Redfish Control Schema Proposal

Document Class: Informative

Document Status: Work in progress

Document Language: en-US

For information about patents held by third-parties which have notified the DMTF that, in their opinion, such patent may relate to or impact implementations of DMTF standards, visit http://www.dmtf.org/about/policies/disclosures.php.

This document's normative language is English. Translation into other languages is permitted.

Contents

Foreword

This Informational Specification covers a proposal to enhance the Power and Thermal monitoring schemas for Redfish.

IMPORTANT: These documents are not final. They do not necessarily reflect the views of the DMTF or its members. Because these documents are a Work in Progress, these documents may still change, perhaps profoundly and without notice. These documents are available for public review and comment until superseded.

The following files are part of the Redfish Power and Thermal Enhancement proposal:

- wip-controls - Mockup containing a "Widget" and Chassis showing examples of Control and Sensor components. The "Widget" is an example of a managed element that contains the excerpts of Control and Sensor resources to provide control and sensor data in context.
- DSP-IS0017_v0.5a.pdf - Presentation describing the Redfish Control proposal

How can I provide feedback?

Feedback on all Redfish specifications and documents is encouraged. Feedback can be directed to the DMTF and the Redfish Forum by the following means:

- **Redfish User Forum:** http://www.redfishforum.com User forum monitored by DMTF Redfish Forum personnel to answer questions about any Redfish-related topics.
- **DMTF Feedback Portal:** https://www.dmtf.org/standards/feedback Formal submission portal for enhancements or proposals to the DMTF and Redfish Forum.
- **Redfish Github Repository:** DMTF Redfish Forum member companies are encouraged to open issues on the group's private repository on Github.

Where can I find more information?

The following web sites provide more information about the Redfish standard:

- **Redfish Developer Hub:** http://redfish.dmtf.org Resources for developers building applications using Redfish. Contains an interactive schema explorer, hosted schema and other links.
- **Redfish User Forum:** http://www.redfishforum.com User forum monitored by DMTF Redfish personnel to answer questions about any Redfish-related topics.
- **DMTF Github Repositories:** http://www.github.com/DMTF Open source tools and libraries for working with the Redfish API.
- **Redfish Standards:** http://www.dmtf.org/standards/redfish Schemas, specs, mockups, white papers, FAQ, educational material and more.
- **DMTF Redfish Forum** (Working group that maintains the Redfish standard): http://www.dmtf.org/standards/spmf Companies involved, upcoming schedules and future work, charter, and information about joining.

Using the reference guide

The proposed power and thermal subsystem schemas are listed in the following section for reference. This section should be considered an extension to the contents of DSP2046, the Redfish Resource and Schema Guide, which lists the common Redfish properties, other object definitions, and all released Redfish schemas (including those shown here).

This guide was produced using the contents of the schema files from DMTF Redfish Schema bundle DSP8010 and merged with supplemental text using the DMTF's Redfish Documentation Generator.

Common Properties

Properties and objects defined for all Redfish schemas, or referenced by this white paper are detailed in the Redfish

Schema Reference Guide

Chassis 1.15.0

v1.15	v1.14	v1.13	v1.12	v1.11	v1.10	v1.9	v1.8	v1.7	v1.6	v1.5	...
TBD	2020.3	2020.2	2020.1	2019.4	2019.2	2018.3	2018.2	2018.1	2017.3	2017.1	...

The Chassis schema represents the physical components of a system. This resource represents the sheet-metal confined spaces and logical zones such as racks, enclosures, chassis and all other containers. Subsystems, such as sensors, that operate outside of a system's data plane are linked either directly or indirectly through this resource. A subsystem that operates outside of a system's data plane are not accessible to software that runs on the system.

URIs:

/redfish/v1/Chassis/*{ChassisId}*

Assembly *(v1.6+)* {	object		The link to the assembly associated with this chassis.
@odata.id }	string (URI)	read-only	The unique identifier for a resource.
AssetTag	string	read-write (null)	The user-assigned asset tag of this chassis.
ChassisType	string (enum)	read-only required	The type of physical form factor of the chassis. *For the possible property values, see ChassisType in Property details.*
Controls *(v1.15+)* {	object		The link to the collection of controls located in this chassis.
@odata.id }	string (URI)	read-only	The unique identifier for a resource.
DepthMm *(v1.4+)*	number (mm)	read-only (null)	The depth of the chassis.
Drives *(v1.14+)* {	object		The link to the collection of drives within this chassis.
@odata.id }	string (URI)	read-only	The unique identifier for a resource.
EnvironmentalClass *(v1.9+)*	string (enum)	read-only (null)	The ASHRAE Environmental Class for this chassis. *For the possible property values, see EnvironmentalClass in Property details.*
HeightMm *(v1.4+)*	number (mm)	read-only (null)	The height of the chassis.
IndicatorLED *(deprecated v1.14)*	string (enum)	read-write (null)	The state of the indicator LED, which identifies the chassis. *For the possible property values, see IndicatorLED in Property details. Deprecated in*

