** driver instance, set the following tunable value in loader.conf(5):

```
dev.mrsas.X.disable_ocr=1
```

where X is the adapter number.

To change the I/O timeout value for a specific **mrsas** driver instance, set the following tunable value in loader.conf(5):

```
dev.mrsas.X.mrsas_io_timeout=NNNNNN
```

where NNNNNN is the timeout value in milli-seconds.

To change the firmware fault check timer value for a specific **mrsas** driver instance, set the following tunable value in loader.conf(5):

```
dev.mrsas.X.mrsas_fw_fault_check_delay=NN
```

where NN is the fault check delay value in seconds.

The current number of active I/O commands is shown in the *dev.mrsas.X.fw\_outstanding* sysctl(8) variable.

## DEBUGGING

To enable debugging prints from the **mrsas** driver, set the *hw.mrsas.X.debug\_level* variable, where X is the adapter number, either in loader.conf(5) or via sysctl(8). The following bits have the described effects:

- 0x01 Enable informational prints.
- 0x02 Enable tracing prints.
- 0x04 Enable prints for driver faults.
- 0x08 Enable prints for OCR and I/O timeout.
- 0x10 Enable prints for AEN events.

## PRIORITY

The **mrsas** driver will always set a default (-30) priority in the PCI subsystem for selection of MR-Fusion cards. (It is between BUS\_PROBE\_DEFAULT and BUS\_PROBE\_LOW\_PRIORITY). MR-Fusion Controllers include all cards with the Device IDs - 0x005B, 0x005D, 0x005F.

The mfi(4) driver will set a priority of either BUS\_PROBE\_DEFAULT or BUS\_PROBE\_LOW\_PRIORITY (depending on the device.hints setting) in the PCI subsystem for selection of MR-Fusion cards. With the above design in place, the mfi(4) driver will attach to a MR-Fusion card given that it has a higher priority than **mrsas**.

Using */boot/device.hints* (as mentioned below), the user can provide a preference for the **mrsas** driver to detect a MR-Fusion card instead of the mfi(4) driver.

```
hw.mfi.mrsas_enable="1"
```

At boot time, the mfi(4) driver will get priority to detect MR-Fusion controllers by default. Before changing this default driver selection policy, LSI advises users to understand how the driver selection policy works. LSI's policy is to provide priority to the mfi(4) driver to detect MR-Fusion cards, but allow for the ability to choose the **mrsas** driver to detect MR-Fusion cards.

LSI recommends setting `hw.mfi.mrsas_enable="0"` for customers who are using the older `mfi(4)` driver and do not want to switch to **mrsas**. For those customers who are using a MR-Fusion controller for the first time, LSI recommends using the **mrsas** driver and setting `hw.mfi.mrsas_enable="1"`.

Changing the default behavior is well tested under most conditions, but unexpected behavior may pop up if more complex and unrealistic operations are executed by switching between the `mfi(4)` and **mrsas** drivers for MR-Fusion. Switching drivers is designed to happen only one time. Although multiple switching is possible, it is not recommended. The user should decide from **Start of Day** which driver they want to use for the MR-Fusion card.

The user may see different device names when switching from `mfi(4)` to **mrsas**. This behavior is **Functions As Designed** and the user needs to change the `fstab(5)` entry manually if they are doing any experiments with `mfi(4)` and **mrsas** interoperability.

## FILES

`/dev/da?` array/logical disk interface

`/dev/mrsas?` management interface

## SEE ALSO

`cam(4)`, `mfi(4)`, `pci(4)`, `device.hints(5)`, `camcontrol(8)`

## HISTORY

The **mrsas** driver first appeared in FreeBSD 10.1.

**mfi Driver:** `mfi(4)` is the old FreeBSD driver which started with support for Gen-1 Controllers and was extended to support up to MR-Fusion (Device ID = 0x005B, 0x005D, 0x005F).

**mrsas Driver:** **mrsas** is the new driver reworked by LSI which supports Thunderbolt and onward products. The SAS+SATA RAID controller with device id 0x005b is referred to as the Thunderbolt controller throughout this man page.

**cam aware HBA drivers:** FreeBSD has a `cam(4)` layer which attaches storage devices and provides a common access mechanism to storage controllers and attached devices. The **mrsas** driver is `cam(4)` aware and devices associated with **mrsas** can be seen using `camcontrol(8)`. The `mfi(4)` driver does not understand the `cam(4)` layer and it directly associates storage disks to the block layer.

**Thunderbolt Controller:** This is the 6Gb/s MegaRAID HBA card which has device id 0x005B.

**Invader Controller:** This is 12Gb/s MegaRAID HBA card which has device id 0x005D.

**Fury Controller:** This is the 12Gb/s MegaRAID HBA card which has device id 0x005F.

## AUTHORS

The **mrsas** driver and this manual page were written by Kashyap Desai <*Kashyap.Desai@lsi.com*>.

## TODO

The driver does not support alias for device name (it is required when the user switches between two drivers and does not want to edit */etc/fstab* manually).

The **mrsas** driver exposes devices as */dev/da?*, whereas **mfi(4)** exposes devices as */dev/mfid?*.

**mrsas** does not support the Linux Emulator interface.

**mrsas** will not work with **mfiutil(8)**.