Date	Version	Edit	HPT_U	MCM	FAE	Remake
2022/6/30	v1.12	ZY	CB (22/07/19)	CSL	YT	Add OCE/ORLM operating instruction

HighPoint RAID Management

Command Line Interface Guide

Revision: V1.12
Date: June 2022
HighPoint Technologies, Inc.

Copyright

Trademarks

Disclaimer

Table of Contents

Conventions

The following conventions are used through this guide:

■ Bold font is used for what you type on the command line and for thescreen 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 "".

Introduction to the HighPoint Command Line Interface

The HighPoint CLI (Command Line Interface) is a command line utility that configures and manages HighPoint RAID controllers via a command line. It is ideal for systems where the browser-based RAID Management utility cannot be used.

Supported Operation Systems and Adapters

The current CLI release supports Windows and Linux operating systems.

CLI Command Reference

This chapter discusses the various HighPoint CLI commands: Query, Create, Delete, OCE/ORLM, Rebuild, Verify, Unplug, Switch, Lscard, Rescan, Init, Events, Mail, Task, Set, Clear, Help, Exit and Diag.

Using Create/Delete commands may destroy data stored in the disks, and this lost data can never be recovered. Please be cautious when executing these commands. The CLI utility will not prompt you before each command is executed.

Help Commands

If you input an unknown or error command, you will be told that the command is unknown, you can use help commands to find correct commands.

```
HPT CLI > raid
ERROR: Unknown command raid .
You can input 'help' for more commands

HPT CLI >
```

Syntax

- help

- help {command}

help

- Show generic help about this utility.

Example

HPT CLI> help

```
HPT CLI > help
help [query|create|delete|OCE/ORLM|rebuild|verify|unplug|switch|lscard
rescan|init|events|mail|task|set|clear|help|exit|diag]
```

help {command}

- Show help about a specific command.

Example

HPT CLU> help create

```
HPT CLI > help create
Create Command
        This command allows you to create a new RAID array or add a spare disk.
Syntax:
        create {RAID0|RAID1|RAID3|RAID5|RAID6|RAID10|RAID50|JBOD|spare} [create-options]

create-option:

        disks=1/2,1/3... or disks=*
                Specify the disks used to create array.
        name=array name
                Specify the name of the array which will be created.
        src=source array ID
                If src argument is specified, OCE/ORLM will be started.
        cp=WB, WT or NONE
                Cache Policy option (WB: write back, WT: write through).
        init={foreground|background|keepdata|quickinit}
                Specifies array initialization option.
                foreground:
                        Zero out all data on the array. The array is not
                        accessible by the operating system until initialization is completed
                background:
                        Allow instant access to the array. Parity blocks
                        will be generated in background.
                keepdata:
                        Setup array information blocks on the drives only.
                        Use this option for array recovery.
                quickinit:
                        Setup array information blocks and zero out MBR data on the array.
        capacity=array capacity
                Specify the capacity (xxM,xxG) of the target array.
        matrix=n*m
                When create RAID50 to specify the matrix options.
                n : number of subarray's disk, m: number of subarray.
                For example: When create a RAID50 the option matrix
                can be matrix=3*2. That means 2 RAID5s each with 3 disks to form a RAID50
        bs=size
                Specify the block size (16k,32k,64k,128k,256k,512k,1024k)
        sector=size
                Specify the sector size (512B,1k,2k,4k)
```

Query Commands

Syntax

- query controllers

- query controllers {controller_id}

- query enclosures

- query enclosures {enclosure_id}

- query devices

- query devices {device_id}

- query arrays

- query arrays{array_id}

query controllers

This command reports controller information.

Example

HPT CLI> query controllers

For example SSD7140A

```
HPT CLI > query controllers
ID                              Channel         Name
1                               8               HighPoint NVMe RAID Controller
------------------------------------------------------------------------
```

For example RR3742

```
HPT CLI>query controllers

ID                              Channel        Name
1                               16             RocketRAID 3742 SAS Controller
-----------------------------------------------------------------------------

HPT CLI>
```

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, RAID and BOOT.

Attributes

ID：A device ID is a string used to represent a disk. It is in the format "controller/channel/device" for NVMe RAID HBAs. E.g. 1/E1/1 represents the disk on controller 1 port 1; It is in the format "controller/device" for SAS/SATA RAID HBAs. E.g. 1/1 represents the disk on controller 1 port 1.

Capacity:

The capacity of the disk in GB.

MaxFree:

The Maximum sequence free space on a disk which can be used by creating array.

Flag:

Shows whether the disk is **single** or has been created **RAID**.

Status:

NORMAL: The disk's status is normal.

DISABLED: The disk's cannot be use. (May be related to disk failure or

removed)

RAID: The disk is a member of a RAID array.

SPARE: The disk has been set as a spare disk

ModelNumber:

The disk's model number.