			v1.14 and later. This property has been deprecated in favor of the LocationIndicatorActive property.
Links {	object		The links to other resources that are related to this resource.
ComputerSystems [{	array		An array of links to the computer systems that this chassis directly and wholly contains.
@odata.id }]	string (URI)	read-only	The unique identifier for a resource.
ComputerSystems@odata.count	integer	read-only	The number of items in a collection.
ContainedBy {	object		The link to the chassis that contains this chassis.
@odata.id }	string	read-write	*Link to another Chassis resource.*
Contains [{	array		An array of links to any other chassis that this chassis has in it.
@odata.id }]	string	read-write	*Link to another Chassis resource.*
Contains@odata.count	integer	read-only	The number of items in a collection.
CooledBy [{	array		An array of links to resources or objects that cool this chassis. Normally, the link is for either a chassis or a specific set of fans.
@odata.id }]	string (URI)	read-only	The unique identifier for a resource.
CooledBy@odata.count	integer	read-only	The number of items in a collection.
Drives *(v1.2+)* [{	array		An array of links to the drives located in this chassis.
@odata.id }]	string (URI)	read-only	The unique identifier for a resource.
Drives@odata.count	integer	read-only	The number of items in a collection.
Facility *(v1.11+)* {	object		The link to the facility that contains this chassis.
@odata.id }	string (URI)	read-only	The unique identifier for a resource.
ManagedBy [{	array		An array of links to the managers responsible for managing this chassis.
@odata.id }]	string (URI)	read-only	The unique identifier for a resource.
ManagedBy@odata.count	integer	read-only	The number of items in a collection.
ManagersInChassis *(v1.2+)* [{	array		An array of links to the managers located in this chassis.
@odata.id }]	string (URI)	read-only	The unique identifier for a resource.
ManagersInChassis@odata.count	integer	read-only	The number of items in a collection.

Oem { }	object		The OEM extension property. *See the [Resource](#) schema for details on this property.*
PCIeDevices *(v1.4+, deprecated v1.10)* [{	array		An array of links to the PCIe devices located in this chassis. *Deprecated in v1.10 and later. This property has been deprecated in favor of the PCIeDevices resource collection in the root of this resource.*
@odata.id }]	string (URI)	read-only	The unique identifier for a resource.
PCIeDevices@odata.count	integer	read-only	The number of items in a collection.
PoweredBy [{	array		An array of links to resources or objects that power this chassis. Normally, the link is for either a chassis or a specific set of power supplies.
@odata.id }]	string (URI)	read-only	The unique identifier for a resource.
PoweredBy@odata.count	integer	read-only	The number of items in a collection.
Processors *(v1.9+)* [{	array		An array of links to the processors located in this chassis.
@odata.id }]	string (URI)	read-only	The unique identifier for a resource.
Processors@odata.count	integer	read-only	The number of items in a collection.
ResourceBlocks *(v1.5+)* [{	array		An array of links to the resource blocks located in this chassis.
@odata.id }]	string (URI)	read-only	The unique identifier for a resource.
ResourceBlocks@odata.count	integer	read-only	The number of items in a collection.
Storage *(v1.2+)* [{	array		An array of links to the storage subsystems connected to or inside this chassis.
@odata.id }]	string (URI)	read-only	The unique identifier for a resource.
Storage@odata.count	integer	read-only	The number of items in a collection.
Switches *(v1.7+)* [{	array		An array of links to the switches located in this chassis.
@odata.id }]	string (URI)	read-only	The unique identifier for a resource.
Switches@odata.count }	integer	read-only	The number of items in a collection.
Location *(v1.2+)* { }	object		The location of the chassis. *See the [Resource](#) schema for details on this property.*
LocationIndicatorActive *(v1.14+)*	boolean	read-write (null)	An indicator allowing an operator to physically locate this resource.

LogServices {	object		The link to the logs for this chassis.
@odata.id }	string (URI)	read-only	The unique identifier for a resource.
Manufacturer	string	read-only (null)	The manufacturer of this chassis.
MaxPowerWatts *(v1.12+)*	number (Watts)	read-only (null)	The upper bound of the total power consumed by the chassis.
MediaControllers *(v1.11+)* {	object		The link to the collection of media controllers located in this chassis.
@odata.id }	string (URI)	read-only	The unique identifier for a resource.
Memory *(v1.11+)* {	object		The link to the collection of memory located in this chassis.
@odata.id }	string (URI)	read-only	The unique identifier for a resource.
MemoryDomains *(v1.11+)* {	object		The link to the collection of memory domains located in this chassis.
@odata.id }	string (URI)	read-only	The unique identifier for a resource.
MinPowerWatts *(v1.12+)*	number (Watts)	read-only (null)	The lower bound of the total power consumed by the chassis.
Model	string	read-only (null)	The model number of the chassis.
NetworkAdapters *(v1.4+)* {	object		The link to the collection of network adapters associated with this chassis.
@odata.id }	string (URI)	read-only	The unique identifier for a resource.
PartNumber	string	read-only (null)	The part number of the chassis.
PCIeDevices *(v1.10+)* {	object		The link to the collection of PCIe devices located in this chassis.
@odata.id }	string (URI)	read-only	The unique identifier for a resource.
PCIeSlots *(v1.8+)* {	object		The link to the PCIe slot properties for this chassis.
@odata.id }	string (URI)	read-only	The unique identifier for a resource.
PhysicalSecurity *(v1.1+)* {	object		The state of the physical security sensor.
IntrusionSensor *(v1.1+)*	string (enum)	read-write (null)	This indicates the known state of the physical security sensor, such as if it is hardware intrusion detected. *For the possible property values, see* [IntrusionSensor](#) *in Property details.*

