Intel® Rack Scale Architecture Rack Management Module

API Specification

Software v1.1

September 2015

Revision 003

Information in this document is provided in connection with Intel® products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, lifesaving, or life sustaining applications. Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

This document contains information on products in the design phase of development. Do not finalize a design with this information. Revised information will be published when the product is available. Verify with your local sales office that you have the latest datasheet before finalizing a design.

This document and the software described in it is furnished under license and may only be used or copied in accordance with the terms of the license. The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document.

Contents

Figures

Tables

Revision History

Revision	Description	Date
001	Initial release.	July 31, 2015
002	Minor updates.	August 27, 2015
003	Minor updates	September 29, 2015

§

1 Introduction

1.1 Scope

This document defines Intel® Rack Scale Architecture Rack Management Module (RMM) REST API.

1.2 Intended audience

The intended audience for this document is Intel® Rack Scale Architecture developers who are responsible for defining and implementing the rack management stack.

1.3 Terminology

Table 1 Terminology

Term	Definition
BMC	Baseboard Management Controller
CIMI	Cloud Infrastructure Management Interface
POD	A physical collection of multiple racks
PODM	POD Manager aka RCPM
RCPM	Rack Controller Plane Manager aka PODM
SMC	Shared Memory Controller. Pooled and Shared NVM Controller: Implementation may be with multiple devices
TMC	Tray Management Controller
RUID	Rack unique id
PZ_RUID	Power zone rack unique id
RMC	Rack Management Controller
RMM	Rack Management Module
RUID	Rack unique id
PZ_RUID	Power zone rack unique id
TZ_RUID	Thermal/Cooling zone rack unique id
ZUID	Zone unique id
P_ZUID	PSU zone unique id
F_ZUID	FAN zone unique id
CM	Control Module
MBP	Management Backplane
PUID	POD wide unique id. For example, RackPUID
DCUID	Datacenter wide unique id
DR_UUID	Drawer unique id
DR_RUID	Drawer rack unique id
Int64	Integer 64 bits

1.4 References

Table 2 Reference documents

Number	Title	Location
332971	Intel® Rack Scale Architecture Getting Started Guide	*https://github.com/01org/RSA*
332868	Intel® Rack Scale Architecture GAMI API Specification	*https://github.com/01org/RSA*
332869	Intel® Rack Scale Architecture Pod Manager API Specification	*https://github.com/01org/RSA*
332870	Intel® Rack Scale Architecture Pod Manager Release Notes	*https://github.com/01org/RSA*

332871	Intel® Rack Scale Architecture Pod Manager User Guide	*https://github.com/01org/RSA*
332872	Intel® Rack Scale Architecture PSME Release Notes	*https://github.com/01org/RSA*
332873	Intel® Rack Scale Architecture PSME API Specification	*https://github.com/01org/RSA*
332874	Intel® Rack Scale Architecture PSME User Guide	*https://github.com/01org/RSA*
332875	Intel® Rack Scale Architecture RMM Installation Guide	*https://github.com/01org/RSA*
332876	Intel® Rack Scale Architecture RMM Release Notes	*https://github.com/01org/RSA*
332877	Intel® Rack Scale Architecture RMM API Specification	This document
332878	Intel® Rack Scale Architecture Storage Services API Specification	*https://github.com/01org/RSA*
	Scalable Platforms Management	*http://dmtf.org/standards/wip*
	Intel® Rack Scale Architecture System Manageability Architecture Specification	
	Intel® Rack Scale Architecture Platform Design Guide	
	Intel® Rack Scale Architecture Manageability Model and Interface v.0.3.1	
	China Rack and Bulldog Creek-Alta Rack Management Module (RMM) Requirements v0.8	
	Intel® Rack Scale Architecture-C Manageability Requirements v.01	
	Scorpio 2.0 spec.	
	Bulldog Creek Thermal Control	
	DMTF – DSP 1013 FAN Profile, DSP 1015 Power Supply Profile	
	Redfish Spec. v0.9	
	RESTful Conventions for RSA Manageability	

1.5 Notes and symbol convention

- The following notation is used in JSON serialization descriptions:
- Values in italics indicate data types instead of literal values.
- Characters are appended to items to indicate cardinality:
  - "?" (0 or 1)
  - "*" (0 or more)
  - "+" (1 or more)
- Vertical bars, "|", denote choice. For example, "a|b" means a choice between "a" and "b".
- Parentheses, "(" and ")", are used to indicate the scope of the operators "?", "*", "+" and "|".
- Ellipses (i.e., "...") indicate points of extensibility. Note that the lack of an ellipses does not mean no extensibility point exists, rather just that it is not explicitly called out - usually for the sake of 356 brevity.

1.6 JSON serialization convention

An object is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by : (colon) and the name/value pairs are separated by , (comma).

An array is an ordered collection of values. An array begins with [(left bracket) and ends with] (right bracket). Values are separated by , (comma).

A value can be a string in double quotes; or a number; or true or false or null; or an object or an array. These structures can be nested.

A string is a sequence of zero or more Unicode characters, wrapped in double quotes, using backslash escapes. A character is represented as a single character string. A string is very much like a C or Java string.

A number is very much like a C or Java number, except that the octal and hexadecimal formats are not used.

1.7 HTTP response codes

Table 3 HTTP response documents

Status	Description
200	Success
201	Created
202	Accepted (for asynchronous operations)
400	Resource invalid (improperly formatted request)
401	Unauthorized (failed authentication)
404	Resource not found
406	Not acceptable (i.e., invalid Accept Header)
500	Application Error

1.8 Text

The options are:

- BumpyCase
- camelCase
- lowercase
- UPPERCASE

Some existing conventions are:

- The “CaSe” convention for the following:
  - URLs – lower case.
  - Post/Patch Fields/json body data – camelCase.
  - Response Data– camelCase.
- The Redfish convention for the following:
  - URLs – BumpyCase.
  - Post/Patch Fields/json body data – ?
  - Response Data– ?

The Intel® Rack Scale Architecture convention shall be:

- URLs – camelCase.
- Post/Patch Fields/json body data – camelCase.
- Response Data- BumpyCase.

The convention follows the overall guidance that: 1) camelCase for keys (just like modern-style variable names) and BumpyCase for values (possibly read by humans).

1.9 Acronyms

The options are:

- Capitalize all (for example, setLANConfig, getFRUInfo).
- Capitalize first letter (for example, setLanConfig, getFruInfo).

