NAME

mfiutil, **mrsasutil -** Utility for managing LSI MegaRAID SAS controllers

SYNOPSIS

mfiutil version
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **show adapter**
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **show battery**
mfiutil [**-d**] [**-e**] [**-D** *device*] [**-t** *type*] [**-u** *unit*] **show config**
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **show drives**
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **show events** [**-c** *class*] [**-l** *locale*] [**-n** *count*] [**-v**] [*start* [*stop*]]
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **show firmware**
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **show foreign** [*volume*]
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **show logstate**
mfiutil [**-d**] [**-e**] [**-D** *device*] [**-t** *type*] [**-u** *unit*] **show patrol**
mfiutil [**-d**] [**-e**] [**-D** *device*] [**-t** *type*] [**-u** *unit*] **show progress**
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **show volumes**
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **fail** *drive*
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **good** *drive*
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **rebuild** *drive*
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **syspd** *drive*
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **drive progress** *drive*
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **drive clear** *drive* {start | stop}
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **start rebuild** *drive*
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **abort rebuild** *drive*
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **locate** *drive* {on | off}
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **cache** *volume* [*setting* [*value*] [...]]
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **name** *volume name*
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **volume progress** *volume*
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **clear**
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **create** *type* [**-v**] [**-s** *stripe_size*] *drive*[,*drive*[,...]] [*drive*[,*drive*[,...]]]
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **delete** *volume*
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **add** *drive* [*volume*]
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **remove** *drive*
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **start patrol**
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **stop patrol**
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **patrol** *command* [*interval* [*start*]]
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **foreign scan**
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **foreign clear** [*config*]
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **foreign diag** [*config*]
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **foreign preview** [*config*]

mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **foreign import** [*config*]
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **flash** *file*
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **start learn**
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **bbu** *setting value*
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **ctrlprop** *rebuild* [*rate*]
mfiutil [**-D** *device*] [**-t** *type*] [**-u** *unit*] **ctrlprop** *alarm* [*0/1*]

DESCRIPTION

The **mfiutil** utility can be used to display or modify various parameters on LSI MegaRAID SAS RAID controllers. Each invocation of **mfiutil** consists of zero or more global options followed by a command. Commands may support additional optional or required arguments after the command.

Currently three global options are supported:

-D *device*

device specifies the device node of the controller to use. If no device node is specified, then device will be made of the type and device.

-t *type*

type specifies the type of the controller to work with either mfi(4) or mrsas(4). If no type is specified, then the name of the invoked tool used to derive the type.

-u *unit*

unit specifies the unit of the controller to work with. If no unit is specified, then unit 0 is used.

Various commands accept either or both of the two options:

-d Print numeric device IDs as drive identifier. This is the default. Useful in combination with **-e** to print both, numeric device IDs and enclosure:slot information.

-e Print drive identifiers in enclosure:slot form. See next paragraph on format details in context of input rather than output.

Drives may be specified in two forms. First, a drive may be identified by its device ID. The device ID for configured drives can be found in **show config**. Second, a drive may be identified by its location as [E*xx*:]S*yy* where *xx* is the enclosure and *yy* is the slot for each drive as displayed in **show drives**.

Volumes may be specified in two forms. First, a volume may be identified by its target ID. Second, on the volume may be specified by the corresponding *mfidX* device, such as *mfid0*.

The **mfiutil** utility supports several different groups of commands. The first group of commands provide information about the controller, the volumes it manages, and the drives it controls. The second group of commands are used to manage the physical drives attached to the controller. The third group of commands are used to manage the logical volumes managed by the controller. The fourth group of commands are used to manage the drive configuration for the controller. The fifth group of commands are used to manage controller-wide operations.

The informational commands include:

version

Displays the version of **mfiutil**.

show adapter

Displays information about the RAID controller such as the model number.

show battery

Displays information about the battery from the battery backup unit.

show config

Displays the volume and drive configuration for the controller. Each array is listed along with the physical drives the array is built from. Each volume is listed along with the arrays that the volume spans. If any hot spare drives are configured, then they are listed as well.

show drives

Lists all of the physical drives attached to the controller.

show events [**-c** *class*] [**-l** *locale*] [**-n** *count*] [**-v**] [*start* [*stop*]]

Display entries from the controller’s event log. The controller maintains a circular buffer of events. Each event is tagged with a class and locale.

The *class* parameter limits the output to entries at the specified class or higher. The default class is "warn". The available classes from lowest priority to highest are:

debug Debug messages.

progress

Periodic progress updates for long-running operations such as background initializations, array rebuilds, or patrol reads.

info Informational messages such as drive insertions and volume creations.

warn Indicates that some component may be close to failing.

crit A component has failed, but no data is lost. For example, a volume becoming degraded due to a drive failure.

fatal A component has failed resulting in data loss.

dead The controller itself has died.