IntrusionSensorNumber *(v1.1+)*	integer	read-only (null)	A numerical identifier to represent the physical security sensor.
IntrusionSensorReArm *(v1.1+)* }	string (enum)	read-only (null)	The method that restores this physical security sensor to the normal state. *For the possible property values, see IntrusionSensorReArm in Property details.*
Power {	object		The link to the power properties, or power supplies, power policies, and sensors, for this chassis.
@odata.id }	string (URI)	read-only	The unique identifier for a resource.
PowerState *(v1.0.1+)*	string (enum)	read-only (null)	The current power state of the chassis. *For the possible property values, see PowerState in Property details.*
Sensors *(v1.9+)* {	object		The link to the collection of sensors located in the equipment and sub-components.
@odata.id }	string (URI)	read-only	The unique identifier for a resource.
SerialNumber	string	read-only (null)	The serial number of the chassis.
SKU	string	read-only (null)	The SKU of the chassis.
Status { }	object		The status and health of the resource and its subordinate or dependent resources. *See the Resource schema for details on this property.*
Thermal {	object		The link to the thermal properties, such as fans, cooling, and sensors, for this chassis.
@odata.id }	string (URI)	read-only	The unique identifier for a resource.
UUID *(v1.7+)*	string	read-only (null)	The UUID for this chassis.
WeightKg *(v1.4+)*	number (kg)	read-only (null)	The weight of the chassis.
WidthMm *(v1.4+)*	number (mm)	read-only (null)	The width of the chassis.

Actions

Reset

This action resets the chassis but does not reset systems or other contained resources, although side effects might occur that affect those resources.

Action URI: {Base URI of target resource}/Actions/Chassis.Reset

Perform the action using a POST to the specific Action URI for this resource. Parameters for the action are passed in a JSON body and are defined as follows:

{			

ResetType	string (enum)	optional	The type of reset.
}			*For the possible property values, see [ResetType](#) in Property details.*

Property details

ChassisType:

The type of physical form factor of the chassis.

string	Description
Blade	An enclosed or semi-enclosed, typically vertically-oriented, system chassis that must be plugged into a multi-system chassis to function normally.
Card	A loose device or circuit board intended to be installed in a system or other enclosure.
Cartridge	A small self-contained system intended to be plugged into a multi-system chassis.
Component	A small chassis, card, or device that contains devices for a particular subsystem or function.
Drawer	An enclosed or semi-enclosed, typically horizontally-oriented, system chassis that can be slid into a multi-system chassis.
Enclosure	A generic term for a chassis that does not fit any other description.
Expansion	A chassis that expands the capabilities or capacity of another chassis.
IPBasedDrive (v1.3+)	A chassis in a drive form factor with IP-based network connections.
Module	A small, typically removable, chassis or card that contains devices for a particular subsystem or function.
Other	A chassis that does not fit any of these definitions.
Pod	A collection of equipment racks in a large, likely transportable, container.
Rack	An equipment rack, typically a 19-inch wide freestanding unit.
RackGroup (v1.4+)	A group of racks that form a single entity or share infrastructure.
RackMount	A single-system chassis designed specifically for mounting in an equipment rack.
Row	A collection of equipment racks.
Shelf	An enclosed or semi-enclosed, typically horizontally-oriented, system chassis that must be plugged into a multi-system chassis to function normally.
Sidecar	A chassis that mates mechanically with another chassis to expand its capabilities or capacity.
Sled	An enclosed or semi-enclosed, system chassis that must be plugged into a multi-system chassis to function normally similar to a blade type chassis.
StandAlone	A single, free-standing system, commonly called a tower or desktop chassis.
StorageEnclosure (v1.6+)	A chassis that encloses storage.
Zone	A logical division or portion of a physical chassis that contains multiple devices or systems that cannot be physically separated.

EnvironmentalClass:

The ASHRAE Environmental Class for this chassis.

string	Description
A1	ASHRAE Environmental Class 'A1'.
A2	ASHRAE Environmental Class 'A2'.
A3	ASHRAE Environmental Class 'A3'.
A4	ASHRAE Environmental Class 'A4'.

IndicatorLED:

The state of the indicator LED, which identifies the chassis.

string	Description
Blinking	The indicator LED is blinking.
Lit	The indicator LED is lit.
Off	The indicator LED is off.
Unknown (deprecated v1.2)	The state of the indicator LED cannot be determined. *This value has been deprecated in favor of returning null if the state is unknown.*

IntrusionSensor:

This indicates the known state of the physical security sensor, such as if it is hardware intrusion detected.

string	Description
HardwareIntrusion	A door, lock, or other mechanism protecting the internal system hardware from being accessed is detected to be in an insecure state.
Normal	No abnormal physical security condition is detected at this time.
TamperingDetected	Physical tampering of the monitored entity is detected.