Some existing conventions are:

- Microsoft .Net has a special >2 letter rule for mixed case, however, the don’t mix 2 letter acronyms (for example, getIOCounter). (However, most disagree with this and would prefer a consistant mixed case for all words including 2 letter acronyms.)

The Intel® Rack Scale Architecture convention shall be:

- For 3+ letter acronyms, capitalize first letter.
- For 2 letter acronyms, capitalize all.

1.10 Abbreiviations

The options are:

- Treat as words (for example, Ok).
- Treat as acronyms (for example, OK).

Some existing conventions are:

- ?

Commented [NP1]: ?

The Intel® Rack Scale Architecture convention shall be:

- Treat abbreviations as words.

Note: Since there is distinction between keys (camelCase) and values (BumpyCase), this means that ID as a key would be “id” and OK as a value would be “Ok”.

1.11 UUID and LID

The UUID is used to indicate one component instance within its lifecycle.

- The component can be either physical component (for example, RMM, MBP, PSU, FAN) or virtual component (for example, thermal zone, power zone).
- The component instance is created while the component is inserted into rack system and/or is detected by RMM.
- The component instance will be destroyed while the component is removed/reset/shutdown.
- The special case is RMM reset/shutdown will trigger the destroy of all its owned component instance and release of all these instance UUIDs.
- The external software can use instance UUID to track one component instance only under that instance’s lifecycle, but if this component is removed, or replaced or exchanged to other slot. The original instance will be destroyed and the instance UUID will be invalid.

LID is used as one *unchangeable URI* to refer to the component installed into the specific location of the rack system in against of the instance UUID which is only valid during that instance’s lifecycle. LID equals to the alias of the instance UUID while the component instance lives.

1.12 Available actions (RF 5.1)

The AvailableActions array contains a list of actions supported by resources when used in the schema, and by this particular resource when used in a resource. If a resource indicates an action and its capabilities using the AvailableActions property, the resource shall support the action and the components listed in the resource. This may be a subset of those indicated in the Redfish Schema for that resource. Other semantics are defined in the schema; this property is in the common section of the Redfish Schema.

For instance, if a schema has an AvailableActions section such as this example:

```
{
    "AvailableActions": {
        "type": "array",
        "items": [
            {
                "type": "object",
                "properties": {
                    "Action": {
                        "type": "string",
                        "enum": [
                            "reset"
                        ]
                    },
                    "Capabilities": {
                        "type": "array",
                        "items": {
                            "type": "object",
                            "additionalProperties": false,
                            "properties": {
                                "PropertyName": {
                                    "type": "string",
                                    "enum": [
                                        "resetType"
                                    ]
                                },
                                "AllowableValues": {
                                    "type": "array",
                                    "items": {
                                        "type": "string",
                                        "enum": [
                                            "On",
                                            "Off",
                                            "reset",
                                            "coldReset",
                                            "nmi",
                                            "gracefulShutdown"
                                        ]
                                    }
                                }
                            }
                        }
                    }
                }
            }
        ]
    }
}

And the resource has an AvailableActions property such as this:

{
    "AvailableActions": [
        {
            "Action": "reset",
            "Capabilities": [
                {
                    "PropertyName": "resetType",
                    "AllowableValues": [
                        "on",
                        "off",
                        "reset"
                    ]
                }
            ]
        }
    ]
}
```

Then the following would represent a possible request body for the action:

```
{
    "Action": "reset",
    "resetType": "on"
}
```

§

2 Overview

This section shows general structure of the RMM REST API.

2.1 API structure and relation

Table 4 Resources and URI

Resource	URI
RMM API entry point	/v1/rack /v1/rack/rf_event
All power zones in a rack	/v1/rack/powerzones /v1/rack/powerzones/rf_event
A specific power zone on a rack	/v1/rack/powerzones/{PZ_RUID } /v1/rack/powerzones/{PZ_RUID}/rf_event
All PSUs on a specific power zone	/v1/rack/powerzone/{ PZ_RUID }/psus /v1/rack/powerzone/{ PZ_RUID }/psus/rf_event
A specific psu in a specific power zone	/v1/rack/powerzone/{ PZ_RUID }/psus/{P_ZUID} /v1/rack/powerzones/{ PZ_RUID }/psus/{P_ZUID }/rf_event
All thermal zones in a rack	/v1/rack/thermalzones /v1/rack/thermalzones/rf_event
A specific thermal zone on a rack	/v1/rack/thermalzones/{TZ_RUID } /v1/rack/thermalzones/{ TZ_RUID }/rf_event
All FANs on a specific thermal zone	/v1/rack/thermalzones/{TZ_RUID}/fans /v1/rack/thermalzones/{TZ_RUID}/fans/rf_event
A specific FAN in a specific thermal zone	/v1/rack/thermalzones/{ TZ_RUID }/fans/{F_ZUID} /v1/rack/thermalzones/{ TZ_RUID }/fans/{F_ZUID}/rf_event
All drawer(trays) in a rack	/v1/rack/drawers /v1/rack/drawers/rf_event
A specific drawer in a rack	/v1/rack/drawers/{DR_UUID} /v1/rack/drawers/{DR_UUID}/rf_event
All management backplanes in a rack	/v1/rack/mbps /v1/rack/mbps/rf_event
A specific management backplane in a rack	/v1/rack/mbps/{MBP_ZUID} /v1/rack/mbps/{MBP_ZUID}/rf_event

2.2 Rack management hierarchy and terminologies

Figure 1 and Table 5 illustrate and describe the terms used in the rack management hierarchy.

