

# Hardware Monitor v1.2

Copyright © Riverbed Technology Inc. 2024

Created Mar 27, 2024 at 07:03 PM

# Resource: chassis

## Chassis data

http://{device}/api/npm.hardware\_monitor/1.2/chassis/items/{serial\_number}

### JSON

```
{
  "availability": string,
  "columns": integer,
  "headunit": boolean,
  "health": string,
  "model": string,
  "rows": integer,
  "sensor_status": {
    "cpus": [ cpu ],
    "exhaust_air_sensors": [ exhaust_air_sensor ],
    "intake_air_sensors": [ intake_air_sensor ],
    "power_supplies": [ power_supply ],
    "raid_controllers": [ raid_controller ],
    "storage_unit_controllers": [ storage_unit_controller ],
    "system_fans": [ system_fan ]
  },
  "serial_number": string,
  "unused_slots": [
    [
      integer
    ]
  ]
}
```

Property Name	Type	Description	Notes
<i>chassis</i>	<object>	Chassis data	
<i>availability</i>	<string>		Read-only; Values: ok, missing, foreign, invalid;
<i>chassis.columns</i>	<integer>	The number of columns of disk slots on the chassis. '0' is at the left.	Read-only; Optional; Minimum 0;
<i>chassis.headunit</i>	<boolean>	Indicates whether this is the head unit chassis.	Read-only; Optional;
<i>health</i>	<string>		Read-only; Values: ok, notice, failing, failed;
<i>chassis.model</i>	<string>		Read-only; Optional;
<i>chassis.rows</i>	<integer>	The number of rows of disk slots on the chassis. '0' is at the top.	Read-only; Optional; Minimum 0;
<i>chassis.sensor_status</i>	<object>	Status of the sensors on the chassis	Optional;
<i>chassis.sensor_status.cpus</i>	<array of <cpu>>		Optional;
<i>chassis.sensor_status.cpus[items]</i>	<cpu>		
<i>chassis.sensor_status.exhaust_air_sensors</i>	<array of <exhaust_air_sensor>>		Optional;
<i>chassis.sensor_status.exhaust_air_sensors[items]</i>	<exhaust_air_sensor>		
<i>chassis.sensor_status.intake_air_sensors</i>	<array of <intake_air_sensor>>		Optional;
<i>chassis.sensor_status.intake_air_sensors[items]</i>	<intake_air_sensor>		
<i>chassis.sensor_status.power_supplies</i>	<array of <power_supply>>		Optional;
<i>chassis.sensor_status.power_supplies[items]</i>	<power_supply>		
<i>chassis.sensor_status.raid_controllers</i>	<array of <raid_controller>>		Optional;
<i>chassis.sensor_status.raid_controllers[items]</i>	<raid_controller>		
<i>chassis.sensor_status.storage_unit_controllers</i>	<array of <storage_unit_controller>>		Optional;
<i>chassis.sensor_status.storage_unit_controllers[items]</i>	<storage_unit_controller>		
<i>chassis.sensor_status.system_fans</i>	<array of <system_fan>>		Optional;
<i>chassis.sensor_status.system_fans[items]</i>	<system_fan>		
<i>serial_number</i>	<string>		Read-only;

<code>chassis.unused_slots</code>	<code>&lt;array of &lt;array of &lt;integer&gt;&gt;&gt;</code>	A list of positions in the disk grid that are unused.	Optional;
<code>chassis.unused_slots[items]</code>	<code>&lt;array of &lt;integer&gt;&gt;</code>		
<code>chassis.unused_slots[items][items]</code>	<code>&lt;integer&gt;</code>		

## Links

### chassis: get

```
GET http://{device}/api/npm.hardware_monitor/1.2/chassis/items/{serial_number}
```

#### Response Body

Returns a [chassis](#) data object.

## Relations

### chassis: disks

#### Related resource

[disks](#)

#### Variables

Related var	Data value for replacement
chassis	0/serial_number

### chassis: instances

#### Related resource

[chassis\\_list](#)

## Resource: chassis\_list

Collection of chassis

```
http://{device}/api/npm.hardware_monitor/1.2/chassis{?availability}
```

#### JSON

```
{
  "items": [ chassis ]
}
```

Property Name	Type	Description	Notes
<code>chassis_list</code>	<code>&lt;object&gt;</code>	Collection of chassis	
<code>chassis_list.items</code>	<code>&lt;array of &lt;chassis&gt;&gt;</code>		Optional;
<code>chassis_list.items[items]</code>	<code>&lt;chassis&gt;</code>	Chassis data	

## Links

### chassis\_list: get

```
GET http://{device}/api/npm.hardware_monitor/1.2/chassis{?availability}
```

#### Response Body

Returns a [chassis\\_list](#) data object.