IntrusionSensorReArm:

The method that restores this physical security sensor to the normal state.

string	Description
Automatic	Because no abnormal physical security condition is detected, this sensor is automatically restored to the normal state.
Manual	A manual re-arm of this sensor restores it to the normal state.

PowerState:

The current power state of the chassis.

string	Description
Off	The components within the chassis have no power, except some components might continue to have AUX power, such as the management controller.
On	The components within the chassis have power.
PoweringOff	A temporary state between on and off. The components within the chassis can take time to process the power off action.

PoweringOn	A temporary state between off and on. The components within the chassis can take time to process the power on action.

ResetType:

The type of reset.

string	Description
ForceOff	Turn off the unit immediately (non-graceful shutdown).
ForceOn	Turn on the unit immediately.
ForceRestart	Shut down immediately and non-gracefully and restart the system.
GracefulRestart	Shut down gracefully and restart the system.
GracefulShutdown	Shut down gracefully and power off.
Nmi	Generate a diagnostic interrupt, which is usually an NMI on x86 systems, to stop normal operations, complete diagnostic actions, and, typically, halt the system.
On	Turn on the unit.
PowerCycle	Power cycle the unit. Behaves like a full power removal, followed by a power restore to the resource.
PushPowerButton	Simulate the pressing of the physical power button on this unit.

Example response

```json
{
    "@odata.type": "#Chassis.v1_11_0.Chassis",
    "Id": "1U",
    "Name": "Computer System Chassis",
    "ChassisType": "RackMount",
    "AssetTag": "Chicago-45Z-2381",
    "Manufacturer": "Contoso",
    "Model": "3500RX",
    "SKU": "8675309",
    "SerialNumber": "437XR1138R2",
    "PartNumber": "224071-J23",
    "PowerState": "On",
    "IndicatorLED": "Lit",
    "HeightMm": 44.45,
    "WidthMm": 431.8,
    "DepthMm": 711,
    "WeightKg": 15.31,
    "Location": {
        "PostalAddress": {
            "Country": "US",
            "Territory": "OR",
            "City": "Portland",
            "Street": "1001 SW 5th Avenue",
            "HouseNumber": 1100,
            "Name": "DMTF",
            "PostalCode": "97204"
        },
        "Placement": {
            "Row": "North",
            "Rack": "WEB43",
            "RackOffsetUnits": "EIA_310",
            "RackOffset": 12
        }
    },
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "Thermal": {
        "@odata.id": "/redfish/v1/Chassis/1U/Thermal"
    },
    "Power": {
        "@odata.id": "/redfish/v1/Chassis/1U/Power"
    },
    "Assembly": {
        "@odata.id": "/redfish/v1/Chassis/1U/Assembly"
    },
    "Links": {
        "ComputerSystems": [
```

```
            {
                "@odata.id": "/redfish/v1/Systems/437XR1138R2"
            }
        ],
        "ManagedBy": [
            {
                "@odata.id": "/redfish/v1/Managers/BMC"
            }
        ],
        "ManagersInChassis": [
            {
                "@odata.id": "/redfish/v1/Managers/BMC"
            }
        ]
    },
    "@odata.id": "/redfish/v1/Chassis/1U"
}
```

Control 0.5.0

v0.5
TBD

The Control schema describes a control point and its properties.

URIs:

/redfish/v1/Chassis/*{ChassisId}*/Controls/*{ControlId}*
/redfish/v1/PowerEquipment/Controls/*{ControlId}*

Accuracy *(v0.5+)*	number (%)	read-only (null)	The estimated percent error of measured versus actual values.
AssociatedSensors *(v0.5+)* [{	array		An array of links to the sensors associated with this control.
@odata.id }]	string	read-only	*Link to a Sensor resource. See the Links section and the Sensor schema for details.*
ControlLoop *(v0.5+)* {	object	(null)	The control loop details.
Differential *(v0.5+)*	number	read-write (null)	The differential coefficient.
Integral *(v0.5+)*	number	read-write (null)	The integral coefficient.
Proportional *(v0.5+)* }	number	read-write (null)	The proportional coefficient.
ControlType *(v0.5+)*	string (enum)	read-only (null)	The type of control. *For the possible property values, see ControlType in Property details.*
DeadBand *(v0.5+)*	number	read-write (null)	The maximum deviation from the set point allowed before the control will activate.
Increment *(v0.5+)*	number	read-only (null)	The smallest increment supported for the set point.
Location *(v0.5+)* { }	object		The location information for this control. *See the Resource schema for details on this property.*
OperatingMode *(v0.5+)*	string (enum)	read-write (null)	The current operating mode of the control. *For the possible property values, see OperatingMode in Property details.*

