HighPoint RAID Management

Command Line Interface Guide

Revision: 1.0
Date: Oct. 2004

HighPoint Technologies, Inc.

Copyright

Trademarks

Disclaimer

Table of Contents

Introduction to the HighPoint Command Line Interface

The HighPoint CLI (Command Line Interface) is a non-graphical utility that configures and manages HighPoint RAID controllers via a command line. It is ideal for systems where an X Window system is not configured, and the RAID Management GUI cannot be used.

Supported Operation Systems

This release of the CLI utility is limited to FreeBSD operating systems.

Installing the HighPoint CLI on FreeBSD

Before installation, you must log on as root.

Two files are needed for installation:

> hptcliinstall.sh
> hptcli.tar.gz

Put the two files to some directory and run "**./hptcliinstall.sh -i**" to install the software. Running "hptcliinstall.sh" with no parameters will show help information about the installation program.

The following files will be installed/configured:

/usr/bin/hptsvr	- service executable
/etc/hptcfg	- service config file
/usr/local/etc/rc.d/hptdaemon.sh	- service control script
/usr/bin/hptraidconf	- the CLI executable

Before running the HighPoint CLI , you must have the proper driver loaded, and HighPoint RAID Management Service (**hptsvr**) run on the system. Check www.highpoint-tech.com for the latest software versions and update information.

Note:

You cannot run HighPoint RAID management GUI and CLI at same time on a system.

Running HighPoint CLI

You can run the HighPoint CLI software two ways:

- Entering commands at the CLI prompt

- Entering a string of single commands with screen output

Using the CLI prompt:

You can enter commands at the main prompt and view the output on-screen.

To use the HighPoint CLI prompt, follow these steps:

1   Enter the following command at the shell prompt:

   # hptraidconf

2   Enter the username and password to log on to the CLI.

   Note: The default username is **RAID** and the default password is **hpt**. You can change the password using '**user**' command.

   After successfully logging in, the main prompt of the CLI will be displayed:

   HighPoint CLI>

   Commands are entered at the CLI prompt. The **CLI Command Reference** section discusses these commands in detail (see page 4).

Using a single command with output

You can enter a complete command with parameters at the shell prompt and receive one output result on the screen at a time.

Syntax

   hptraidconf {command} [options] {username} {password}

Example

   # hptraidconf query controllers RAID hpt

   Will display information of the detected controllers.

Note:

In shell command line mode, you need to specify the username and password in addition to the normal command in interactive mode. For example, if you want to query all the information of the physical devices, in the command line mode, you need to enter the complete query command at the shell prompt (assume the default user name **RAID** and default password **hpt** is used):

   # hptraidconf query devices RAID hpt

While in interactive mode, you need only enter the following command at the

CLI prompt. The last two parameters, user name and password have been omitted:

HighPoint CLI> query devices

Conventions

The following conventions are used through this guide:

- **Bold font** is used for what you type on the command line and for the screen output.

- In commands, braces {} around an item indicate that it must be specified.

- In commands, square brackets [] around an item indicate that it is optional.

- In commands, braces with pipes inside {| | | |} indicate you must specify one option among multiple items.

- In commands, square brackets with pipes inside [| | | |] indicate you can either specify one option or more options.

- Whenever you type a parameter with a space, it should be enclosed with double quotation marks "".

CLI Command Reference

This chapter discusses the various HighPoint CLI commands: Query, Create, Delete, Rename, Rebuild, Device, Verify, Task, Rescan, Events, Reboot, User, Mail, Help and Exit.

Warning:

Operations under the Create/Delete commands may destroy data stored in the disks, and the lost data can never be recovered. So you should take special care when executing these commands. The CLI utility will not prompt before each operation is committed.

Query Commands

Syntax

- query controllers
- query devices
- query devices {device_id}
- query arrays
- query arrays {array_id}

query controllers

This command reports controller information.

Example

HighPoint CLI> query controllers

Typical output:

```
ID   Channels    Name
----------------------------------------------------
1    8           RocketRAID 182x SATA Controller
```

The output indicates that **controller 1** is a **RocketRAID 182x SATA Controller** with **8 channels**, which supports a maximum of 8 SATA hard disks.

query devices

This command presents status of all the physical devices to the controllers. It provides a list of device ID, capacity, model number, status and array attributes. Each device's status will be listed as one of the following: NORMAL, DISABLED, SPARE, and BOOT. The BOOT status means the disk has been set to boot disk in the controller BIOS.