Example

HPT CLI> query devices
NVMe RAID HBAs
Single card:

```
HPT CLI > query devices
ID      Capacity   MaxFree    Flag    Status   ModelNumber
-----------------------------------------------------------------------
1/E1/1  500.03     0          RAID    NORMAL   Samsung SSD 970 EVO Plus 500GB
1/E1/2  500.03     0          RAID    NORMAL   Samsung SSD 970 EVO Plus 500GB
```

Cross-Sync:

```
HPT CLI > query devices
ID      Capacity   MaxFree    Flag     Status    ModelNumber
-------------------------------------------------------------------
1/E1/1  512.11     0          SINGLE   LEGACY    Samsung SSD 970 PRO 512GB
1/E1/2  512.11     0          SINGLE   LEGACY    Samsung SSD 970 PRO 512GB
1/E1/3  512.11     0          SINGLE   LEGACY    Samsung SSD 970 PRO 512GB
1/E1/4  512.11     0          SINGLE   LEGACY    Samsung SSD 970 PRO 512GB
1/E2/1  1000.20    0          SINGLE   LEGACY    WDS100T3X0C-00SJG0
1/E2/2  1000.20    0          SINGLE   LEGACY    WDS100T3X0C-00SJG0
1/E2/3  1000.20    0          SINGLE   LEGACY    WDS100T3X0C-00SJG0
1/E2/4  1000.20    0          SINGLE   LEGACY    WDS100T3X0C-00SJG0
```

SAS/SATA RAID HBAs

```
HPT CLI>query devices

ID      Capacity   MaxFree    Flag     Status    ModelNumber
-------------------------------------------------------------------
1/9     18000.21   0          SINGLE   LEGACY    WDC WUH721818ALE6L4
1/10    18000.21   0          SINGLE   LEGACY    WDC WUH721818ALE6L4
1/11    18000.21   0          SINGLE   LEGACY    WDC WUH721818ALE6L4
1/12    18000.21   0          SINGLE   LEGACY    WDC WUH721818ALE6L4
1/13    18000.21   0          SINGLE   LEGACY    WDC WUH721818ALE6L4
1/14    18000.21   0          SINGLE   LEGACY    WDC WUH721818ALE6L4
1/15    18000.21   0          SINGLE   LEGACY    WDC WUH721818ALE6L4
1/16    18000.21   0          SINGLE   LEGACY    WDC WUH721818ALE6L4
-------------------------------------------------------------------
```

query enclosures

This command reports Product ID information.
(Only support SSD7000 series)

Example

HPT CLI> query enclosures

Single card：
SSD7101A-1:

```
HPT CLI > query enclosures
ID   VendorID        ProductID         NumberOfPYH
----------------------------------------------------
1    HPT             SSD7101A-1        4
```

Cross-Sync：
Note: This function is only supported by SSD7101A-1, SSD7105, SSD7104, SSD7120
SSD7202 and SSD7505

SSD7101A-1:

```
HPT CLI > query enclosures
ID   VendorID        ProductID         NumberOfPYH
----------------------------------------------------
1    HPT             SSD7101A-1        4
2    HPT             SSD7101A-1        4
```

query devices {device_id}

This command presents information for the specified device.

Attributes

Serial Number:

Disk Serial Number

Firmware Version:

The disk's Firmware version.

Capacity：

The disk's capacity.

Status：

The disk's status.

Mode Number:

The disk's model number.

Read Ahead/Write Cache/TCQ/NCQ Status:

Disk's Read Ahead/Write Cache/TCQ/NCQ status could be enabled/disabled/--(NVMe not support)

Pcie width：

The disk's PCIE width.

Temperature：

The disk's temperature and setting temperature threshold.

S.M.A.R.T Attributes:

S.M.A.R.T Attributes detailed information reported by hard disk

Example

HPT CLI> query devices 1/E1/1

for example NVMe disk

```
HPT CLI > query devices 1/E1/1
Mode Number:        Samsung SSD 970 EVO Plus 500GB
Serial Number:      S4EVNMFN502918J
Firmware Version:   2B2QEXM7
Capacity(GB):   500.03          TotalFree(GB):  500.03
Status:         SINGLE          Flag:           NORMAL
PCIe Width:     x4              PCIe Speed:     Gen 3
Temperature (C):                            47
Warning Composite Temperature Threshold (C):    85
Critical Composite Temperature Threshold (C):   85
------------------------------------------------------------------
                      S.M.A.R.T Attributes
S.M.A.R.T Status OK.
Name                                    Value
------------------------------------------------------------------
Critical Warning                        : 0x0
Composite Temperature (C)               : 47
Available Spare                         : 100%
Available Spare Threshold               : 10%
Percentage Used                         : 7%
Data Units Read                         : 0x7da5bdd
Data Units Written                      : 0x6b05bb1
Host Read Commands                      : 0x8cb661dc
Host Write Commands                     : 0x6a64a263
Controller Busy Time                    : 0x61f
Power Cycles                            : 0xd8c
Power On Hours                          : 0x1cb
Unsafe Shutdowns                        : 0xa6f
Media and Data Integrity Errors         : 0x0
Number of Error Information Log Entries  : 0x9d5
Warning Temperature Time                : 0x0
Critical Composite Temperature Time     : 0x0
Temperature Sensor 1 (C)                : 47
Temperature Sensor 2 (C)                : 56
Temperature Sensor 3 (C)                : 0
Temperature Sensor 4 (C)                : 0
Temperature Sensor 5 (C)                : 0
Temperature Sensor 6 (C)                : 0
Temperature Sensor 7 (C)                : 0
Temperature Sensor 8 (C)                : 0
------------------------------------------------------------------
```