PhysicalContext *(v0.5+)*	string (enum)	read-only (null)	The area or device to which this control applies. *For the possible property values, see PhysicalContext in Property details.*
PhysicalSubContext *(v0.5+)*	string (enum)	read-only (null)	The usage or location within a device to which this control applies. *For the possible property values, see PhysicalSubContext in Property details.*
Sensor *(v0.5+)*	(excerpt)	read-only	The sensor reading associated with this control.
SetPoint *(v0.5+)*	number	read-write (null)	The desired set point of the control.
SetPointRangeMax *(v0.5+)*	number	read-only (null)	The maximum possible value for this control.
SetPointRangeMin *(v0.5+)*	number	read-only (null)	The minimum possible value for this control.
SetPointUnits *(v0.5+)*	string	read-only (null)	The units of the set point.
SetPointUpdateTime *(v0.5+)*	string (date-time)	read-only (null)	The date and time that the set point was changed.
Status *(v0.5+)* { }	object		The status and health of the resource and its subordinate or dependent resources. See the Resource schema for details on this property.

Property details

ControlType:

The type of control.

string	Description
AirFlow	Airflow.
Altitude	Altitude.
AngularPosition	Position (Angular).
Barometric	Barometric pressure.
Current	Current.
EnergyJoules	Energy (Joules).
EnergykWh	Energy (kWh).
Force	Force or strain.
Frequency	Frequency.
Humidity	Relative Humidity.
LinearPosition	Position (Linear).
LiquidFlow	Liquid flow.
LiquidLevel	Liquid level.

Luminosity	Luminosity.
Percent	Percent.
Power	Power.
Pressure	Pressure.
Rotational	Rotational.
Sound	Sound.
Temperature	Temperature.
Velocity	Velocity.
Voltage	Voltage (AC or DC).
Weight	Weight.

OperatingMode:

The current operating mode of the control.

string	Description
Automatic	Automatically adjust control to meet the set point.
Direct	The set point directly affects the control value.
Disabled	An automatic control loop has been disabled.
Physical	A physical control that cannot be adjusted through this interface.

PhysicalContext:

The area or device to which this control applies.

string	Description
Accelerator	An accelerator.
ACInput	An AC input.
ACMaintenanceBypassInput	An AC maintenance bypass input.
ACOutput	An AC output.
ACStaticBypassInput	An AC static bypass input.
ACUtilityInput	An AC utility input.
ASIC	An ASIC device, such as a networking chip or chipset component.
Back	The back of the chassis.
Backplane	A backplane within the chassis.
Chassis	The entire chassis.
ComputeBay	Within a compute bay.
CoolingSubsystem	The entire cooling, or air and liquid, subsystem.
CPU	A processor (CPU).

CPUSubsystem	The entire processor (CPU) subsystem.
DCBus	A DC bus.
Exhaust	The air exhaust point or points or region of the chassis.
ExpansionBay	Within an expansion bay.
Fan	A fan.
FPGA	An FPGA.
Front	The front of the chassis.
GPU	A graphics processor (GPU).
GPUSubsystem	The entire graphics processor (GPU) subsystem.
Intake	The air intake point or points or region of the chassis.
LiquidInlet	The liquid inlet point of the chassis.
LiquidOutlet	The liquid outlet point of the chassis.
Lower	The lower portion of the chassis.
Memory	A memory device.
MemorySubsystem	The entire memory subsystem.
Motor	A motor.
NetworkBay	Within a networking bay.
NetworkingDevice	A networking device.
PowerSubsystem	The entire power subsystem.
PowerSupply	A power supply.
PowerSupplyBay	Within a power supply bay.
Rectifier	A rectifier device.
Room	The room.
StorageBay	Within a storage bay.
StorageDevice	A storage device.
SystemBoard	The system board (PCB).
Transformer	A transformer.
Upper	The upper portion of the chassis.
VoltageRegulator	A voltage regulator device.

PhysicalSubContext:

The usage or location within a device to which this control applies.

string	Description
Input	The input.

Output	The output.

Example response

```
{
    "@odata.type": "#Control.v0_5_0.Control",
    "Id": "Thermostat",
    "Name": "System Thermostat",
    "PhysicalContext": "Chassis",
    "ControlType": "Temperature",
    "OperatingMode": "Automatic",
    "SetPoint": 22,
    "SetPointUnits": "cel",
    "SetPointRangeMax": 45,
    "SetPointRangeMin": 18,
    "SetPointUpdateTime": "2020-09-13T09:14:33",
    "Increment": 0.5,
    "DeadBand": 0.5,
    "Sensor": {
        "Reading": 22.5,
        "DataSourceUri": "/redfish/v1/Chassis/1U/Sensors/AmbientTemp"
    },
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    },
    "@odata.id": "/redfish/v1/Chassis/1/Controls/Thermostat",
    "@Redfish.Copyright": "Copyright 2014-2020 DMTF. For the full DMTF copyright policy, see
http://www.dmtf.org/about/policies/copyright."
}
```

Sensor 1.2.0

v1.2	v1.1	v1.0
TBD	2019.4	2018.3

The Sensor schema describes a sensor and its properties.