Note:

A device ID is a string used to represent a disk. It is in the format "controller/port" for SATA controllers, and "controller/channel/device" for PATA controllers. e.g. 1/2 represents the disk on controller 1 port 2; 1/2/1 represents the disk on controller 1 channel 2 master disk; 1/2/2 represents the disk on controller 1 channel 2 slave disk.

Example

HighPoint CLI> query devices

Typical output:

```
ID    Capacity(GB)  Model Number          Status    Array

-------------------------------------------------------------

1/2   250.000       WDC WD2500JD-00GBB0   SPARE     Single

1/3    80.000       Maxtor 6Y080M0        NORMAL    myraid1

1/5   250.000       WDC WD2500JD-00GBB0   NORMAL    myraid1

1/8    50.000       Maxtor 7B250S0        NORMAL    Single
```

query devices {device_id}

This command presents information for the specified device.

Example

HighPoint CLI> query devices 1/2

Typical output:

```
Model Number: WDC WD2500JD-00GBB0

Capacity: 250.000GB

Status: NORMAL

Array: myraid1

Write Cache: Enabled

Read Ahead: Enabled
```

query arrays

This command lists information for all configured arrays. It will list each array's ID, capacity, RAID level, and status information.

Note: An array ID is generally represented by number or set of numbers. For RAID1/0 arrays; each sub-array will be represented by an ID in the format "1-1", "1-2". An array ID is used in the command line where an array needs to be specified.

Example

HighPoint CLI> query arrays

Typical output:

```
ID      Capacity(GB)  Type      Status     BS (K)  Progress  Name

-------------------------------------------------------------------

1       80.0          RAID0     BOOT       64      --        myraid

2       248.83        RAID5     NORMAL     64      --        data

1/8     120.02        Spare     NORMAL     --      --        --
```

query arrays {array_id}

This command will present information of each disk of a specified array.

Example

HighPoint CLI> query arrays 1

Typical output:

```
ID    Capacity(GB)    Type      Model Number

-----------------------------------------------------------

1/1   250.0           RAID5     WDC WD2500JD-00GBB1

1/2   250.0           RAID5     WDC WD2500JD-00GBB2

1/3   250.0           RAID5     Maxtor 7B250S222
```

Create Command

This command allows you to create a new RAID array or add a spare disk.

Syntax

create {RAID0|RAID1|RAID10|RAID5|JBOD|spare} [create-options]

Parameters

You can specify one or more create options for this command, separated by space. The options can be typed in any order.

disks=   specifies member disks which will compose a new array, e.g. disks=1/1, 1/2, disks=*. The character * means all available drives on the controller when you have only one supported controller installed.

bs=   specifies block size in KB, the default value is 64K. This parameter is used when creating a RAID 0, 5 or 10 array. The available values may vary among different controllers and driver versions (typically 16,32,64,128,256,1024,2048).

init=   specifies initialization option (create-only, zero-build, scrub, duplicate). The default option is create-only. The create-only option is applicable for all the RAID types, which is to create an array without any initialization process. The zero-build option requires every stripe in a RAID 5 array to be zeroed out. The scrub option will create a RAID 5 array and initialize parity using the existing data on the hard disks. The duplicate option is available for creating a RAID1 array to copy all data from the first member disk to the second one.

name=   specifies the name for the array being created.

           If the option is omitted, the utility will assign a default name for the array.

Examples

■   **HighPoint CLI> create RAID0 name=myraid0 disks=1/3,1/4**

This command instructs the system to create an RAID 0 array using the disks attached to controller 1 channels 3 and 4, and name it **myraid0**.

■   **HighPoint CLI> create RAID5 name=myraid5 disks=* bs=64**

This command instructs the controller to use all the available disks (a minimum three disks required) to create a **RAID5** array named **myraid5**. The specified block size is **64K**.

■   **HighPoint CLI> create spare disks=1/4**

This command indicates that you will set the disk on controller 1 channel 4 to be a spare disk.

Delete Command

This command allows you to delete an existing RAID array or remove a spare disk. After deletion, the original array and all data on it will be lost. All the member disks will be listed as available single disks.