Figure 1 Rack management hierarchy model

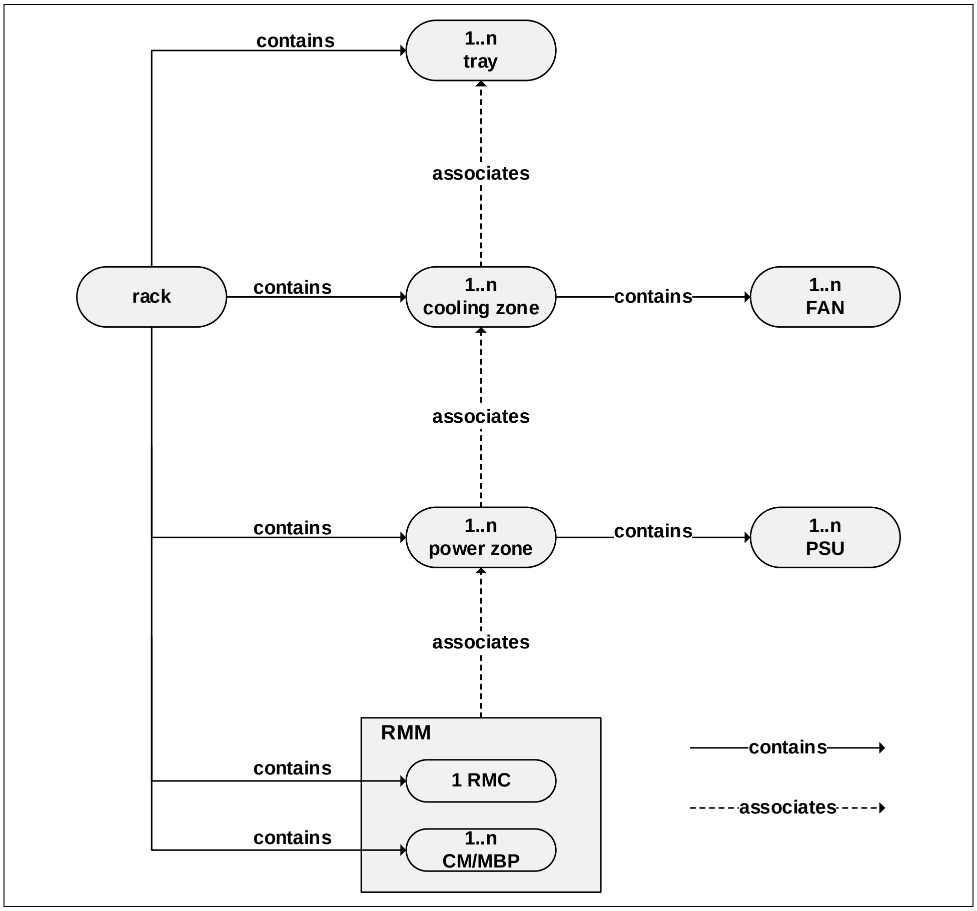

Table 5 Rack management terminologies

Term	Definition
Rack	Includes 1 or multiple Power Zones.
Power Zone	Includes a power shelf (or PSUs) and a number of trays powered by that power shelf. A Power Zone is one power management domain; the servers in a power zone share the same PSUs.
Thermal Zone	Includes thermal modules (fans) and a number of trays cooled by that thermal module. A thermal Zone is one thermal management domain; the servers in a thermal zone share the same thermal modules.
Drawer	Includes 1 or multiple server modules, and may also include the pooled resources.
RMM	Rack management module. The rack contains 1 RMM.
RMC	Rack management controller. The RMM contains 1 RMC.
CM or MBP	Controller Module or Management Backplane. The RMM contains 0..n CM/MBP.

§

3 Rack

3.1 Rack module entry point

The Rack Management Module entry point (which contains basic information of the rack management module) also provides top level resource collections of the rack.

Table 6 Rack module entry point information attributes

Name	Rack	
Type URI	/v1/rack	
Attribute	**Type**	**Description**
id	Int64	The index of rack within POD, generally. This index is defined as RackPUID(rack unique ID within POD). Default: 0, indicates no RackPUID is assigned. RackPUID must start as 1.
uuid	String	Unique identifier for the rack. UUID Pattern.
name	String	Human readable name for the rack.
description	String	**Editable**. The description of the rack.
createdDate	String	Timestamp when Rack was created (ISO 8601) ISO 8601: YYYY-MM-DDThh:mm:ss.SSS'Z' Year, Month, Date (UTC) 1134 Hour, Minute, Second 1135 Relative Time Zone Shift (Hour and Minute) Ex: "2013-01-10T01:09:57+08:00"
updatedDate	String	Timestamp when rack status was changed.
apiVersion	String	RMM REST API version, for example,. “v0.9”.
rackPUID	Int64	**Editable**. Rack Pod Unique ID. We will use the ordered Rack Index within the Pod. This ID is assigned by PODM while rack system registers to PODM.
podDCUID	String	**Editable**. Pod Datacenter Unique ID = Pod Fully Qualified Host Name. For example, pod1.jf5.intel.com
PODMAddress	String	**Editable**. Host name or IP address of PODM.
geoTag	String	**Editable**. Unique User Defined string for a Geo Location.
RackDCUID	String	PodDCUID-RackPUID. “none-0”: default value.
assetTag	String	**Editable**. String number of the asset tag of rack. Generally, this value is set by the rack user for inventory management purpose.
FRUInfo	FRUInfo	Rack FRU(Field Replacement Unit) Information. Generally, FRUInfo is set by the manufacture and normally not changeable.
availableActions	Action Array	The availableActions property and structure indicate the supported action by resource, the client can query the resource directly to determine the actions that are available as well as possible parameters.
links	uri-str	List of resource URL under current rack entry: “drawers” “powerzones” “thermalzones” “rf_event”
RMMPresent	Boolean	true or false.
rackSupportsDisaggregatedPowerCooling	Boolean	true or false.

pod	String	Indicates which Pod the rack belong to.
systemPresence	String	Each bit identify one drawer.
MBPPresence	String	Each bit identify one CM.

Table 7 Rack module FRU information attributes

Name	FRUInfo	
Attribute	**Type**	**Description**
serialNumber	String	The serial number of the device.
manufacturer	String	Manufacture information.
modelNumber	String	Model Information.
partNumber	String	The part number.
firmwareRevision	String	The firmware version.

3.1.1 JSON serialization

```
{
       “id”: int64,
       “uuid”: string,
       “name”: string,
       “description”: string,
       “createdDate”: string,
       “updatedDate”: string,
       “apiVersion”: string,
       “rackPUID”: int64,
       “PODMAddress”: string,
       “podDCUID”: string,
       “geoTag”: string,
       “rackDCUID”: string,
       “assetTag”: string,
       “FRUInfo”: {
               “serialNumber”: string,
               “manufacture”: string,
               “modelNumber”: string,
               “partNumber”: string,
               “firmwareRevision””: string,
               },
       “availableActions”: [],
       “links”: {
               “drawers”: uri-str,
               “powerzones”: uri-str,
               “thermalzones”: uri-str,
               “rf_event”: uri-str,
               },
       “RMMPresent”: Boolean,
       “rackSupportsDisaggregatedPowerCooling”: Boolean,
       “pod”: string,
       “systemPresence”:string,
       “MBPPresence”: string
}
```

3.1.2 Operations

This resource supports the GET, PUT, and POST operations.