URIs:

/redfish/v1/Chassis/*{ChassisId}*/Sensors/*{SensorId}*

/redfish/v1/PowerEquipment/FloorPDUs/*{PowerDistributionId}*/Sensors/*{SensorId}*

/ redfish/ v1/ PowerEquipment/ RackPDUs/*{PowerDistributionId}*/Sensors/*{SensorId}*

/redfish/v1/PowerEquipment/Sensors/*{SensorId}*

/redfish/v1/PowerEquipment/TransferSwitches/*{PowerDistributionId}*/Sensors/*{SensorId}*

Accuracy	number (%)	read-only (null)	The estimated percent error of measured versus actual values.
AdjustedMaxAllowableOperatingValue	number	read-only (null)	The adjusted maximum allowable operating value for this equipment based on the environmental conditions.
AdjustedMinAllowableOperatingValue	number	read-only (null)	The adjusted minimum allowable operating value for this equipment based on the environmental conditions.
ApparentVA	number (V.A)	read-only (null)	The product of voltage and current for an AC circuit, in Volt-Ampere units.
CrestFactor *(v1.1+)*	number	read-only (null)	The crest factor for this sensor.
ElectricalContext	string (enum)	read-only (null)	The combination of current-carrying conductors. *For the possible property values, see ElectricalContext in Property details.*
Implementation *(v1.1+)*	string (enum)	read-only (null)	The implementation of the sensor. *For the possible property values, see Implementation*

			in Property details.
LifetimeReading *(v1.1+)*	number	read-only (null)	The total accumulation value for this sensor.
Links *(v1.2+)* {	object		The links to other resources that are related to this resource.
AssociatedControls *(v1.2+)* [{	array		An array of links to the controls that can affect this sensor.
@odata.id }]	string	read-only	*Link to a Control resource. See the Links section and the [Control](#) schema for details.*
AssociatedControls@odata.count	integer	read-only	The number of items in a collection.
Oem { } }	object		The OEM extension property. *See the [Resource](#) schema for details on this property.*
LoadPercent *(deprecated v1.1)*	number (%)	read-only (null)	The power load utilization for this sensor. *Deprecated in v1.1 and later. This property has been deprecated in favor of using a sensor instance with a ReadingType of `Percent` to show utilization values when needed.*
Location { }	object		The location information for this sensor. *See the [Resource](#) schema for details on this property.*
MaxAllowableOperatingValue	number	read-only (null)	The maximum allowable operating value for this equipment.
MinAllowableOperatingValue	number	read-only (null)	The minimum allowable operating value for this equipment.
PeakReading	number	read-only (null)	The peak sensor value.
PeakReadingTime	string (date-time)	read-only (null)	The time when the peak sensor value occurred.
PhysicalContext	string (enum)	read-only (null)	The area or device to which this sensor measurement applies. *For the possible property values, see [PhysicalContext](#) in Property details.*
PhysicalSubContext	string (enum)	read-only (null)	The usage or location within a device to which this sensor measurement applies. *For the possible property values, see [PhysicalSubContext](#) in Property details.*
PowerFactor	number	read-only (null)	The power factor for this sensor.
Precision	number	read-only (null)	The number of significant digits in the reading.
ReactiveVAR	number (V.A)	read-only (null)	The square root of the difference term of squared ApparentVA and squared Power (Reading) for a circuit, in var units.
Reading	number	read-only (null)	The sensor value.
ReadingRangeMax	number	read-only	The maximum possible value for this sensor.

		(null)	
ReadingRangeMin	number	read-only (null)	The minimum possible value for this sensor.
ReadingTime *(v1.1+)*	string (date-time)	read-only (null)	The date and time that the reading was acquired from the sensor.
ReadingType	string (enum)	read-only (null)	The type of sensor. *For the possible property values, see ReadingType in Property details.*
ReadingUnits	string	read-only (null)	The units of the reading and thresholds.
RelatedItem *(v1.2+)* [{	array		An array of links to resources or objects that this sensor services.
@odata.id }]	string (URI)	read-only	The unique identifier for a resource.
SensingFrequency *(deprecated v1.1)*	number	read-only (null)	The time interval between readings of the physical sensor. *Deprecated in v1.1 and later. This property has been deprecated in favor of the SensingInterval property, which uses the duration time format for interoperability.*
SensingInterval *(v1.1+)*	string	read-only (null)	The time interval between readings of the sensor.
SensorResetTime	string (date-time)	read-only (null)	The date and time when the time-based properties were last reset.
SpeedRPM *(v1.2+)*	number (RPM)	read-only (null)	The rotational speed.
Status { }	object		The status and health of the resource and its subordinate or dependent resources. *See the Resource schema for details on this property.*
THDPercent *(v1.1+)*	number	read-only (null)	The total harmonic distortion (THD).
Thresholds {	object		The set of thresholds defined for this sensor.
LowerCaution {	object		The value at which the reading is below normal range.
Activation	string (enum)	read-write (null)	The direction of crossing that activates this threshold. *For the possible property values, see Activation in Property details.*
DwellTime	string	read-write (null)	The duration the sensor value must violate the threshold before the threshold is activated.
Reading }	number	read-write (null)	The threshold value.
LowerCautionUser *(v1.2+)* {	object		The value at which the reading is below normal range.
Activation	string (enum)	read-write (null)	The direction of crossing that activates this threshold. *For the possible property values, see Activation in Property details.*