## Resource: disk

http://{device}/api/npm.hardware\_monitor/1.2/disks/items/{id}

## JSON

```
{
  "blink": boolean,
  "chassis": string,
  "device_type": string,
  "disk_type": string,
  "health": string,
  "id": string,
  "location": {
    "column": integer,
    "label": string,
    "row": integer
  },
  "model": string,
  "serial_number": string,
  "size_mb": integer,
  "status": string
}
```

Property Name	Type	Description	Notes
<i>disk</i>	<object>		Required properties: [id, blink];
<i>disk.blink</i>	<boolean>	Set to true to blink the drive for identification.	
<i>chassis</i>	<string>		Read-only;
<i>disk.device_type</i>	<string>		Read-only; Optional; Values: disk;
<i>disk.disk_type</i>	<string>	Disk technology	Read-only; Optional;
<i>health</i>	<string>		Read-only; Values: ok, notice, failing, failed;
<i>disk.id</i>	<string>	A unique identifier for this disk's slot on this system.	Read-only;
<i>location</i>	<object>		Read-only;
<i>location.column</i>	<integer>		Read-only; Optional; Minimum 0;
<i>location.label</i>	<string>		Read-only; Optional;
<i>location.row</i>	<integer>		Read-only; Optional; Minimum 0;
<i>disk.model</i>	<string>	Model string of the disk	Read-only; Optional;
<i>disk.serial_number</i>	<string>	Unique identifier of the disk	Read-only; Optional;
<i>disk.size_mb</i>	<integer>	Size of the disk in mebibytes	Read-only; Optional;
<i>status</i>	<string>		Read-only; Values: ok, failed, failing, rebuilding, missing, invalid, new;

## Links

### disk: get

GET http://{device}/api/npm.hardware\_monitor/1.2/disks/items/{id}

#### Response Body

Returns a [disk](#) data object.

### disk: set

PUT http://{device}/api/npm.hardware\_monitor/1.2/disks/items/{id}

#### Request Body

Provide a [disk](#) data object.

#### Response Body

Returns a [disk](#) data object.

## Relations

### disk: chassis

Related resource

[chassis](#)

Variables

Related var	Data value for replacement
serial_number	0/chassis

## disk: instances

Related resource

[disks](#)

## Resource: disks

`http://{device}/api/npm.hardware_monitor/1.2/disks{?status,serial_number,chassis,disk_type}`

JSON

```
{
  "items": [ disk ]
}
```

Property Name	Type	Description	Notes
<code>disks</code>	<code>&lt;object&gt;</code>		
<code>disks.items</code>	<code>&lt;array of &lt;disk&gt;&gt;</code>		Optional;
<code>disks.items[items]</code>	<code>&lt;disk&gt;</code>		

## Links

### disks: get

GET `http://{device}/api/npm.hardware_monitor/1.2/disks{?status,serial_number,chassis,disk_type}`

Response Body

Returns a [disks](#) data object.

## Resource: hardware\_change\_event

Event sent when the status of a hardware object changes.

`http://{device}/api/npm.hardware_monitor/1.2/event_queues/hardware_change_events/{id}`

JSON

```
{
  "action": string,
  "device_id": string,
  "device_type": string,
  "hdr": event_header,
  "jobs": [
    {
      "job_id": string,
      "name": string
    }
  ],
  "previous_status": string,
  "status": string
}
```

Property Name	Type	Description	Notes
<code>hardware_change_event</code>	<code>&lt;object&gt;</code>	Event sent when the status of a hardware object changes.	Required properties: [device_type, device_id];