3.1.2.1 GET

Method	URI	Description	Response code
GET	/v1/rack	List detail information of rack	Refer to Section 1.7 HTTP response codes

Request:

```
GET /v1/rack
Content-Type: application/json
```

Response:

```
{
       “id”: 1,
       “uuid”: “25892e17-80f6-415f-9c65-7395632f0223”,
       “name”: “RSA_Rack_23”,
       “description”: “Rack No. 23”,
       “createdDate”: “2014-09-10T01:09:26+08:00”,
       “updatedDate”: “2014-09-11T02:19:17+08:00”,
       “apiVersion”: “v0.9”,
       “rackPUID”: 1,
       “podDCUID”: ”pod1.jf5.intel.com”,
       “PODMAddress”: “podm1.pod1.jf5.intel.com”,
       “geoTag”: “Oregon-JF5”,
       “rackDCUID”: “pod1.jf5.intel.com-1”,
       “assetTag”: “”,
                “FRUInfo”: {
                “serialNumber”: “”,
                “manufacture”:””,
                “modelNumber”:””,
                “partNumber”: “”,
                “firmwareVersion”:””,
                },
       “availableActions”: [
                {
                      “action”: “reset”,
                      “capabilities”: [
                               {
                                                “propertyName”: “resetType”,
                                                “allowableValues”: [
                                                      “reset”
                                                ]
                               }
                      ]
                },
                {
                      “action”: “update”,
                      “capabilities”: [
                               {
                                                “propertyName”:
“componentType”,
                               },
                               {
                                                “propertyName”: “imageURL”,
                               }
                      ]
                },
                {
                        “action”: “loadFactoryDefault”,
                },
        ],
        “links”: {
                “drawers”: “http://localhost:8080/v1/rack/drawers/”,
                “powerzones”: “http://localhost:8080/v1/rack/powerzones/”,
                “thermalzones”: “http://localhost:8080/v1/rack/thermalzones/”,
                “mbps": "http://localhost:8080/v1/rack/mbps/”,
        },
        “RMMPresent”: true,
        “rackSupportsDisaggregatedPowerCooling”: false,
        “pod”: “1234567890-abcde”
}
```

3.1.2.2 PUT

Table 8 PUT operation: Update editable properties

Name	Editable Properties	
Attribute	**Type**	**Description**
description	String	**Optional.** Description of the system.
rackPUID	Int64	**Optional**. Rack Pod Unique ID. We will use the ordered Rack Index within the Pod.
podDCUID	String	**Optional**. Pod Datacenter Unique ID = Pod Fully Qualified Host Name. For example, pod1.jf5.intel.com.
geoTag	String	**Optional**. Unique User Defined string for a Geo Location.
PODMAddress	String	**Optional**. Hostname or IP address of PODM.
assetTag	String	**Optional**. String number of the asset tag of rack. Generally, this value is set by the rack user for inventory management purpose.

Method	**URI**	**Description**	**Response code**
PUT	/v1/rack	Update editable properties	Refer to Section 1.7 HTTP response codes

Request:

```
PUT /v1/rack
Content-Type: application/json
{
        “description”: “storage rack”,
        “rackPUID”: 1,
        “podDCUID”: ”pod2.jf5.intel.com”,
        “geoTag”: “Oregon-JF4”,
        “PODMAddress”: “podm2.pod1.jf5.intel.com”
}
```

Response:

```
{
        “id”: 1,
        “uuid”: “25892e17-80f6-415f-9c65-7395632f0223”,
        “name”: “RSA_Rack_23”,
        “description”: “storage rack”,
        “createdDate”: “2014-09-10T01:09:26+08:00”,
        “updatedDate”: “2014-09-11T02:19:17+08:00”,
        “apiVersion”: “rev 0.9”,
       “rackPUID”: 1,
       “podDCUID”: ”pod2.jf5.intel.com”,
       “PODMAddress”: “podm2.pod1.jf5.intel.com”,
       “geoTag”: “Oregon-JF4”,
       “rackDCUID”: “pod1.jf5.intel.com-1”,
       “assetTag”: “”,
                “FRUInfo”: {
                “serialNumber”: “”,
                “manufacture”:””,
                “modelNumber”:””,
                “partNumber”: “”,
                “firmwareVersion”:””,
                },
       “availableActions”: [
                {
                       “action”: “reset”,
                       “capabilities”: [
                               {
                                                “propertyName”: “resetType”,
                                                “allowableValues”: [
                                                      “service”,
                                                      “rmm”
                                                ]
                               }
                       ]
                },
                {
                       “action”: “update”,
                       “capabilities”: [
                               {
                                                “propertyName”:
“componentType”,
                               },
                               {
                                                “propertyName”: “imageURL”,
                               }
                       ]
                },
                {
                       “action”: “loadFactoryDefault”,
                },
       ],
       “links”: {
                “systems”: “http://localhost:8080/v1/rack/drawers/”,
                “powerzones”: “http://localhost:8080/v1/rack/powerzones/”,
                “thermalzones”: “http://localhost:8080/v1/rack/thermalzones/”,
                “rf_even”: “http://localhost:8080/v1/rack/rf_event/”,
                “mbps": "http://localhost:8080/v1/rack/mbps/”,
       },
       “RMMPresent”: True,
       “rackSupportsDisaggregatedPowerCooling”: False,
       “pod”: “1234567890-abcde”
}
```

3.1.2.3 POST

Table 9 Reset action

Name	Reset Action	
Attribute	**Type**	**Description**
action	String	Action type, “reset”.
resetType	String	Rest type: “service”: restart RMM services. “rmm”: reset RMM HW.

Method	URI	Description	Response code
POST	/v1/rack	Reset RMM or RMM service	Refer to section 1.7 HTTP response codes.

Request:

```
POST /v1/rack/
Content-Type: application/json
{
       “action”:  “reset”,
       “resetType”: “service”
}
```

Response:

Normal code: 202 Accepted. The request has been accepted for processing.

This operation does not return a response body.

Table 10 Update action

Name	Update Action	
Attribute	**Type**	**Description**
action	String	Action type, “udpate”.
componentType	String	The component of RMM to be upgraded: “rsa-rmm” – RMM base software. “rsa-rmm-fancontrol” – RMM FAN speed control component. “rsa-rmm-meta” – RMM meta package to install all rmm components.
Image	String	RMM component package content encoded with base64.