HPT CLI> query devices 1/1
for example SATA disk

```
HPT CLI>query devices 1/9

Mode Number:        WDC WUH721818ALE6L4
Serial Number:      3WJN6AWL
Firmware Version:   PCGNW232
Capacity(GB):   18000.21        TotalFree(GB):  0
Status:         SINGLE          Flag:           LEGACY
Read Ahead:     enabled         Write Cache:    enabled
TCQ:            --              NCQ:            enabled
------------------------------------------------------------------
                      S.M.A.R.T Attributes
Status: S.M.A.R.T OK.
ID  Name                    Threshold  Value    Worst    Status
------------------------------------------------------------------
1   Read Error Rate sdddsd      1       100      100      OK
2   Throughput Performance      54      135      135      OK
3   Spin-up Times   ddd         1       83       83       OK
4   Start/Stop Count            0       100      100      OK
5   Re-allocated Sector Count   1       100      100      OK
7   Seek Error Rate             1       100      100      OK
8   Seek Time Performance       20      140      140      OK
9   Power-on Hours Count        0       100      100      OK
A   Spin-up Retry Count         1       100      100      OK
C   Drive Power Cycle Count     0       96       96       OK
16  Unknown Attribute           25      100      100      OK
C0  Power-Off Retract Count     0       100      100      OK
C1  Emergency Retract Cycle Ct  0       100      100      OK
C2  HDA Temperature             0       60       60       OK
C4  Relocation Event Count      0       100      100      OK
C5  Current Pending Sector Count 0      100      100      OK
C6  Off-line Scan Uncorrectable Sector Count 0    100      100      OK
C7  Ultra ATA CRC Error Rate    0       100      100      OK
------------------------------------------------------------------
```

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 a number or set of numbers.

Attributes:
Type:
SSD7202/7502:
The array's type (RAID0, RAID1)

SSD7101A-1/7105/7120/7204/7104/6540/6540M/7505/7184/7180/7140A/7540/7580:
The array's type (RAID0, RAID1, RAID10)

RR3700 series/RR2800 Series/RR800 Series/RR2700 Series/RR4500 Series/RS643xTS/RR6628A/RR6674T:
The array's type (RAID0, RAID1, RAID10, RAID5, RAID50, RAID6)

RR600 Series (*RR620L/620 not support RAID10, RAID5*):
The array's type (RAID0, RAID1, RAID10, RAID5)

Status:

- NORMAL: Array status is normal
- DISABLED: Array is disabled.
- REBUILDING: Array is being rebuilt
- VERIFYING: Array is verifying
- INIT(F): Initializing an array using Foreground mode
- INIT(B): Initializing an array using Background mode
- UNINITIALIZED: Array is not initialized
- CRITICAL: Array is in a degraded status (no data redundancy)

Block:
Array Block size.

Sector:
Bytes per sector.

Cache:
Array Cache Policy
WT: Write Through
WB: Write Back
NONE: No Cache policy enabled

Example

HPT CLI> query arrays

```
HPT CLI > query arrays
ID   Capacity(GB)     Type        Status   Block  Sector  Cache          Name
---------------------------------------------------------------------------------
1        500.03   RAID1         NORMAL     --     512B   NONE        RAID_1_0
```

query arrays {array_id}

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

Example

HPT CLI> query arrays 1
Typical output:

```
HPT CLI > query arrays 1
ID:              1                   Name:          RAID_1_0
Type:            RAID1               Status:        NORMAL
Capacity(GB):    500.03              BlockSize:     --
SectorSize:      512B                CachePolicy:   NONE
Progress:        --
ID       Capacity    MaxFree    Flag    Status    ModelNumber
---------------------------------------------------------------------------------
1/E1/1   500.03      0          NORMAL  RAID      Samsung SSD 970 EVO Plus 500GB
1/E1/2   500.03      0          NORMAL  RAID      Samsung SSD 970 EVO Plus 500GB
```

```
HPT CLI>query arrays 1

ID:             1              Name:          RAID0_0
Type:           RAID0          Status:        NORMAL
Capacity(GB):   144001.13      BlockSize:     64k
SectorSize:     512B           CachePolicy:   NONE
Progress:       --
WWN:            200193cba9090fd00
ID      Capacity    MaxFree    Flag     Status   ModelNumber
--------------------------------------------------------------------
1/9     18000.14    0          NORMAL   RAID     WDC WUH721818ALE6L4
1/10    18000.14    0          NORMAL   RAID     WDC WUH721818ALE6L4
1/11    18000.14    0          NORMAL   RAID     WDC WUH721818ALE6L4
1/12    18000.14    0          NORMAL   RAID     WDC WUH721818ALE6L4
1/13    18000.14    0          NORMAL   RAID     WDC WUH721818ALE6L4
1/14    18000.14    0          NORMAL   RAID     WDC WUH721818ALE6L4
1/15    18000.14    0          NORMAL   RAID     WDC WUH721818ALE6L4
1/16    18000.14    0          NORMAL   RAID     WDC WUH721818ALE6L4
--------------------------------------------------------------------
```

Create Command

This command allows you to create a new RAID array, or add a spare disk,or expand/migrate an existing array.