Syntax

delete {array_or_spare_ID}

Examples

■ **HighPoint CLI> delete 1/3**

This command indicates to remove the spare disk on controller 1 channel 3.

■ **HighPoint CLI> delete 1**

This command indicates to delete the array whose id is "**1**". You can query the array ID before the deletion.

Rename Command

This command allows you to rename the specified array and type a new name for the array.

Syntax

rename {array_id} {new_name}

Example

HighPoint CLI> rename 1 hptraid

This command instructs the controller to rename the array "**1**" to **hptraid**.

Rebuild Commands

You can use rebuild commands to rebuild a RAID1, RAID1/0 or RAID5 array when it is critical or broken.

Syntax

■ rebuild {array_id} {device_id}

■ rebuild {array_id} {start|stop}

rebuild {array_id} {device_id}

This command allows you to add the specified disk to a broken array and rebuild it.

Example

HighPoint CLI> rebuild 1 1/3

This command instructs the controller to add the disk "1/3" to rebuild the array "1". You can use the query commands first to verify the device ID and the array ID information before the rebuild command.

rebuild {array_id} {start|stop}

This command allows you to start or stop the rebuilding process on the specified array. After you stopped a rebuilding process, you can resume it at a later time by the rebuild start command.

Examples

- **HighPoint CLI> rebuild 1 start**

This command starts the rebuilding process on the array "1".

- **HighPoint CLI> rebuild 1 stop**

This command stops the rebuilding process on the array "1".

Verify Command

Syntax

- verify {array_id} {start|stop}

This command starts or stops the verify process on the specified array.

Examples

- **HighPoint CLI> verify 1 start**

This command starts to verify the array "1".

- **HighPoint CLI> verify 1 stop**

This command stops the verify process on the array "1".

Init Command

Syntax

- init {array_id} stop

This command stops initialization process of an array being created. This command only applies to a zero-build RAID5 array. The array will be destroyed after this command.

Examples

- **HighPoint CLI> init 1 stop**

This command stops the initialization of RAID5 array "**1**".

Device Command

This command enables or disables the optional device features. This command is only available when the controller driver supports setting device parameters and the selected disk supports the specified feature.

Syntax

device {device_id} {e|d} {wc|ra|tcq|ncq}

Parameters

e	enable a feature
d	disable a feature
wc	write cache
ra	read look-ahead
tcq	Tagged Command Queuing (TCQ)
ncq	Native Command Queuing (NCQ)

Examples

- **HighPoint CLI> device 1/4 e wc**

This command will enable write cache on the disk on controller 1 channel 4.

- **HighPoint CLI> device 1/5/1 d wc**

This command will disable write cache on the disk on controller 1 channel 5.

Rescan Command

This command will rescan all of the physical devices attached to the RAID controller.

Syntax

rescan

Example

HighPoint CLI> rescan

Events Commands

The CLI system will automatically record three types of events: Information (shortened to "Inf"), Warning (shortened to "War"), and Error (shortened to "Err") on the screen

output. These commands allow you to query, save, or clear the logged events.

Syntax

- events

- events clear

- events save {file_name}

events

This command will display a list of all the logged events.

Example

HighPoint CLI> events

Typical output:

```
ID Type        Time                    Description
-------------------------------------------------------------------------------
1  Inf  [10/12/2004 15:26:21]   User RAID (from 127.0.0.1) exited from system.
2  Inf  [10/12/2004 15:26:30]   User RAID (from 127.0.0.1) logged on system.
3  Inf  [10/12/2004 15:26:34]   Array 'myraid' verifying started.
4  War  [10/12/2004 15:31:35]   Array 'myraid' verifying aborted.
5  Inf  [10/12/2004 15:31:36]   User RAID (from 127.0.0.1) exited from system.
6  Inf  [10/12/2004 15:32:15]   User RAID (from 127.0.0.1) logged on system.
7  Inf  [10/12/2004 15:32:51]   User RAID (from 127.0.0.1) exited from system.
8  Inf  [10/12/2004 15:37:00]   User RAID (from 127.0.0.1) logged on system.
```

events clear

This command will clear all the logged events.

Example

HighPoint CLI> events clear

events save {file_name}

This command will save all the logged events as a plain text file.