The *locale* parameter limits the output to entries for the specified part of the controller. The default locale is "all". The available locales are "volume", "drive", "enclosure", "battery", "sas", "controller", "config", "cluster", and "all".

The *count* parameter is a debugging aid that specifies the number of events to fetch from the controller for each low-level request. The default is 15 events.

By default, matching event log entries from the previous shutdown up to the present are displayed. This range can be adjusted via the *start* and *stop* parameters. Each of these parameters can either be specified as a log entry number or as one of the following aliases:

newest
The newest entry in the event log.

oldest The oldest entry in the event log.

clear The first entry since the event log was cleared.

shutdown
The entry in the event log corresponding to the last time the controller was cleanly shut down.

boot The entry in the event log corresponding to the most recent boot.

show firmware
Lists all of the firmware images present on the controller.

show foreign
Displays detected foreign configurations on disks for importation or removal.

show logstate

Display the various sequence numbers associated with the event log.

show patrol

Display the status of the controller's patrol read operation.

show progress

Report the current progress and estimated completion time for active operations on all volumes and drives.

show volumes

Lists all of the logical volumes managed by the controller.

The physical drive management commands include:

fail *drive*

Mark *drive* as failed. *Drive* must be an online drive that is part of an array.

good *drive*

Mark *drive* as an unconfigured good drive. *Drive* must not be part of an existing array.

rebuild *drive*

Mark a failed *drive* that is still part of an array as a good drive suitable for a rebuild. The firmware should kick off an array rebuild on its own if a failed drive is marked as a rebuild drive.

syspd *drive*

Present the drive to the host operating system as a disk SYSPD block device in the format /dev/mfisyspdX. Clear this flag with **good** *drive*

drive progress *drive*

Report the current progress and estimated completion time of drive operations such as rebuilds or patrol reads.

drive clear *drive* {start | stop}

Start or stop the writing of all 0x00 characters to a drive.

start rebuild *drive*

Manually start a rebuild on *drive*.

abort rebuild *drive*

Abort an in-progress rebuild operation on *drive*. It can be resumed with the **start rebuild** command.

locate *drive* {on | off}

Change the state of the external LED associated with *drive*.

The logical volume management commands include:

cache *volume* [*setting* [*value*] [...]]

If no *setting* arguments are supplied, then the current cache policy for *volume* is displayed; otherwise, the cache policy for *volume* is modified. One or more *setting* arguments may be given. Some settings take an additional *value* argument as noted below. The valid settings are:

enable

Enable caching for both read and write I/O operations.

disable

Disable caching for both read and write I/O operations.

reads Enable caching only for read I/O operations.

writes Enable caching only for write I/O operations.

write-back

Use write-back policy for cached writes.

write-through

Use write-through policy for cached writes.

read-ahead *value*

Set the read ahead policy for cached reads. The *value* argument can be set to either "none", "adaptive", or "always".

bad-bbu-write-cache *value*

Control the behavior of I/O write caching if the battery is dead or missing. The *value* argument can be set to either "disable" or "enable". In general this setting should be left disabled to avoid data loss when the system loses power.

write-cache *value*

Control the write caches on the physical drives backing *volume*. The *value* argument can be set to either "disable", "enable", or "default".

In general this setting should be left disabled to avoid data loss when the physical drives lose power. The battery backup of the RAID controller does not save data in the write caches of the physical drives.

name *volume name*

Sets the name of *volume* to *name*.

volume progress *volume*

Report the current progress and estimated completion time of volume operations such as consistency checks and initializations.

The configuration commands include:

clear Delete the entire configuration including all volumes, arrays, and spares.

create *type* [**-v**] [**-s** *stripe_size*] *drive*[,*drive*[,...]] [*drive*[,*drive*[,...]]]

Create a new volume. The *type* specifies the type of volume to create. Currently supported types include:

jbod Creates a RAID0 volume for each drive specified. Each drive must be specified as a separate argument.

raid0 Creates one RAID0 volume spanning the drives listed in the single drive list.

raid1 Creates one RAID1 volume spanning the drives listed in the single drive list.

raid5 Creates one RAID5 volume spanning the drives listed in the single drive list.

raid6 Creates one RAID6 volume spanning the drives listed in the single drive list.

raid10

Creates one RAID10 volume spanning multiple RAID1 arrays. The drives for each RAID1 array are specified as a single drive list.

raid50

Creates one RAID50 volume spanning multiple RAID5 arrays. The drives for each RAID5 array are specified as a single drive list.

raid60

Creates one RAID60 volume spanning multiple RAID6 arrays. The drives for each

RAID6 array are specified as a single drive list.

concat

Creates a single volume by concatenating all of the drives in the single drive list.

Note: Not all volume types are supported by all controllers.