Method	URI	Description	Response code
POST	/v1/rack	Remotely update RMM components.	Success: 202 Failure: 406

Request:

```
POST /v1/rack
Content-Type: application/json
{
   “action”: “update”,
   “componentType”: “rsa-rmm-meta”,
   “image”: “XAXX111ddxx###.....”
}
```

Successful Response:

Response code: 202(Accepted).
Response body:

```
{
        “result”: “updating”,
        “new image rev”: “rev 0.9.2564”
}
```

The request has been accepted for processing. The client should query the status and firmware version to verify if the update is successful or not.

Failure Response:

Response code: 406(Not acceptable).

Response body:

```
{
        “result”: “unmatched version”,
}
 Or
{
        “result”: “invalid image”,
}
```

Table 11 LoadFactoryDefault action

Name	LoadFactoryDefault action	
Attribute	**Type**	**Description**
action	String	Action type, “LoadFactoryDefault”.

Method	**URI**	**Description**	**Response code**
POST	/v1/rack	Restore RMM configuration into factory default	Refer to section 1.7 HTTP response codes.

Request:

```
POST /v1/rack/
Content-Type: application/json
{
        “action”:  “loadFactoryDefault”
}
```

Response:

Normal code: 202 Accepted. The request has been accepted for processing. This operation does not return a response body.

§

4 Power

4.1 Power zone collections

Power Zone collection resource provides collection of to all power zones that belong to the rack.

Table 12 Power zone collection attributes

Name	Collections of power zone	
Type URI	/v1/rack/powerzones/	
Attribute	**Type**	**Description**
id	Int64	Index of power zone within one rack, equal to PZ_RUID. PZ_RUID stands for power zone unique index within rack. It starts from 1.
uuid	uid-str	Unique identifier for the power zone. UUID Pattern.
name	String	Human readable name for the power zone.
links	uri-str	*http://localhost:8080/v1/rack/powerzones/{PZ_RUID}*.

4.1.1 JSON serialization

```
{
       “powerzones”: [
                      {
                              “id”: int64,
                              “uuid”: uid-str,
                              “name”: string,
                              “links”: uri-str,
                      }, +
                ]
}
```

4.1.2 Operations

This resource supports the GET operation.

4.1.2.1 GET

Method	**URI**	**Description**	**Response code**
GET	/v1/rack/powerzones	List collection information for all power zones.	Refer to section 1.7 HTTP response codes.

Request:

```
GET /v1/rack/powerzones
Content-Type: application/json
```

Response:

```
{
       “powerzones”: [
                      {
                              “id”: 1,
                              “uuid”: “25892e17-80f6-415f-9c65-7395632e0223”,
                              “name”: “power zone 1”,
                      “links”: “http://localhost:8080/v1/rack/powerzones/1”
                      },
                      {
                               “id”: 2,
                               “uuid”: “25892e17-80f6-415f-9c65-7395632e0233”,
                               “name”: “power zone 2”,
                      “links”: “http://localhost:8080/v1/rack/powerzones/2”

                      },
       ]
}
```

4.2 Power zone

Table 13 Power zone attributes

Name	Power Zone	
Type URI	/v1/rack/powerzones/{PZ_RUID} or /v1/rack/powerzones/{powerzone_uuid} - alias resource URI.	
Attribute	**Type**	**Description**
id	Int64	Index of power zone within the rack, equal to PZ_RUID.
uuid	uid-str	Unique identifier for the power zone. UUID Pattern.
name	String	Human readable name for the computing system.
description	String	**Editable**. Description of the computing system.
createdDate	String	A time/date stamp for when the power zone is created.
updatedDate	String	A time/date stamp for when the power zone status is changed.
rackLocation	RSARack.RackLocation	Power zone Location.
presentPowerInput	Int64	The total power consumption of whole power zone, sum of trays’ power consumption.
presentPowerOutput	Int64	The total power production of whole power zone, sum of PSUs’ output.
maxRatedPowerCapacity	Int64	
maxPSUsSupported	Int64	
presence	String	Indicates the aggregated compute module presence information. Aggregated compute module presence format: Length of string indicate total slot of compute module component. For each byte the string: “1” means present. “0” means not present.
numberOfPSUsPresent	Int64	Indicates the aggregated PSU presence information. Aggregated PSU presence format: Length of string indicate total slot of PSU component. For each byte the string: “1” means present. “0” means not present.

Table 14 Power zone location attributes

Name	RackLocation	
Attribute	**Type**	**Description**
rackUnits	Int64	RSARack.Units: indicates the rack unit type. 0---RU: rack unit, each RU 44.5mm. 1---OU: openU, each OU 48mm.
uLocation	Int64	The index of the top-most U of the component, from top to bottom(1..MAXIMUM). “0” indicates not available.
uHeight	Int64	The height of managed zone. For example, 8 for 8U, 16 for 16U.
xLocation	Int64	The horizontal location within uLocation, from left to right(1.. MAXIMUM). “0” indicates not available.

4.2.1 JSON serialization:

JSON media type: application/json

```
{
       “id”: int64,
       “uuid”: uid-str,
       “name”: string,
       “description”: string,
       “createdDate”: string,
       ”updatedDate”: string,
       “rackLocation”: {
               “rackUnits” : int64,
       “xLocation”: int64,
               “uLocation” : int64,
               “uHeight”: int64,
       },
       “presentPowerInput”: int64,
       “presentPowerOutput”: int64,
       “maxRatePowerCapacity”: int64,
       “maxPSUsSupported”: int64,
       “numberOfPSUsPresent”: int64
}
```

4.2.2 Operations

This resource supports GET and PUT operations.

4.2.2.1 GET

Method	URI	Description	Response code
GET	/v1/rack/powerzones/{PZ_RUID}	List detail information of specific power zone.	Refer to section 1.7 HTTP response codes.

Request:

```
GET /v1/rack/powerzones/{PZ_RUID}
Content-Type: application/json
```

Response:

```
{
               “id”: 1,
               “uuid”: “25892e17-80f6-415f-9c65-7395632e0223”,
               “name”: “power zone 1”,
               “description”: “power zone 1”,
               “createdDate”: “2014-09-11T01:15:16+08:00”,
               “updatedDate”: “2014-09-11T01:15:16+08:00”,
               “rackLocation”: {
                      “rackunits” : 0,
           “xLocation” : 0,
                      “uLocation” : 1,
                      “uHeight”: 8,
               },
               “presentPowerInput”: 2700,
               “presentPowerOutput”: 3200,
               “maxRatePowerCapacity”: 5000,
               “maxPSUsSupported”: 6,
               “numberOfPSUsPresent”: 6
}
```

4.2.2.2 PUT

Table 15 Update editable properties of the specific system

Name	Editable Properties	
Attribute	**Type**	**Description**
description	String	**Optional**. Description of the power zone.