<code>hardware_change_event.action</code>	<code>&lt;string&gt;</code>	Type of object change the event is describing.	Optional; Values: new, removed, changed;
<code>hardware_change_event.device_id</code>	<code>&lt;string&gt;</code>	Identifying unique attribute of the device	
<code>device_type</code>	<code>&lt;string&gt;</code>	The type of device that changed.	Values: disk, chassis;
<code>hardware_change_event.hdr</code>	<code>&lt;event_header&gt;</code>	Common properties for all events	
<code>hardware_change_event.jobs</code>	<code>&lt;array of &lt;object&gt;&gt;</code>	A list of jobs created in response to this event (future use).	Optional;
<code>hardware_change_event.jobs[items]</code>	<code>&lt;object&gt;</code>	A job record with name and job id	
<code>hardware_change_event.jobs[items].job_id</code>	<code>&lt;string&gt;</code>		Optional;
<code>hardware_change_event.jobs[items].name</code>	<code>&lt;string&gt;</code>		Optional;
<code>hardware_change_event.previous_status</code>	<code>&lt;string&gt;</code>	The previous status, if status changed.	Optional;
<code>hardware_change_event.status</code>	<code>&lt;string&gt;</code>	The current status of the device. Not included if device was removed.	Optional;
<code>hardware_change_event.oneOf[0]</code>	<code>&lt;object&gt;</code>		
<code>hardware_change_event.oneOf[0].device_type</code>	<code>&lt;string&gt;</code>		Optional; Values: disk;
<code>hardware_change_event.oneOf[0].previous_status</code>	<code>&lt;disk_status&gt;</code>		Values: ok, failed, failing, rebuilding, missing, invalid, new;
<code>hardware_change_event.oneOf[0].status</code>	<code>&lt;disk_status&gt;</code>		Values: ok, failed, failing, rebuilding, missing, invalid, new;
<code>hardware_change_event.oneOf[1]</code>	<code>&lt;object&gt;</code>		
<code>hardware_change_event.oneOf[1].availability</code>	<code>&lt;chassis_availability&gt;</code>		Values: ok, missing, foreign, invalid;
<code>hardware_change_event.oneOf[1].device_type</code>	<code>&lt;string&gt;</code>		Optional; Values: chassis;
<code>hardware_change_event.oneOf[1].previous_status</code>	<code>&lt;disk_status&gt;</code>		Values: ok, failed, failing, rebuilding, missing, invalid, new;

## Links

### hardware\_change\_event: get

GET `http://{device}/api/npm.hardware_monitor/1.2/event_queues/hardware_change_events/{id}`

#### Response Body

Returns a [hardware\\_change\\_event](#) data object.

## Relations

### hardware\_change\_event: instances

#### Related resource

[hardware\\_change\\_events](#)

## Resource: hardware\_change\_events

`http://{device}/api/npm.hardware_monitor/1.2/event_queues/hardware_change_event{?limit,since_id}`

#### JSON

```
{
  "items": [ hardware\_change\_event ]
}
```

Property Name	Type	Description	Notes
<code>hardware_change_events</code>	<code>&lt;object&gt;</code>		
<code>hardware_change_events.items</code>	<code>&lt;array of &lt;hardware_change_event&gt;&gt;</code>		Optional;
<code>hardware_change_events.items[items]</code>	<code>&lt;hardware_change_event&gt;</code>	Event sent when the status of a hardware object changes.	

---

## Links

### hardware\_change\_events: create

```
POST http://{device}/api/npm.hardware_monitor/1.2/event_queues/hardware_change_event{?limit,since_id}
```

#### Request Body

Provide a [hardware\\_change\\_event](#) data object.

#### Response Body

Returns a [hardware\\_change\\_event](#) data object.

### hardware\_change\_events: get

```
GET http://{device}/api/npm.hardware_monitor/1.2/event_queues/hardware_change_event{?limit,since_id}
```

#### Response Body

Returns a [hardware\\_change\\_events](#) data object.

---

## Resource: hardware\_health

```
http://{device}/api/npm.hardware_monitor/1.2/status
```

#### JSON

```
{
  "status": string
}
```

Property Name	Type	Description	Notes
<i>hardware_health</i>	<object>		
<i>status</i>	<string>		Read-only; Values: ok, notice, failing, failed;

---

## Links

### hardware\_health: get

```
GET http://{device}/api/npm.hardware_monitor/1.2/status
```

#### Response Body

Returns a [hardware\\_health](#) data object.

### hardware\_health: refresh

Force a refresh of hardware status

```
POST http://{device}/api/npm.hardware_monitor/1.2/status/refresh
```

#### Request Body

Do not provide a request body.

#### Response Body

On success, the server does not provide any body in the responses.