Note: A drive must be initialized first before being used to create arrays.

Syntax

create {RAID0|RAID1|RAID10|RAID5|RAID50|RAID6|spare}[create-options]

Note: RR600 series no support RAID5/RAID50/RAID6 and RR620L/620 no support RAID10

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., it is in the format 'disks=1/E1/1,1/E1/2 or disks=*' for NVMe RAID HBAs; it is in the format 'disks=1/1,1/2 or disks=*' for SAS/SATA RAID HBAs. The character * means all available drives.

NOTE: When you enter a complete command with parametersdisks=* at the shell prompt, the correct writing is disks="*".

For example:
hptraidconf –u RAID –p hpt create RAID0 disks="*".

init=   specifies the initialization option (foreground, background, quickinit, keep old data). 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. Initialization is needed for **redundant arrays** to provide data

redundancy (*RAID0 no support foreground and background*)

foreground: Initialize array using foreground mode. This is therecommended method when creating redundant RAID arrays.

background: Initialize array using background mode. The arrayis accessible during array initialization.

keep old data: This option will create the RAID array but keep existing data on RAID array. This option should be selected when trying to recover a RAID array.

quickinit: Performs a quick initialization, and allows the array to be immediately used. This option will delete previous data from the disks but will not build parity. For this reason, it is not recommended for RAID 5, 6 or 50.

name=   specifies the name for the array being created.

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

src=   specifies an existing array to be expanded/migrated. All data on the source array will be redistributed online to the target array. **If this parameter is omitted, a new array is created**.

This parameter is only applicable to the "**Online Capacity Expand/ Online RAID Level Migration (OCE/ORLM)**" function.

capacity= specifies the capacity, in size of MB, for the target array. Maximum capacity is default.

bs=   specifies the block size, in KB, for the target array. This option is only valid for striped RAID levels.

sector=   specifies the logical sector size, in B/KB, for the target array. This option is only valid for striped RAID levels. The default is 512 Bytes.

matrix= n*m
When create RAID50 to specify the matrix options.
n: number of subarray's disk, m: number of subarray.
For example: When create a RAID50 the option matrix can be matrix=3*2. That means 2 RAID5s each with 3 disks to from a RAID50.

Example

OCE (Online Capacity Expand)**:**

HPT CLI> create RAID0 disks=* capacity=500GB init=quickinit bs=64k src=1

```
HPT CLI > create RAID0 disks=* capacity=300GB init=quickinit bs=64k
Create array successfully.

HPT CLI > query arrays
ID   Capacity(GB)    Type        Status   Block  Sector  Cache          Name
-------------------------------------------------------------------------------
1          300.00    RAID0       NORMAL   64k    512B    NONE        RAID0_3

HPT CLI > create RAID0 disks=* capacity=500GB init=quickinit bs=64k src=1

HPT CLI > query arrays
ID   Capacity(GB)    Type        Status   Block  Sector  Cache          Name
-------------------------------------------------------------------------------
1          500.00    RAID0       NORMAL   64k    512B    NONE        RAID0_4

HPT CLI >
```

When the capacity of your RAID is insufficient and you need a larger capacity, you can use this command to expand the capacity, and this operation will not damage your original data，**src= query arrays {arrays_id}**

Note：This command must be performed on an existing RAID

HPT CLI> create RAID0 disks=1/1,1/3,1/4,1/5 capacity=72000GB init=quickinit bs=64k src=1

```
HPT CLI > create RAID0 disks=1/3,1/4,1/5 capacity=* init=quickinit bs=64k
Create array successfully.

HPT CLI > query arrays
ID   Capacity(GB)    Type        Status   Block  Sector  Cache          Name
-------------------------------------------------------------------------------
1        54000.42    RAID0       NORMAL   64k    512B    NONE        RAID0_17

HPT CLI > create RAID0 disks=1/3,1/4,1/5,1/1 capacity=72000GB init=quickinit bs=64k src=1

HPT CLI > query arrays
ID   Capacity(GB)    Type        Status   Block  Sector  Cache          Name
-------------------------------------------------------------------------------
1        54000.42    RAID0       EXP/MIG  64k    512B    NONE        RAID0_17
1.1      72000.00    RAID0       EXP/MIG  64k    512B    NONE        RAID0_18

HPT CLI >
```

This command expands the RAID0 of 3 disks to RAID0 of 4 disks, The capacity of a single disk is 18TB, the capacity of 3-disk RAID0 is 54TB, and the capacity is expanded to 4-disk RAID0, and the capacity is 72TB.