If the **-v** flag is specified after *type*, then more verbose output will be enabled. Currently this just provides notification as drives are added to arrays and arrays to volumes when building the configuration.

The **-s** *stripe_size* parameter allows the stripe size of the array to be set. By default a stripe size of 64K is used. Valid values are 512 through 1M, though the MFI firmware may reject some values.

delete *volume*

Delete the volume *volume*.

add *drive* [*volume*]

Mark *drive* as a hot spare. *Drive* must be in the unconfigured good state. If *volume* is specified, then the hot spare will be dedicated to arrays backing that volume. Otherwise, *drive* will be used as a global hot spare backing all arrays for this controller. Note that *drive* must be as large as the smallest drive in all of the arrays it is going to back.

remove *drive*

Remove the hot spare *drive* from service. It will be placed in the unconfigured good state.

The controller management commands include:

patrol *command* [*interval* [*start*]]

Set the patrol read operation mode. The *command* argument can be one of the following values:

disable

Disable patrol reads.

auto Enable periodic patrol reads initiated by the firmware. The optional *interval* argument specifies the interval in seconds between patrol reads. If patrol reads should be run continuously, then *interval* should consist of the word "continuously". The optional *start* argument specifies a non-negative, relative start time for the next patrol read. If an interval or start time is not specified, then the existing setting will be used.

manual

Enable manual patrol reads that are only initiated by the user.

start patrol

Start a patrol read operation.

stop patrol

Stop a currently running patrol read operation.

foreign scan

Scan for foreign configurations and display the number found. The *config* argument for the commands below takes the form of a number from 0 to the total configurations found.

foreign clear [config]

Clear the specified foreign *config* or all if no *config* argument is provided.

foreign diag [config]

Display a diagnostic display of the specified foreign *config* or all if no *config* argument is provided.

foreign preview [config]

Preview the specified foreign *config* after import or all if no *config* argument is provided.

foreign import [config]

Import the specified foreign *config* or all if no *config* argument is provided.

flash *file*

Updates the flash on the controller with the firmware stored in *file*. A reboot is required for the new firmware to take effect.

start learn

Start a battery relearn. Note that this seems to always result in the battery being completely drained, regardless of the BBU mode. In particular, the controller write cache will be disabled during the relearn even if transparent learning mode is enabled.

bbu *setting value*

Update battery backup unit (BBU) properties related to battery relearning. The following settings are configurable:

learn-delay

Add a delay to the next scheduled battery relearn event. This setting is given in hours and must lie in the range of 0 to 255.

autolearn-mode

Enable or disable automatic periodic battery relearning. The setting may be set to "enable" or "disable" to respectively enable or disable the relearn cycle. Alternatively, a mode of 0, 1 or 2 may be given. Mode 0 enables periodic relearning, mode 1 disables it, and mode 2 disables it and logs a warning to the event log when it detects that a battery relearn should be performed.

bbu-mode

Set the BBU's mode of operation. This setting is not supported by all BBUs. Where it is supported, the possible values are the integers between 1 and 5 inclusive. Modes 1, 2 and 3 enable a transparent learn cycle, whereas modes 4 and 5 do not. The BBU's data retention time is greater when transparent learning is not used.

ctrlprop *rebuild* [*rate*]

With no arguments display the rate of rebuild (percentage)a for volumes. With an integer argument (0-100), set that value as the new rebuild rate for volumes.

ctrlprop *alarm* [*0/1*]

With no arguments display the current alarm enable/disable status. With a 0, disable alarms. With a 1, enable alarms.

EXAMPLES

Configure the cache for volume mfid0 to cache only writes:

```
mfiutil cache mfid0 writes
mfiutil cache mfid0 write-back
```

Create a RAID5 array spanning the first four disks in the second enclosure:

```
mfiutil create raid5 e1:s0,e1:s1,e1:s2,e1:s4
```

Configure the first three disks on a controller as JBOD:

```
mfiutil create jbod 0 1 2
```

Create a RAID10 volume that spans two arrays each of which contains two disks from two different enclosures:

```
mfiutil create raid10 e1:s0,e1:s1 e2:s0,e2:s1
```

Add drive with the device ID of 4 as a global hot spare:

```
mfiutil add 4
```

Add the drive in slot 2 in the main chassis as a hot spare for volume mfid0:

```
mfiutil add s2 mfid0
```

Reconfigure a disk as a SYSPD block device with no RAID

```
mfiutil syspd 0
```

Configure the adapter to run periodic patrol reads once a week with the first patrol read starting in 5 minutes:

```
mfiutil patrol auto 604800 300
```

Display the second detected foreign configuration:

```
mfiutil show foreign 1
```

Set the current rebuild rate for volumes to 40%:

```
mfiutil ctrlprop rebuild 40
```

SEE ALSO

mfi(4), mrsas(4)

HISTORY

The **mfiutil** utility first appeared in FreeBSD 8.0.