---

## Resource: volume

Data Volume

```
http://{device}/api/npm.hardware_monitor/1.2/volumes/items/{id}
```

## JSON

```
{
  "chassis": string,
  "health": string,
  "id": string,
  "mode": string,
  "roles": [
    string
  ],
  "status": string
}
```

Property Name	Type	Description	Notes
<i>volume</i>	<object>	Data Volume	Required properties: [id, mode, status, health, chassis, roles];
<i>chassis</i>	<string>		Read-only;
<i>health</i>	<string>		Read-only; Values: ok, notice, failing, failed;
<i>volume.id</i>	<string>		Read-only;
<i>mode</i>	<string>		Read-only; Values: , RAID0, RAID5, RAID6;
<i>volume.roles</i>	<array of <string>>		
<i>volume.roles[items]</i>	<string>		Values: os, data, capture;
<i>status</i>	<string>		Read-only; Values: ok, degraded, rebuilding, uninitialized, failed, missing, invalid;

## Links

### volume: get

```
GET http://{device}/api/npm.hardware_monitor/1.2/volumes/items/{id}
```

#### Response Body

Returns a [volumes](#) data object.

## Relations

### volume: chassis

#### Related resource

[chassis](#)

#### Variables

Related var	Data value for replacement
serial_number	0/chassis

### volume: instances

#### Related resource

[volumes](#)

## Resource: volumes

#### Data Volumes

```
http://{device}/api/npm.hardware_monitor/1.2/volumes{?status}
```

## JSON

```
{
  "items": [ volumes ]
}
```

Property Name	Type	Description	Notes
<i>volumes</i>	<object>	Data Volumes	Required properties: [items];
<i>volumes.items</i>	<array of <volumes>>		
<i>volumes.items[items]</i>	<volumes>	Data Volumes	

## Links

### volumes: get

```
GET http://{device}/api/npm.hardware_monitor/1.2/volumes{?status}
```

#### Response Body

Returns a [volumes](#) data object.

## Type: chassis\_availability

### JSON

```
string
```

Property Name	Type	Description	Notes
<i>chassis_availability</i>	<string>		Values: ok, missing, foreign, invalid;

## Type: cpu

### JSON

```
{
  "name": string,
  "status": sensor_status,
  "status_msg": string,
  "temperature": sensor_temperature
}
```

Property Name	Type	Description	Notes
<i>cpu</i>	<object>		Required properties: [status, name, status_msg, temperature];
<i>cpu.name</i>	<string>		
<i>cpu.status</i>	<sensor_status>		Values: ok, missing, unknown, failing, failed;
<i>cpu.status_msg</i>	<string>		
<i>cpu.temperature</i>	<sensor_temperature>		

## Type: device\_id

### JSON

```
string
```

Property Name	Type	Description	Notes
<i>device_id</i>	<string>		

## Type: device\_identity

### JSON

```
{
  "device_id": string,
  "manufacturer": string,
  "model": string
}
```

Property Name	Type	Description	Notes
<i>device_identity</i>	<object>		Required properties: [device_id];
<i>device_identity.device_id</i>	<string>		
<i>device_identity.manufacturer</i>	<string>		Optional;
<i>device_identity.model</i>	<string>		Optional;

## Type: device\_type

JSON

```
string
```

Property Name	Type	Description	Notes
<i>device_type</i>	<string>		Values: disk, chassis;

## Type: disk\_location

JSON

```
{
  "column": integer,
  "label": string,
  "row": integer
}
```

Property Name	Type	Description	Notes
<i>disk_location</i>	<object>		
<i>disk_location.column</i>	<integer>		Optional; Minimum 0;
<i>disk_location.label</i>	<string>		Optional;
<i>disk_location.row</i>	<integer>		Optional; Minimum 0;

## Type: disk\_status

JSON

```
string
```

Property Name	Type	Description	Notes
<i>disk_status</i>	<string>		Values: ok, failed, failing, rebuilding, missing, invalid, new;

## Type: exhaust\_air\_sensor

JSON

```
{
  "name": string,
  "status": sensor_status,
  "status_msg": string,
  "temperature": sensor_temperature
}
```

Property Name	Type	Description	Notes
---------------	------	-------------	-------

<i>exhaust_air_sensor</i>	<object>		Required properties: [status, name, status_msg, temperature];
<i>exhaust_air_sensor.name</i>	<string>		
<i>exhaust_air_sensor.status</i>	<sensor_status>		Values: ok, missing, unknown, failing, failed;
<i>exhaust_air_sensor.status_msg</i>	<string>		
<i>exhaust_air_sensor.temperature</i>	<sensor_temperature>		

## Type: health\_status

JSON

```
string
```

Property Name	Type	Description	Notes
<i>health_status</i>	<string>		Values: ok, notice, failing, failed;