If you want to stop the conversion, please refer to: OCE/ORLM command

ORLM (Online RAID Level Migration)**:**

HPT CLI> create RAID5 disks=* capacity=500GB init=quickinit bs=64k src=1

```
HPT CLI > query arrays
ID   Capacity(GB)    Type        Status   Block  Sector  Cache          Name
-------------------------------------------------------------------------------
1          500.00    RAID0       NORMAL   64k    512B    NONE        RAID0_4

HPT CLI > create RAID5 disks=* capacity=500GB init=quickinit bs=64k src=1

HPT CLI > query arrays
ID   Capacity(GB)    Type        Status   Block  Sector  Cache          Name
-------------------------------------------------------------------------------
1          500.00    RAID0       EXP/MIG  64k    512B    NONE        RAID0_4
1.1        500.00    RAID5       EXP/MIG  64k    512B    WB          RAID5_5

HPT CLI >
```

This command converts RAID0 to RAID5. The conversion process will take a while, depending on the size of the partition. After the conversion starts, it is recommended to wait for the conversion to end, and do not stop in the middle to avoid unnecessary troubles. **src= query arrays {arrays_id}**

Note：This command must be performed on an existing RAID

If you want to stop the conversion, please refer to: OCE/ORLM command

Create a RAID/spare:

HPT CLI> create RAID0 disks=* capacity=* init=quickinit bs=512k

```
HPT CLI > create RAID0 disks=* capacity=* init=quickinit bs=512k

HPT CLI > query arrays 1
ID:            1                Name:          RAID0_0
Type:          RAID0            Status:        NORMAL
Capacity(GB):  4096.33          BlockSize:     512k
SectorSize:    512B             CachePolicy:   NONE
Progress:      --
ID      Capacity    MaxFree     Flag     Status    ModelNumber
-----------------------------------------------------------------------
1/E1/1  512.04      0           NORMAL   RAID      Samsung SSD 970 PRO 512GB
1/E1/2  512.04      0           NORMAL   RAID      Samsung SSD 970 PRO 512GB
1/E1/3  512.04      0           NORMAL   RAID      Samsung SSD 970 PRO 512GB
1/E1/4  512.04      0           NORMAL   RAID      Samsung SSD 970 PRO 512GB
1/E2/1  1000.12     488.08      NORMAL   RAID      WDS100T3X0C-00SJG0
1/E2/2  1000.12     488.08      NORMAL   RAID      WDS100T3X0C-00SJG0
1/E2/3  1000.12     488.08      NORMAL   RAID      WDS100T3X0C-00SJG0
1/E2/4  1000.12     488.08      NORMAL   RAID      WDS100T3X0C-00SJG0
-----------------------------------------------------------------------
```

This command instructs the system to create a RAID0 array using the disks attached to controller 1 channels 1/2/3/4, and controller 2 channels 1/2/3/4; capacity is maximum, Block Size is 512KB

HPT CLI> create RAID0 disks=* capacity=100000 init=quickinit bs=512k

```
HPT CLI > create RAID0 disks=* capacity=100000 init=quickinit bs=512k

HPT CLI > query arrays 1
ID:            1                Name:          RAID0_0
Type:          RAID0            Status:        NORMAL
Capacity(GB):  100.00           BlockSize:     512k
SectorSize:    512B             CachePolicy:   NONE
Progress:      --
ID      Capacity    MaxFree     Flag     Status    ModelNumber
-----------------------------------------------------------------------
1/E1/1  500.03      450.03      NORMAL   RAID      Samsung SSD 970 EVO Plus 500GB
1/E1/2  500.03      450.03      NORMAL   RAID      Samsung SSD 970 EVO Plus 500GB
-----------------------------------------------------------------------

HPT CLI >
```

This command instructs the system to create a RAID0 array using the disks attached to controller 1 channels 1 and 2; capacity is 100GB, Block Size is 512KB.

HPT CLI> create spare disks=1/E1/1

```
HPT CLI > create spare disks=1/E1/1

HPT CLI > query devices
ID      Capacity    MaxFree     Flag     Status    ModelNumber
-----------------------------------------------------------------------
1/E1/1  500.03      450.03      RAID     SPARE     Samsung SSD 970 EVO Plus 500GB
1/E1/2  500.03      450.03      RAID     NORMAL    Samsung SSD 970 EVO Plus 500GB
-----------------------------------------------------------------------

HPT CLI >
```

This command instructs the system to set the disk on controller 1 channel 1 to function as 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.

Note： If you want to use a single disk after deleting the RAID, please restart the system after deleting the RAID. When the single disk status shows the Legacy status in WEBGUI or CLI, it can be used normally.

Syntax

delete {array_or_spare_ID}

Examples

HPT CLI> delete 1

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

HPT CLI> delete 1/E1/1

This command is used to remove the spare disk on controller 1 channel 1

Unplug Command

This command allows you to unplug an existing RAID array or device. After you have unplugged the array or device, you can hot plug it. Also, by running the rescan command you can find it back. For more information, please see **Rescan** .

In order to ensure data security, use the **unplug** command before physically removing any SAS/SATA HDD or SSD from the controller/enclosure.