Example

HighPoint CLI> events save /tmp/raidlog.txt

This command will save all the events to /tmp/**raidlog.txt**.

Mail Commands

Syntax

- ■ mail recipient

- ■ mail recipient add {recipient_name}{mail_address} [Inf|War|Err]

- ■ mail recipient delete {recipient_name}

- ■ mail recipient test {recipient_name}

- ■ mail recipient set {recipient_name}{Inf|War|Err}

- ■ mail get server

- ■ mail set server {server_address}{port} {status}

- ■ mail set server {a|p|s|m} {value}

mail recipient

--- List all the mail recipients

Example

HighPoint CLI> mail recipient

Typical output:

```
ID  Name       Mail Address              Notify Types
-----------------------------------------------------------------
1   RAID       admin@somecompany.com     Information, Warning, Error.
2   Kate       kate@somecompany.com      Warning, Error.
```

mail recipient add {recipient_name} {mail_address} [Inf|War|Err]

--- Add a not exist recipient

Example

HighPoint CLI> mail recipient add admin admin@somecompany.com Inf War Err

This command will setup the RAID system to send mail to
admin@somecompany.com about all logged events.

mail recipient delete {recipient_name}

--- Delete an exist recipient.

Example

> **HighPoint CLI> mail recipient delete Ferry**

mail recipient test {recipient_name}

--- Send a testing mail to specified recipient.

Example

> **HighPoint CLI> mail recipient test RAID**

mail recipient set {recipient_name} {Inf|War|Err}

--- Set the notification type for a recipient.

Example

> **HighPoint CLI> mail recipient set admin War Err**

mail get server

--- Print the SMTP server information

Example

> **HighPoint CLI> mail get server**

Typical output:

```
Server Address        Port   Status   Mail from
-------------------------------------------------------------
mail.somecompany.com  25     disabled admin@somecompany.com
```

mail set server {server_address} {port} {status} {from_address}

--- Use this command to configure mail server settings.

{server_address}	– SMTP server address
{port}	– port, generally 25
{status}	– status, 'e' for enable or 'd' for disable
{from_address}	– mail from address

Examples:

> **HighPoint CLI> mail set server mail.somecompany.com 25 e admin@somecompany.com**

HighPoint CLI> mail set server mail.somecompany.com 25 d admin@somecompany.com

mail set server {a|p|s|m} {value}

--- Use this to separate set your mail server value

Parameters

a – SMTP server address

p – port, generally 25

s – status, 'e' for enable or 'd' for disable

m – mail from address

Examples:

HighPoint CLI> mail set server a smtp.somecompany.com

--- Change the server address

HighPoint CLI> mail set server p 25

--- Change the port

HighPoint CLI> mail set server s d

--- Disable mail notification

HighPoint CLI> mail set server s e

--- Enable mail notification

Task Commands

When an array requires that it be regularly verified or rebuilt, you can use the task commands to automatically execute this process in the background. As long as you have the appropriate privileges, you can add new tasks, and modify or delete existing tasks.

Syntax

- ■ task

- ■ task rebuild {array_id}{name=} {once|daily|monthly|weekly} {day} {interval}time=mm/dd/yyyy, mm/dd/yyyy, hh:mm:ss

- ■ task verify {array_id}{name=}{once|daily|monthly|weekly} {day} {interval}time=mm/dd/yyyy, mm/dd/yyyy, hh:mm:ss{login_user_name}

- ■ task delete {task_id}

- ■ task enable {task_id}

■ task disable {array_id}

task

This command displays detailed information about all scheduled tasks.

Example

HighPoint CLI> task

This command displays the current background tasks.

```
ID Task Description                      Start-Date End-Date    Time      S
------------------------------------------------------------------------
1 Rebuild Array_1(created by RAID) 10/08/2004 --/--/---- 10:20:30 E
```

task rebuild {array_id} {once|daily|weekly|monthly} {day} {interval}

time=mm/dd/yyyy, mm/dd/yyyy, hh:mm:ss

This command allows you to schedule the frequency by once, daily or monthly and the detailed time range to rebuild a specified array. The first **mm/dd/yyyy** specifies the task start date, while the second **mm/dd/yyyy** specifies the task end date.