## Type: intake\_air\_sensor

JSON

```
{
  "name": string,
  "status": sensor_status,
  "status_msg": string,
  "temperature": sensor_temperature
}
```

Property Name	Type	Description	Notes
<i>intake_air_sensor</i>	<object>		Required properties: [status, name, status_msg, temperature];
<i>intake_air_sensor.name</i>	<string>		
<i>intake_air_sensor.status</i>	<sensor_status>		Values: ok, missing, unknown, failing, failed;
<i>intake_air_sensor.status_msg</i>	<string>		
<i>intake_air_sensor.temperature</i>	<sensor_temperature>		

## Type: power\_supply

JSON

```
{
  "name": string,
  "status": sensor_status,
  "status_msg": string,
  "temperature": sensor_temperature
}
```

Property Name	Type	Description	Notes
<i>power_supply</i>	<object>		Required properties: [status, name, status_msg, temperature];
<i>power_supply.name</i>	<string>		
<i>power_supply.status</i>	<sensor_status>		Values: ok, missing, unknown, failing, failed;
<i>power_supply.status_msg</i>	<string>		
<i>power_supply.temperature</i>	<sensor_temperature>		

## Type: raid\_controller

JSON

```
{
  "battery": sensor_status,
  "name": string,
  "status": sensor_status,
  "status_msg": string,
  "temperature": sensor_temperature
}
```

Property Name	Type	Description	Notes
<i>raid_controller</i>	< <i>object</i> >		Required properties: [status, name, status_msg, battery, temperature];
<i>raid_controller.battery</i>	< <i>sensor_status</i> >		Values: ok, missing, unknown, failing, failed;
<i>raid_controller.name</i>	< <i>string</i> >		
<i>raid_controller.status</i>	< <i>sensor_status</i> >		Values: ok, missing, unknown, failing, failed;
<i>raid_controller.status_msg</i>	< <i>string</i> >		
<i>raid_controller.temperature</i>	< <i>sensor_temperature</i> >		

## Type: raid\_mode

JSON

```
string
```

Property Name	Type	Description	Notes
<i>raid_mode</i>	< <i>string</i> >		Values: , RAID0, RAID5, RAID6;

## Type: rpm

Revolutions per minute

JSON

```
integer
```

Property Name	Type	Description	Notes
<i>rpm</i>	< <i>integer</i> >	Revolutions per minute	

## Type: sensor\_status

JSON

```
string
```

Property Name	Type	Description	Notes
<i>sensor_status</i>	< <i>string</i> >		Values: ok, missing, unknown, failing, failed;

## Type: sensor\_temperature

JSON

```
{
  "current_temperature": integer,
  "threshold": integer
}
```

Property Name	Type	Description	Notes
---------------	------	-------------	-------

<i>sensor_temperature</i>	<object>		Required properties: [current_temperature, threshold];
<i>sensor_temperature.current_temperature</i>	<integer>	Current temperature in celsius	
<i>sensor_temperature.threshold</i>	<integer>	Threshold temperature in celsius	

## Type: storage\_unit\_controller

### JSON

```
{
  "name": string,
  "status": sensor_status,
  "status_msg": string,
  "temperature": sensor_temperature
}
```

Property Name	Type	Description	Notes
<i>storage_unit_controller</i>	<object>		Required properties: [status, name, status_msg, temperature];
<i>storage_unit_controller.name</i>	<string>		
<i>storage_unit_controller.status</i>	<sensor_status>		Values: ok, missing, unknown, failing, failed;
<i>storage_unit_controller.status_msg</i>	<string>		
<i>storage_unit_controller.temperature</i>	<sensor_temperature>		

## Type: system\_fan

### JSON

```
{
  "fan_speed": rpm,
  "name": string,
  "status": sensor_status,
  "status_msg": string
}
```

Property Name	Type	Description	Notes
<i>system_fan</i>	<object>		Required properties: [status, name, status_msg, fan_speed];
<i>system_fan.fan_speed</i>	<rpm>	Revolutions per minute	
<i>system_fan.name</i>	<string>		
<i>system_fan.status</i>	<sensor_status>		Values: ok, missing, unknown, failing, failed;
<i>system_fan.status_msg</i>	<string>		

## Type: volume\_status

### JSON

```
string
```

Property Name	Type	Description	Notes
<i>volume_status</i>	<string>		Values: ok, degraded, rebuilding, uninitialized, failed, missing, invalid;