Note: The SSD7580B controller is the only NVMe RAID solution that supports NVMe Hot-Plug and Hot-Swap.

Syntax

> unplug {array_id or device_id}

Example

HPT CLI> unplug 1

```
HPT CLI > query arrays
ID   Capacity(GB)   Type        Status   Block  Sector   Cache        Name
-------------------------------------------------------------------------
1        500.03    RAID1       NORMAL     --     512B    NONE       RAID1_3

HPT CLI > unplug 1

HPT CLI > query arrays
ID   Capacity(GB)   Type        Status   Block  Sector   Cache        Name
-------------------------------------------------------------------------

HPT CLI >
```

This command instructs the controller to disconnect array "1". Once executed, the arrays' drives can be safely removed.

HPT CLI> unplug 1/E1/1

```
HPT CLI > query devices
ID     Capacity   MaxFree    Flag     Status    ModelNumber
------------------------------------------------------------------------
1/E1/1 500.03     500.03     SINGLE   NORMAL    Samsung SSD 970 EVO Plus 500GB
1/E1/2 500.03     500.03     SINGLE   NORMAL    Samsung SSD 970 EVO Plus 500GB
------------------------------------------------------------------------

HPT CLI > unplug 1/E1/1

HPT CLI > query devices
ID     Capacity   MaxFree    Flag     Status    ModelNumber
------------------------------------------------------------------------
1/E1/2 500.03     500.03     SINGLE   NORMAL    Samsung SSD 970 EVO Plus 500GB

HPT CLI >
```

This command allows you to remove a disk from a running system without shutting down.

Init Commands

You can use init commands to initialize disks or arrays. **A drive must be initialized first before being used to create arrays.**

Syntax

- ■ init {device_id}
- ■ init {array_id} {start|stop}

init {device_id}

> This command initialize a disk for first use or a legacy disk on the controller.

Example

HPT CLI> init 1/E1/1

After entering the CLI, enter the command: '**query devices**' to view the current NVMe status is '**LEGACY**', enter '**init 1/E1/1**', NVMe status is 'NORMAL'.

HPT CLI> init 1/9

After entering the CLI, enter the command: '**query devices**' to view the current NVMe status is '**LEGACY**', enter '**init 1/E1/1**', NVMe status is 'NORMAL'.

Note: This command instructs the controller to initialize the disk on controller 1 channel 1. All data on the disk will be destroyed.

Init {array_id} {start|stop}

This command starts/stops initialization process on a redundant array.

Example

HPT CLI> init 1 stop

This command instructs the controller to stop the initialization process on array 1.

Rebuild Commands

You can use rebuild commands to rebuild a RAID1, RAID1/0, RAID5, RAID6 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

HPT CLI> rebuild 1 1/E1/1

```
HPT CLI > rebuild 1 1/E1/1

HPT CLI > query arrays
ID   Capacity(GB)    Type       Status   Block  Sector  Cache        Name
-------------------------------------------------------------------------------
1         500.03    RAID1      CRITICAL    --    512B    NONE       RAID1_3
```

This command instructs the controller to add the disk "**1/E1/1**" 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 stopping a rebuild process, you can resume it at a later time using the rebuild start command.

Examples

HPT CLI> rebuild 1 start

```
HPT CLI > rebuild 1 start

HPT CLI > query arrays
ID   Capacity(GB)    Type       Status   Block  Sector  Cache        Name
-------------------------------------------------------------------------------
1         500.03    RAID1     REBUILDING   --    512B    NONE       RAID1_3
```

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

HPT CLI> rebuild 1 stop

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

Verify Command

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

Syntax

- verify {array_id} {start|stop}

Examples

HPT CLI> verify 1 start

This command starts to verify the array "1".

HPT CLI> verify 1 stop

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

OCE/ORLM Command

This command starts or stops the **Online Capacity Expand/ Online RAID Level Migration** process on the specified array.

Syntax

- OCE/ORLM {array_id} {start|stop}

Examples

HPT CLI> OCE/ORLM 1 stop

Note: If you want to implement this function, you need to execute RAID OCE/ORLM first. For detailed steps, see **Create Commands**.

Rescan Command

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

Syntax

- ■ Rescan

Example

HPT CLI> rescan

Lscard Commands

The lscard command is used to list multi RAID controllers.

Syntax

Lscard

Example

HPT CLI> lscard

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

HPT CLI> events

Typical output:

events save {file_name}

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

Example

HPT CLI> events save C:/raidlog.txt

For Windows this command will save all the events to C**:/raidlog.txt**.

HPT CLI> events save /home/test/Documents/raidlog.txt

For Linux this command will save all the events to **/home/test/Documents/raidlog.txt**

Mail Commands

These commands can be used to configure Email recipients for the Email Notification features.