Method	**URI**	**Description**	**Response code**
PUT	/v1/rack/powerzones/{PZ_RUID}	Update editable properties of the specific power zone.	Refer to section 1.7 HTTP response codes.

Request:

```
PUT /v1/rack/powerzones/{PZ_RUID}
Content-Type: application/json
{
       “description”: “rack power zone 1”,
}
```

Response:

```
{
                “id”: 1,
                “uuid”: “25892e17-80f6-415f-9c65-7395632e0223”,
                “name”: “power zone 1”,
                “description”: “rack power zone 1”,
                “createdDate”: “2014-09-11T01:15:16+08:00”,
                “updatedDate”: “2014-09-11T11:15:16+08:00”,
                “rackLocation”: {
                       “rackUnits” : 0,
            “xLocation” : 0,
                       “uLocation” : 1,
                       “uHeight”: 8,
                },
                “presentPowerInput”: 2700,
                “presentPowerOutput”: 3200,
                “maxRatePowerCapacity”: 5000,
                “maxPSUsSupported”: 6,
                “numberOfPSUsPresent”: 6
}
```

4.3 Power supply unit collections

The Power Supply Unit Collections resource provides a list of all Power Supply Units that belong to a specific power zone.

Table 16 PSU collection attributes

Name	PSU collections	
Type URI	/v1/rack/powerzones/{PZ_RUID}/psus/	
Attribute	**Type**	**Description**
id	Int64	The index of the power supply unit within a specific power zone. The index equals P_ZUID. P_ZUID stands for the power supply unit unique index within a power zone. This index starts from 1 (Left to right and bottom-to-top).
uuid	uid-str	Unique identifier for the power supply unit. UUID Pattern.
name	String	Human readable name for the power supply unit.
links	uri-str	*http://localhost:8080/v1/rack/powerzones/{PZ_RUID}/psus/{P_ZUID}*.

4.3.1 JSON serialization

JSON media type: application/json

```
{
       “psus”: [
                      {
                              “id”: int64,
                              “uuid”: uid-str,
                              “name”: string,
                              “links”: uri-str,
                      }, +
               ]
}
```

4.3.2 Operations

This resource supports the GET operation.

4.3.2.1 GET

Method	**URI**	**Description**	**Response code**
GET	/v1/rack/powerzones/{PZ_RUID}/psus	List PSU collection of specific power zone.	Refer to section 1.7 HTTP response codes.

Request:

```
GET /v1/rack/powerzones/{PZ_RUID}/psus
Content-Type: application/json
```

Response:

```
{
       “psus”: [
                      {
                              “id”: 1,
                              “uuid”: “25892e17-80f6-415f-9c65-7395632d0223”,
                              “name”: “PSU 1”,
                       “links”:
“http://localhost:8080/v1/rack/powerzones/{PZ_RUID}/psus/1”
```

```
                      },
                      {
                              “id”: 2,
                              “uuid”: “25892e17-80f6-415f-9c65-7395632d0233”,
                              “name”: “PSU 2”,
                      “links”:
“http://localhost:8080/v1/rack/powerzones/{PZ_RUID}/psus/2”
                      },
       ]
}
```

4.4 Power supply unit (PSU)

Table 17 PSU collections attributes

Name	PSU	
Type URI	/v1/rack/powerzones/{PZ_RUID }/psus/{P_ZUID} or /v1/rack/powerzones/{PZ_RUID }/psus/{psu_uuid} - alias resource URI	
Attribute	**Type**	**Description**
id	Int64	The index of power supply unit. The index number equals to P_ZUID.
uuid	uid-str	Unique identifier for the power supply unit. UUID Pattern.
name	String	Human readable name for the power supply unit.
description	String	**Editable**. Description of the power supply unit.
createdDate	String	A time/date stamp for when the PSU created status.
updatedDate	String	A time/date stamp for when the PSU last changed status.
enabledState	RSARack.PSUstate	**Editable**. The enabled state of the power supply unit. This definition is derived from CIM_PowerSupply.EnabledState defined in DMTF Power Supply Unit Profile. Supported states: 2: Enabled. 3: Disabled. 5: Not applicable. 6: Enabled but offline (for redundancy design).
redundancySet	TBD	**Editable**. Reserved, the PSU redundancy is not defined in BDC-A.
healthCode	Int64	Health status of power supply unit. This definition is derived from CIM_PowerSupply.healthState defined in DMTF Power Supply Unit Profile. Supported states: 0: Unknown. 5: Health. 10: Degraded/warning. 25: Critical failure.
presentPowerInput	Int64	Input of power supply unit, using watts as unit.
presentPowerOutput	Int64	Output of power supply unit, using watts as unit.
maxRatedPowerCapacity	Int64	
availableActions	Action Array	RF 5.1 AvailableActions.
rackLocation	RackLocation	
assetTag	String	**Editable**. String number of the asset tag of specific power supply unit. Generally, this value is set by the power supply user for inventory management purpose.
FRUInfo	RSARack.FRUInfo	FRU Information of the power supply unit.

Commented [NP2]: tbd

4.4.1 JSON serialization:

JSON media type: application/json

```
{
       “id”: int64,
       “uuid”: uid-str,
       “name”: string,
       “description”: string,
       “createdDate”: string,
       “updatedDate”: string,
       “enabledState”: PSUState,
       “redundancySet”: TBD,
       “healthCode”: int64,
       “presentPowerInput”: int64,
       “presentPowerOutput”: int64,
       “maxRatedPowerCapacity”: int64,
       “assetTag”: string,
       “rackLocation”: {
                “rackunits” : int64,
      “xLocation” : int64,
                “uLocation” : int64,
                “uHeight”: int64,
       },
       “FRUInfo” : {
                “serialNumber”: string,
                “manufacture”: string,
                “modelNumber”: string,
                “partNumber”: string,
                “firmwareRevision”: string
       },
       “availableActions”: [
                      {
                              “action”: “requestStateChange”,
                              “capabilities”: [
                                                {
                                                      “propertyName”:
“enabledState”,
                                                      “allowableValues”: [
                                                            2,
                                                            3,
                                                            6,
                                                      ]
                                                },
                              ]
                      },
                ]
}
```

Commented [NP3]: tbd

4.4.2 Operations

This resource supports GET, PUT, and POST operations.

