## NAME

**mptutil** - Utility for managing LSI Fusion-MPT controllers

## SYNOPSIS

**mptutil version**
**mptutil** [**-u** *unit*] **show adapter**
**mptutil** [**-u** *unit*] **show config**
**mptutil** [**-u** *unit*] **show drives**
**mptutil** [**-u** *unit*] **show events**
**mptutil** [**-u** *unit*] **show volumes**
**mptutil** [**-u** *unit*] **fail** *drive*
**mptutil** [**-u** *unit*] **online** *drive*
**mptutil** [**-u** *unit*] **offline** *drive*
**mptutil** [**-u** *unit*] **name** *volume name*
**mptutil** [**-u** *unit*] **volume status** *volume*
**mptutil** [**-u** *unit*] **volume cache** *volume enable|enabled|disable|disabled*
**mptutil** [**-u** *unit*] **clear**
**mptutil** [**-u** *unit*] **create** *type* [**-q**] [**-v**] [**-s** *stripe_size*] *drive*[,*drive*[,...]]
**mptutil** [**-u** *unit*] **delete** *volume*
**mptutil** [**-u** *unit*] **add** *drive* [*volume*]
**mptutil** [**-u** *unit*] **remove** *drive*

## DESCRIPTION

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

Currently one global option is supported:

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

Volumes may be specified in two forms. First, a volume may be identified by its location as [*xx*:]*yy* where *xx* is the bus ID and *yy* is the target ID. If the bus ID is omitted, the volume is assumed to be on bus 0. Second, on the volume may be specified by the corresponding *daX* device, such as *da0*.

The mpt(4) controller divides drives up into two categories. Configured drives belong to a RAID volume either as a member drive or as a hot spare. Each configured drive is assigned a unique device ID such as 0 or 1 that is show in **show config**, and in the first column of **show drives**. Any drive not associated with a RAID volume as either a member or a hot spare is a standalone drive. Standalone

drives are visible to the operating system as SCSI disk devices. As a result, drives may be specified in three forms. First, a configured drive may be identified by its device ID. Second, any drive may be identified by its location as *xx*:*yy* where *xx* is the bus ID and *yy* is the target ID for each drive as displayed in **show drives**. Note that unlike volumes, a drive location always requires the bus ID to avoid confusion with device IDs. Third, a standalone drive that is not part of a volume may be identified by its corresponding *daX* device as displayed in **show drives**.

The **mptutil** 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 informational commands include:

**version**
: Displays the version of **mptutil**.

**show adapter**
: Displays information about the RAID controller such as the model number.

**show config**
: Displays the volume and drive configuration for the controller. Each volume is listed along with the physical drives 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**
: Display all the entries from the controller's event log. Due to lack of documentation this command is not very useful currently and just dumps each log entry in hex.

**show volumes**
: Lists all of the logical volumes managed by the controller.

The physical drive management commands include:

**fail** *drive*
: Mark *drive* as "failed requested". Note that this state is different from the "failed" state that is

used when the firmware fails a drive. *Drive* must be a configured drive.

**online** *drive*
Mark *drive* as an online drive. *Drive* must be part a configured drive in either the "offline" or "failed requested" states.

**offline** *drive*
Mark *drive* as offline. *Drive* must be a configured, online drive.

The logical volume management commands include:

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

**volume cache** *volume enable|enabled|disable|disabled*
Enables or disables the drive write cache for the member drives of *volume*.

**volume status** *volume*
Display more detailed status about a single volume including the current progress of a rebuild operation if one is being performed.

The configuration commands include:

**clear** Delete the entire configuration including all volumes and spares. All drives will become standalone drives.

**create** *type* [**-q**] [**-v**] [**-s** *stripe_size*] *drive*[,*drive*[,...]]
Create a new volume. The *type* specifies the type of volume to create. Currently supported types include:

**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.

**raid1e** Creates one RAID1E volume spanning the drives listed in the single drive list.

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

If the **-q** flag is specified after *type*, then a "quick" initialization of the volume will be done. This is useful when the drives do not contain any existing data that need to be preserved.

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 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. The list of valid values for a given *type* are listed in the output of **show adapter**.

**delete** *volume*

Delete the volume *volume*. Member drives will become standalone drives.

**add** *drive* [*volume*]

Mark *drive* as a hot spare. *Drive* must not be a member of a volume. If *volume* is specified, then the hot spare will be dedicated to that volume. Otherwise, *drive* will be used as a global hot spare backing all volumes for this controller. Note that *drive* must be as large as the smallest drive in all of the volumes it is going to back.

**remove** *drive*

Remove the hot spare *drive* from service. It will become a standalone drive.

## EXAMPLES

Mark the drive at bus 0 target 4 as offline:

```
mptutil offline 0:4
```

Create a RAID1 array from the two standalone drives *da1* and *da2*:

```
mptutil create raid1 da1,da2
```

Mark standalone drive *da3* as a global hot spare:

```
mptutil add da3
```

## SEE ALSO

mpt(4)

## HISTORY

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

## BUGS

The handling of spare drives appears to be unreliable. The mpt(4) firmware manages spares via spare drive "pools". There are eight pools numbered 0 through 7. Each spare drive can only be assigned to a

single pool. Each volume can be backed by any combination of zero or more spare pools. The **mptutil** utility attempts to use the following algorithm for managing spares. Global spares are always assigned to pool 0, and all volumes are always backed by pool 0. For dedicated spares, **mptutil** assigns one of the remaining 7 pools to each volume and assigns dedicated drives to that pool. In practice however, it seems that assigning a drive as a spare does not take effect until the box has been rebooted. Also, the firmware renumbers the spare pool assignments after a reboot which undoes the effects of the algorithm above. Simple cases such as assigning global spares seem to work ok (albeit requiring a reboot to take effect) but more "exotic" configurations may not work reliably.

Drive configuration commands result in an excessive flood of messages on the console.

The mpt version 1 API that is used by **mptutil** and mpt(4) does not support volumes above two terabytes. This is a limitation of the API. If you are using this adapter with volumes larger than two terabytes, use the adapter in JBOD mode. Utilize geom(8), zfs(8), or another software volume manager to work around this limitation.