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 server

- mail server set {server_address}{port} { status } {from_address} [username]
  [password]

- mail server set {a|p|s|m|u|t} {value}

mail recipient

--- List all the mail recipients

Example

HPT CLI> mail recipient

Typical output:

```
HPT CLI > mail recipient
ID   Name      Mail  Address                    Notify Types
-----------------------------------------------------------------------
1    hpt       yzhang@highpoint-tech.com        Information Warning Error
```

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

--- Add a new Email recipient

Example

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

```
HPT CLI > mail recipient add hpt yzhang@highpoint-tech.com Inf War Err

HPT CLI > mail recipient
ID   Name      Mail  Address                    Notify Types
-----------------------------------------------------------------------
1    hpt       yzhang@highpoint-tech.com        Information Warning Error
```

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

mail recipient delete {recipient_name}

--- Delete an exist recipient.

Example

HPT CLI> mail recipient delete hpt

mail recipient test {recipient_name}

--- Send a test mail to specified recipient.

Example

HPT CLI> mail recipient test hpt

You will receive a test email.

Mon, 11 May 2020 07:52:30 :
This is a test mail.

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

--- Set the notification type for a recipient.

Example

HPT CLI> mail recipient set admin War Err

mail server

--- Display the SMTP server information

Example

HPT CLI> mail server
Typical output:

Mail server set {server_address}{port} {ssl}{status}{for_address} [uersname]

[password]

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

{server_address} – SMTP server address

{port}　　　　　 – port, generally 25

{ssl}　　　　　 – used ssl, '1' for enable and port need 465, '0' for

disble

{status}　　　　 – status, 'e' for enable or 'd' for disable

{from_address} – mail from address

{username}　　 – username

{password}　　 – the user's password

Examples:

HPT CLI> mail server set secure.emailsrvr.com 465 1 e

name@somecompany.com[**name@somecompany.com**](name@somecompany.com) **password**

HPT CLI> mail server set mail.somecompany.com 25 0 e admin@somecompany.com

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

---This command can be used to configure the Email server parameters:

Parameters

a – SMTP server address

p – port, generally 25

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

m – mail from address

u – username

t – user's password

Examples:

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

--- Change the server address

HPT CLI> mail server set p 465

--- Change the port

```
HPT CLI > mail server set secure.emailsrvr.com 25 0 e yzhang@highpoint-tech.com yzhang@highpoint-tech.com

HPT CLI > mail server
ServerAddress       Port  ssl Status   Mail From           User Name
-----------------------------------------------------------------------
secure.emailsrvr.com25   0    Enabled  yzhang@highpoint-tech.comyzhang@highpoint-tech.com
```

HPT CLI> mail server set s d

--- Disable Email notification

```
HPT CLI > mail server set s d

HPT CLI > mail server
ServerAddress       Port   ssl Status   Mail From           User Name
------------------------------------------------------------------------
smtp.163.com        465    0   Disabled yzhang@highpoint-tech.comyzhang@highpoint-tech.com
```

HPT CLI> mail server set s e

--- Enable Email notification

```
HPT CLI > mail server set s e

HPT CLI > mail server
ServerAddress       Port   ssl Status   Mail From           User Name
------------------------------------------------------------------------
smtp.163.com        465    0   Enabled  yzhang@highpoint-tech.comyzhang@highpoint-tech.com
```

Task Commands

When an array requires regular rebuild or verification sessions, you can use the task commands to automate this process in the background. As long as you have the appropriate privileges, you can add new tasks, and modify or deleteexisting tasks.

Execute the '**help task**' command to view:

Syntax

- ■ task

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

- ■ task delete {task_id}

- ■ task enable {task_id}

- ■ task disable {task_id}

task

This command displays detailed information about all scheduled tasks.

Example

HPT CLI> task

This command displays the current background tasks.

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

This command allows you to schedule the frequency by once, daily, weekly 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

HPT CLI> task rebuild 1 name=test once start=5/11/2020 time=17:03:35

This command adds a task schedule named **test** to rebuild the array"**1**"at **17:03:35** on **5/11/2020**. The rebuild frequency is set to **once**.

HPT CLI> task rebuild 4 name=myraid4 daily=2

start=2/8/2020 end=2/22/2020 time=13:49:58

This command adds a task schedule named **myraid4** to rebuild thearray "**4**" at **13:49:58** every **2** days from **2/8/2020** to **2/22/2020**.

HPT CLI> task rebuild 3 name=myraid3 weekly=2

interval=3 start=2/8/2020 end=2/22/2020 time=13:49:58

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** weeksfrom **2/8/2020** to **2/22/2020**

HPT CLI> task rebuild 2 name=myraid2 monthly=3

interval=4 start=2/8/2020 end=2/8/2020 time=12:30:33

This command adds a task schedule named **myraid3** to rebuild the

array "**2**" at **12:30:33** on the **3ʳᵈ** day of a month every **4** months

from **2/8/2020** to **2/8/2020**.

task verify{array_id}{name=}{once|daily|weekly|monthly}={day}interval={interval} start=mm/dd/yyyy end=mm/dd/yyyy time=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.