4.4.2.1 GET

Method	URI	Description	Response code
GET	/v1/rack/powerzones/{PZ_RUID}/psus/{P_ZUID}	Query power supply unit information.	Refer to section 1.7 HTTP response codes.

Request:

```
GET /v1/rack/powerzones/{PZ_RUID}/psus/{P_ZUID}
Content-Type: application/json
```

Response:

```
{
                “id”: 1,
                “uuid”: “25892e17-80f6-415f-9c65-7395632d0223”,
                “name”: “PSU 1”,
                “description”: “PSU 1”,
                “createdDate”: “2014-09-11T01:15:16+08:00”,
                “updatedDate”: “2014-09-15T03:12:16+08:00”,
                “EnabledState”: 2,
                “redundancySet”: TBD,
                “healthCode”: 5,
                “presentPowerInput”: 200,
                “presentPowerOutput”: 200,
                “maxRatedPowerCapacity”: 300
                “assetTag”: “”,
                “rackLocation”: {
                       “rackunits” : 0,
            “xLocation” : 0,
                       “uLocation” : 1,
                       “uHeight”: 8,
                },
                “FRUInfo”: {
                       “serialNumber”: “”,
                       “manufacturer”: “”,
                       “modelNumber”: “”,
                       “partNumber”: “”,
                       “firmwareRevision”: “”,
                }
       “availableActions”: [
                       {
                               “action”: “requestStateChange”,
                               “capabilities”: [
                                                {
                                                      “propertyName”:
“enabledState”,
                                                      “allowableValues”: [
                                                            2,
                                                            3,
                                                            6,
                                                      ]
                                                },
                               ]
                       },
       ]
}
```

Commented [NP4]: tbd

4.4.2.2 PUT

Table 18 PUT operation: Update editable properties of specific PSU

Name	Editable Properties	
Attribute	**Type**	**Description**
description	String	**Optional**. Description of the power supply unit.
assetTag	String	**Optional**. String number of the asset tag of specific power supply unit. Generally, this value is set by the power supply user for inventory management purpose.

Method	**URI**	**Description**	**Response code**
PUT	/v1/rack/powerzones/{PZ_RUID}/psus/{P_ZUID}	Update editable properties of specific power supply unit.	Refer to section 1.7 HTTP response codes.

Request:

```
PUT /v1/rack/powerzones/{PZ_RUID}/psus/{P_ZUID}
Content-Type: application/json
{
       “description”: “power supply unit 1”,
       “assetTag”: “1234567890”
}
```

Response:

```
{
                “id”: 1,
                “uuid”: “25892e17-80f6-415f-9c65-7395632d0223”,
                “name”: “PSU 1”,
                “description”: “PSU 1”,
                “createdDate”: “2014-09-11T01:15:16+08:00”,
                “updatedDate”: “2014-09-15T03:12:16+08:00”,
                “enabledState”: 2,
                “redundancySet”: TBD,
                “healthCode”: 5,
                “presentPowerInput”: 200,
                “presentPowerOutput”: 200,
                “maxRatedPowerCapacity”: 300
                “assetTag”: “1234567890”,
                “rackLocation”: {
                       “rackunits” : 0,
            “xLocation” : 0,
                       “uLocation” : 1,
                       “uHeight”: 8,
                },
                “FRUInfo”: {
                       “serialNumber”: “”,
                       “manufacturer”: “”,
                       “modelNumber”: “”,
                       “partNumber”: “”,
                       “firmwareRevision”: “”,
                }
       “availableActions”: [
                       {
                               “action”: “requestStateChange”,
                               “capabilities”: [
```

Commented [NP5]: TBD

```
                                                {
                                                      “propertyName”:
“enabledState”,
                                                      “allowableValues”: [
                                                            2,
                                                            3,
                                                            6,
                                                      ]
                                                },
                               ]
                      },
       ]
}
```

4.4.2.3 POST

Table 19 RequestStateChange action

Name	RequestStateChange Action	
Attribute	**Type**	**Description**
action	String	Action type. “RequestStateChange”
enabledState	Int64	The desired EnabledState. Allowable values: Supported states: 2: Enabled. 3: Disabled. 6: Enabled but offline (for redundancy design).

Method	**URI**	**Description**	**Response code**
POST	/v1/rack/powerzones/{PZ_RUID}/psus/{P_ZUID}	Request enabledState change for a specific PSU.	Refer to section 1.7 HTTP response codes.

Request:

```
POST /v1/rack/powerzones/{PZ_RUID }/psus/{P_ZUID}
Content-Type: application/json
{
       “action”: “requestStateChange”,
       “enabledState”: 2
}
```

Response:

Normal code: 202 Accepted. The request has been accepted for processing. It is expected that the client will use “GET /v1/rack/powerzones/{PZ_RUID }/psus/{P_ZUID }” to query the processing status later. This operation does not return a response body.

§

5 Thermal

5.1 Thermal Zone Collections

The Thermal Zone Collection resource provides a collection of to all Thermal zones that belong to the rack.

Table 20 Thermal zone collections attributes

Name	Collections of thermal zone	
Type URI	/v1/rack/thermalzones/	
Attribute	**Type**	**Description**
id	Int64	Index of thermal zone within one rack, equal to TZ_RUID. TZ_RUID stands for thermal zone unique index within rack. It starts from 1.
uuid	uid-str	Unique identifier for the thermal zone. UUID Pattern.
name	String	Human readable name for the thermalzone.
links	uri-str	*http://localhost:8080/v1/rack/thermalzones/{TZ_RUID}*.

5.1.1 JSON serialization

```
{
       “thermalzones”: [
                      {
                               “id”: int64,
                               “uuid”: uid-str,
                               “name”: string,
                               “links”: uri-str,
                      }, +
                ]
}
```

5.1.2 Operations

This resource supports the GET operation.

5.1.2.1 GET

Method	**URI**	**Description**	**Response code**
GET	/v1/rack/thermalzones	List collection information for all thermal zones.	Refer to section 1.7 HTTP response codes.