DwellTime	string	read-write (null)	The duration the sensor value must violate the threshold before the threshold is activated.
Reading }	number	read-write (null)	The threshold value.
LowerCritical {	object		The value at which the reading is below normal range but not yet fatal.
Activation	string (enum)	read-write (null)	The direction of crossing that activates this threshold. *For the possible property values, see [Activation](#) in Property details.*
DwellTime	string	read-write (null)	The duration the sensor value must violate the threshold before the threshold is activated.
Reading }	number	read-write (null)	The threshold value.
LowerCriticalUser *(v1.2+)* {	object		The value at which the reading is below normal range but not yet fatal.
Activation	string (enum)	read-write (null)	The direction of crossing that activates this threshold. *For the possible property values, see [Activation](#) in Property details.*
DwellTime	string	read-write (null)	The duration the sensor value must violate the threshold before the threshold is activated.
Reading }	number	read-write (null)	The threshold value.
LowerFatal {	object		The value at which the reading is below normal range and fatal.
Activation	string (enum)	read-write (null)	The direction of crossing that activates this threshold. *For the possible property values, see [Activation](#) in Property details.*
DwellTime	string	read-write (null)	The duration the sensor value must violate the threshold before the threshold is activated.
Reading }	number	read-write (null)	The threshold value.
UpperCaution {	object		The value at which the reading is above normal range.
Activation	string (enum)	read-write (null)	The direction of crossing that activates this threshold. *For the possible property values, see [Activation](#) in Property details.*
DwellTime	string	read-write (null)	The duration the sensor value must violate the threshold before the threshold is activated.
Reading }	number	read-write (null)	The threshold value.
UpperCautionUser *(v1.2+)* {	object		The value at which the reading is above normal range.
Activation	string (enum)	read-write (null)	The direction of crossing that activates this threshold. *For the possible property values, see [Activation](#) in Property details.*
DwellTime	string	read-write	The duration the sensor value must violate the

		(null)	threshold before the threshold is activated.
Reading }	number	read-write (null)	The threshold value.
UpperCritical {	object		The value at which the reading is above normal range but not yet fatal.
Activation	string (enum)	read-write (null)	The direction of crossing that activates this threshold. *For the possible property values, see Activation in Property details.*
DwellTime	string	read-write (null)	The duration the sensor value must violate the threshold before the threshold is activated.
Reading }	number	read-write (null)	The threshold value.
UpperCriticalUser *(v1.2+)* **{**	object		The value at which the reading is above normal range but not yet fatal.
Activation	string (enum)	read-write (null)	The direction of crossing that activates this threshold. *For the possible property values, see Activation in Property details.*
DwellTime	string	read-write (null)	The duration the sensor value must violate the threshold before the threshold is activated.
Reading }	number	read-write (null)	The threshold value.
UpperFatal {	object		The value at which the reading is above normal range and fatal.
Activation	string (enum)	read-write (null)	The direction of crossing that activates this threshold. *For the possible property values, see Activation in Property details.*
DwellTime	string	read-write (null)	The duration the sensor value must violate the threshold before the threshold is activated.
Reading } }	number	read-write (null)	The threshold value.
VoltageType	string (enum)	read-only (null)	The voltage type for this sensor. *For the possible property values, see VoltageType in Property details.*

Actions

ResetMetrics

Resets metrics related to this sensor.

Action URI: {Base URI of target resource}/Actions/Sensor.ResetMetrics

Perform the action using a POST to the specific Action URI for this resource. This action takes no parameters.

Property details

Activation:

The direction of crossing that activates this threshold.

string	Description
Decreasing	Value decreases below the threshold.
Either	Value crosses the threshold in either direction.
Increasing	Value increases above the threshold.

ElectricalContext:

The combination of current-carrying conductors.

string	Description
Line1	The circuits that share the L1 current-carrying conductor.
Line1ToLine2	The circuit formed by L1 and L2 current-carrying conductors.
Line1ToNeutral	The circuit formed by L1 and neutral current-carrying conductors.
Line1ToNeutralAndL1L2	The circuit formed by L1, L2, and neutral current-carrying conductors.
Line2	The circuits that share the L2 current-carrying conductor.
Line2ToLine3	The circuit formed by L2 and L3 current-carrying conductors.
Line2ToNeutral	The circuit formed by L2 and neutral current-carrying conductors.
Line2ToNeutralAndL1L2	The circuit formed by L1, L2, and Neutral current-carrying conductors.
Line2ToNeutralAndL2L3	The circuits formed by L2, L3, and neutral current-carrying conductors.
Line3	The circuits that share the L3 current-carrying conductor.
Line3ToLine1	The circuit formed by L3 and L1 current-carrying conductors.
Line3ToNeutral	The circuit formed by L3 and neutral current-carrying conductors.
Line3ToNeutralAndL3L1	The circuit formed by L3, L1, and neutral current-carrying conductors.
LineToLine	The circuit formed by two current-carrying conductors.
LineToNeutral	The circuit formed by a line and neutral current-carrying conductor.
Neutral	The grounded current-carrying return circuit of current-carrying conductors.
Total	The circuit formed by all current-carrying conductors.