Note:

When you add a task to rebuild a selected array **once**, the parameter **{day}** should be omitted.

Examples

■ **HighPoint CLI> task rebuild 1 once time=10/8/2005, 12:35:46**

This command adds a task schedule named **myraid1** to rebuild the array"**1**"at **12:35:46** on **10/8/2005**. The rebuild frequency is set to **once**.

■ **HighPoint CLI> task rebuild 4 daily 2 time=2/8/2005, 2/22/2005, 13:49:58**

This command adds a task schedule named **myraid4** to rebuild the array "**4**" at **13:49:58** every **2** days from **2/8/2005** to **2/22/2005**.

■ **HighPoint CLI> task rebuild 3 weekly 2 3 time=2/8/2004, 2/22/2008, 13:49:58 RAID" "**

This command adds a task schedule named **myraid3** to rebuild the array "**3**" at **13:49:58** on **Monday** (the **2nd** day in a week) every **3** weeks from **2/8/2004** to **2/22/2008**.

■ **HighPoint CLI> task rebuild 2 monthly 3 4 time=2/8/2004, 2/8/2006, 12:30:33**

This command adds a task schedule to rebuild the array "**2**" at **12:30:33** on

the **3rd** day of a month every **4** months from **2/8/2004** to **2/8/2006**.

task verify {array_id} {name=} {once|daily|weekly|monthly} {day} {interval} time=mm/dd/yyyy, mm/dd/yyyy, hh:mm:ss

This command allows you to schedule a verify task. The usage of this command is the same as adding a rebuild task schedule.

task delete {task_id}

This command allows you to delete a scheduled task. You can query the task ID by **task** command.

Example

HighPoint CLI> task delete 2

This command will delete the task **"2"**.

task enable {task_id}

This command will enable a disabled task.

Example

HighPoint CLI> task enable 1

This command will enable the disabled task **"1"**.

task disable {task_id}

This command will disable a scheduled task manually.

Example

HighPoint CLI> task disable 1

This command will disable the scheduled task **"1"**.

User Commands

Syntax

- user

- user add {name} {password}

- ■ user delete {name}

- ■ user set password {name} {password}

- ■ user set privilege {name} {privileges}

- ■ user unset privilege {name} {privileges}

Parameters

Privileges can be one or more of the following, separated by space:

MA	manage arrays
CA	create arrays
DA	delete arrays
RA	rename arrays
RB	rebuild arrays
VA	verify arrays
IA	initialize arrays
DU	duplicate arrays
MS	manage spare disk
RS	reboot server
MD	device management
MU	manage user
SM	set mail server
SS	set server system
CE	clear events
TM	task management

user

This command will list all the valid users with their management privileges and descriptions.

Note:

The default user **RAID** owns complete privileges and can assign different privileges to other users.

Example

HighPoint CLI> user

Typical output:

```
User Name MA CA DA RA RB VA IA DU MS MU SM RS CE MD SS TM   Description
-------------------------------------------------------------------------
Hello2     MA CA DA RA RB VA IA DU -- -- -- RS CE MD SS TM  Created by RAID
RAID       MA CA DA RA RB -- IA DU MS MU SM RS CE MD SS TM Automatically created
                                                            user for management
```

user add {name} {password}

This command allows you to add a new user.

Example

HighPoint CLI> user add John 000000

This command will add a new user named **John**. The password of this user is **000000**.

user delete {name}

This command will delete an existing user. Note the current user logged on cannot be deleted.

Example

HighPoint CLI> user delete John

This command will delete the user named **John**.

user set password {name} {password}

This command will change a user's password.

Example

HighPoint CLI> user set password Jerry 000000

This command will change **Jerry**'s password to **000000**.

user set privilege {name} {privileges}

This command will set privileges for a specified user.

Example

HighPoint CLI> user set privilege Jerry MA MS

This command will let **Jerry** have privileges to manage arrays and spare disks.

user unset privilege {name} {privileges}

This command will unset privileges for a specified user.

Help Commands

Syntax

- help

- help {command}

help

-- Show generic help about this utility.

Example

HighPoint CLI> help

help {command}

-- Show help about a specific command.

Example

HighPoint CLI> help create

Exit Command

Syntax

exit

-- Exit from the interactive mode and return to the shell.