Examples

HPT CLI> task verify 1 name=test once start=5/11/2020 time=17:12:33

```
HPT CLI >│task verify 1 name=test once start=5/11/2020 time=17:12:23 │

HPT CLI > task
ID   Name      Start-Date     End-Date     S-F       Description
--------------------------------------------------------------------
1    test      05/11/2020     N/A          E-O       Verify raid RAID_1_0 (created by )

HPT CLI >
```

delete {task_id}

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

Example

HPT CLI> task delete 1

```
HPT CLI > task
ID   Name      Start-Date     End-Date     S-F       Description
--------------------------------------------------------------------
1    test      05/11/2020     N/A          E-O       Verify raid RAID_1_0 (created by )

HPT CLI >│task delete 1 │

HPT CLI > task
ID   Name      Start-Date     End-Date     S-F       Description
--------------------------------------------------------------------

HPT CLI >
```

This command will delete the task *"1"*.

task enable {task_id}

This command will enable a disabled task.

Example

HPT CLI> task enable 1

```
HPT CLI > task enable 1

HPT CLI > task
ID   Name      Start-Date      End-Date         S-F        Description
---------------------------------------------------------------------------
1    test      05/11/2020      N/A              E-O        Verify raid RAID_1_0 (created by )
```

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

task disable {task_id}

This command will disable a scheduled task manually.

Example

HPT CLI> task disable 1

```
HPT CLI > task disable 1

HPT CLI > task
ID   Name      Start-Date      End-Date         S-F        Description
---------------------------------------------------------------------------
1    test      05/11/2020      N/A              D-O        Verify raid RAID_1_0 (created by )
```

This command will disable the scheduled task '1'

Set Commands

Set the system, device or array's param.

Syntax

- set

- set [name]={value}

set

Show the system settable parameters.

Windows:

```
HPT CLI > set
----------------------------------------------------------------------
                Show the system setable parameters.
----------------------------------------------------------------------
[AR] Auto Rebuild                        Enable
[CE] Continue Rebuild On Error           Enable
[AA] Audible Alarm                       Enable
[RP] Rebuild Priority                    Medium
[SD] Spindown Idle Disk (minutes)        Disable
[BP] Beeper                              Enable
[FS] Eclosure Fan Speed                  Auto
[TT] Temperature threshold               149
[TU] Temperature unit                    F
[PS] Password                            --
----------------------------------------------------------------------

HPT CLI >
```

Linux:

```
HPT CLI>set
-------------------------------------------------------------
               Show the system setable parameters.
-------------------------------------------------------------
[AR] Auto Rebuild                        Disable
[CE] Continue Rebuild On Error           Enable
[RP] Rebuild Priority                    Medium
[SD] Spindown Idle Disk (minutes)        Disable
[BP] Beeper                              Disable
-------------------------------------------------------------
HPT CLI
```

set {name=}

set AR={y|n}}

Enable or Disable the [Auto Rebuild] parameter.

Example

HPT CLI> set AR=y

set CE={y|n}

Enable or Disable the [Continue Rebuilding On Error] parameter.

Example

HPT CLI> set CE=y

set AA={y|n}

Enable or Disable the [Audible Alarm] parameter.

Example

HPT CLI> set AA=y

set RP={0-100}

Change rebuilding priority. If a controller is not specified, this command will set the global rebuilding priority.

Note:

[0-12] Lowest
[13-37] Low
[38-67] Medium
[68-87] High
[>88] Highest

Example

HPT CLI> set RP=50

set SD={minutes}

Set value of [Spindown Idle Disk]
[1-10] 10
[11-20] 20
[21-30] 30
[31-60] 60
[61-120] 120
[121-180] 180
[181-240] 240

Example

HPT CLI> set SD=10

set BP={y|n}

Set enable or disable beeper.

Example

HPT CLI> set BP=y

set FS={Auto|Off|Low|Medium|High}

Change Enclosure Fan Speed.

Example

HPT CLI> set FS=Medium

set TT={value}, default=149F

Only support windows system.
Set temperature threshold.

Example

HPT CLI> set TT=135

set TU={C|F}
Only support windows system.
Set temperature unit to Celsius equals or Fahrenheit equals.

Example

HPT CLI> set TU=C

set PS

Only support windows system.
Set or change your password and confirm it.

Example

HPT CLI> set PS

```
HPT CLI > set PS
Password :******
Confirm  :******
Password has been changed, please login with your new password.
HighPoint Windows CLI, Please Input
        Password:
```

Diag Command

This command allows you to collect the diagnostic information.

Example

Linux：HPT CLI> diag

```
HPT CLI>diag

The diagnostic information has been saved in /usr/share/hpt/HighPoint_2021.04.07.
tar.gz
HPT CLI>
```

Windows：HPT CLI> diag

```
HPT CLI > diag
The diagnostic information will be saved in C:\Program Files (x86)\HighPoint Technologies, Inc\HighPoint RAID Management
\Service\webguiroot\HighPoint_rsnvme_1.3.19.0_2021.11.10_16.06.zip.It may take a few minutes to be ready.
```

The saving path will be displayed after entering this command.

Clear Commands

This command is used to clear screen.

Syntax

clear/cls/clr

Exit Command

Exit from the interactive mode and close to the window.

Syntax

exit