Implementation:

The implementation of the sensor.

string	Description
PhysicalSensor	The reading is acquired from a physical sensor.
Reported	The reading is obtained from software or a device.
Synthesized	The reading is obtained by applying a calculation on one or more properties. The calculation is not provided.

PhysicalContext:

The area or device to which this sensor measurement applies.

string	Description

Accelerator	An accelerator.
ACInput	An AC input.
ACMaintenanceBypassInput	An AC maintenance bypass input.
ACOutput	An AC output.
ACStaticBypassInput	An AC static bypass input.
ACUtilityInput	An AC utility input.
ASIC	An ASIC device, such as a networking chip or chipset component.
Back	The back of the chassis.
Backplane	A backplane within the chassis.
Chassis	The entire chassis.
ComputeBay	Within a compute bay.
CoolingSubsystem	The entire cooling, or air and liquid, subsystem.
CPU	A processor (CPU).
CPUSubsystem	The entire processor (CPU) subsystem.
DCBus	A DC bus.
Exhaust	The air exhaust point or points or region of the chassis.
ExpansionBay	Within an expansion bay.
Fan	A fan.
FPGA	An FPGA.
Front	The front of the chassis.
GPU	A graphics processor (GPU).
GPUSubsystem	The entire graphics processor (GPU) subsystem.
Intake	The air intake point or points or region of the chassis.
LiquidInlet	The liquid inlet point of the chassis.
LiquidOutlet	The liquid outlet point of the chassis.
Lower	The lower portion of the chassis.
Memory	A memory device.
MemorySubsystem	The entire memory subsystem.
Motor	A motor.
NetworkBay	Within a networking bay.
NetworkingDevice	A networking device.
PowerSubsystem	The entire power subsystem.
PowerSupply	A power supply.

PowerSupplyBay	Within a power supply bay.
Rectifier	A rectifier device.
Room	The room.
StorageBay	Within a storage bay.
StorageDevice	A storage device.
SystemBoard	The system board (PCB).
Transformer	A transformer.
Upper	The upper portion of the chassis.
VoltageRegulator	A voltage regulator device.

PhysicalSubContext:

The usage or location within a device to which this sensor measurement applies.

string	Description
Input	The input.
Output	The output.

ReadingType:

The type of sensor.

string	Description
AirFlow	Airflow.
Altitude	Altitude.
AngularPosition *(v1.2+)*	Position (Angular).
Barometric	Barometric pressure.
Current	Current.
EnergyJoules	Energy (Joules).
EnergykWh	Energy (kWh).
Force *(v1.2+)*	Force or strain.
Frequency	Frequency.
Humidity	Relative Humidity.
LinearPosition *(v1.2+)*	Position (Linear).
LiquidFlow	Liquid flow.
LiquidLevel	Liquid level.
Luminosity *(v1.2+)*	Luminosity.
Percent *(v1.1+)*	Percent.
Power	Power.

Pressure	Pressure.
Rotational	Rotational.
Sound *(v1.2+)*	Sound.
Temperature	Temperature.
Velocity *(v1.2+)*	Velocity.
Voltage	Voltage (AC or DC).
Weight *(v1.2+)*	Weight.

VoltageType:

The voltage type for this sensor.

string	Description
AC	Alternating current.
DC	Direct current.

Example response

```json
{
    "@odata.type": "#Sensor.v1_1_0.Sensor",
    "Id": "CabinetTemp",
    "Name": "Rack Temperature",
    "ReadingType": "Temperature",
    "ReadingTime": "2019-12-25T04:14:33+06:00",
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "Reading": 31.6,
    "ReadingUnits": "C",
    "ReadingRangeMin": 0,
    "ReadingRangeMax": 70,
    "Accuracy": 0.25,
    "Precision": 1,
    "SensingInterval": "PT3S",
    "PhysicalContext": "Chassis",
    "Thresholds": {
        "UpperCritical": {
            "Reading": 40,
            "Activation": "Increasing"
        },
        "UpperCaution": {
            "Reading": 35,
            "Activation": "Increasing"
        },
        "LowerCaution": {
            "Reading": 10,
            "Activation": "Increasing"
        }
    },
    "Oem": {},
    "@odata.id": "/redfish/v1/Chassis/1/Sensors/CabinetTemp"
}
```

Redfish documentation generator

This document was created using the Redfish Documentation Generator utility, which uses the contents of the Redfish schema files (in JSON schema format) to automatically generate the bulk of the text. The source code for the utility is available for download at the DMTF's Github repository located at http://www.github.com/DMTF/Redfish-Tools.