Request:

```
GET /v1/rack/thermalzones
Content-Type: application/json
```

Response:

```
{
       “thermalzones”: [
                      {
                               “id”: 1,
                               “uuid”: “25892e17-80f6-415f-9c65-7395632f0223”,
                               “name”: “thermal zone 1”,
               “links”: “http://localhost:8080/v1/rack/thermalzones/1”

                    },
                    {
                            “id”: 2,
                            “uuid”: “25892e17-80f6-415f-9c65-7395632f0224”,
                            “name”: “thermalzone 2”,
                      “links”:
“http://localhost:8080/v1/rack/thermalzones/2”
                    }
       ]
}
```

5.2 Thermal Zone

Table 21 Thermal zone attributes

Name	Thermal Zone	

Type URI	/v1/rack/thermalzones/{TZ_RUID}
	or
	/v1/rack/thermalzones/{thermalzone_uuid}	

Attribute	**Type**	**Description**
id	Int64	Index of thermalzone within one rack, equal to TZ_RUID. TZ_RUID stands for thermalzone unique index within rack. It starts from 1.
uuid	uid-str	Unique identifier for the thermalzone. UUID Pattern.
name	String	Human readable name for the thermalzone.
description	String	**Editable.** Description of the thermalzone.
createdDate	String	A time/date stamp for when the thermalzone created.
updatedDate	string	A time/date stamp for when the thermalzone last changed status.
rackLocation	RackLocation	Location of the thermalzone.
presence	String	Indicates the aggregated FAN presence information. Aggregated FAN presence format: Length of string indicate total slot of FAN component. For each byte the string: “1” means present. “0” means not present.
policy	TBD	**Editable**. thermalpolicy. To Be defined, need alignment with BDC-A cooling algorithms
presentTemperature	Int64	
maxFansSupported	Int64	
numberOfFansPresent	Int64	
outletTemperature	Int64	
volumetricAirflow	Int64	

Commented [NP6]: TBD

Table 22 Thermal zone location attributes

Name	RackLocation	
Attribute	**Type**	**Description**
rackUnits	Int64	RSARack.Units: indicates the Rack unit number.
uLocation	Int64	The index of the top-most U of the component, from top to bottom(1..MAXIMUM) 0 indicate not available.
uHeight	Int64	The height of managed zone. For example, 8 for 8U, 16 for 16U.

xLocation	Int64	The horizontal location within uLocation, from left to right(1.. MAXIMUM). “0” indicates not available.

5.2.1 JSON serialization

```
{
                “id”: int64,
                “uuid”: uid-str,
                “name”: string,
                “description”: string,
                “createdDate”: string,
                “updatedDate”: string,
                “rackLocation”: {
                       “rackUnis” : int64,
            “xLocation” : int64,
                       “uLocation” : int64,
                       “uHeight”: int64
                },
                “presence”:  string,
                “policy”: TBD,
                “presentTemperature”: int64,
                “maxFansSupported”: int64,
                “numberOfFansPresent”: int64,
      “outletTemperature”: int64,
      “volumetricAirflow”: int64
}
```

Commented [NP7]: TBD

5.2.2 Operations

This resource supports the GET, PUT, and POST operations.

5.2.2.1 GET

Method	URI	Description	Response code
GET	/v1/rack/thermalzones/{TZ_RUID}	List detailed information of a specific thermal/cooling zone.	Refer to section 1.7 HTTP response codes.

Request:

```
GET /v1/rack/thermalzones/{TZ_RUID}
Content-Type: application/json
```

Response:

```
{
                “id”: 1,
                “uuid”: “25892e17-80f6-415f-9c65-7395632f0232”,
                “name”: “thermal zone 1”,
                “description”: “thermal zone 1”,
                “createDate”: “2014-09-11T01:15:16+08:00”,
                “udpateDate”: “2014-09-11T02:15:11+08:00”,,
                “rackLocation”: {
                       “rackUnits” : 0,
            “xLocation” : 0,
                       “uLocation” : 1,
                       “uHeight”: 8
                },
                “presence”: “111100”,
                “policy”: TBD,
                “presentTemperature”: 50,
                “maxFansSupported”: 6,
                “numberOfFansPresent”: 6,
      “outletTemperature”: 60,
      “volumetricAirflow”: 80
}
```

Commented [NP8]: tbd

5.2.2.2 PUT

Table 23 Editable properties of a specific thermal zone

Name	Editable Properties	
Attribute	**Type**	**Description**
description	String	**Optional**. Description of the thermal zone.
policy	TBD	**Optional**. Thermal policy. To be defined, need alignment with BDC-A thermal algorithms.

Commented [NP9]: tbd

Method	URI	Description	Response code
PUT	/v1/rack/thermalzones/{TZ_RUID}	Update editable properties of a specific thermal zone.	Refer to section 1.7 HTTP response codes.

Request:

```
PUT /v1/rack/thermalzones/{TZ_RUID}
Content-Type: application/json
{
       “description”: “rack thermal zone 1”,
       “policy”: TBD,
}
```

Commented [NP10]: tbd

Response:

```
{
                “id”: 1,
                “uuid”: “25892e17-80f6-415f-9c65-7395632f0232”,
                “name”: “thermal zone 1”,
                “description”: “thermal zone 1”,
                “createDate”: “2014-09-11T01:15:16+08:00”,
                “udpateDate”: “2014-09-11T02:15:11+08:00”,,
                “rackLocation”: {
                       “rackUnits” : 0,
            “xLocation” : 0,
                       “uLocation” : 1,
                       “uHeight”: 8
                },
                “presence”: “111100”,
                “policy”: TBD,
                “presentTemperature”: 50,
                “maxFansSupported”: 6,
                “numberOfFansPresent”: 6,
                “outletTemperature”: 60,
                “volumetricAirflow”: 80
}
```

Commented [NP11]: tbd

5.3 Fan collections

The Fan Collection resource provides a collection of all fans that belong to a specific thermal zone.

Table 24 Fan collections attributes

Name	Collections of FAN	
Type URI	/v1/rack/thermalzones/{TZ_RUID}/fans	
Attribute	**Type**	**Description**
id	Int64	The index of fan. It is one integer between 1 and 6. Index of FAN within one specific thermalzone, equal to F_ZUID. F_ZUID stands for FAN unique index within specific thermalzones. F_ZUID starts from 1 with left-to-right order.
uuid	uid-str	Unique identifier for the thermalzone. UUID Pattern.
name	String	Human readable name for the thermalzone.
links	uri-str	*http://localhost:8080/v1/rack/thermalzones/{TZ_RUID}/fans/{F_ZUID}*.

5.3.1 JSON serialization

```
{
       “fans”: [
                      {
                               “id”: int64,
                               “uuid”: uid-str,
                               “name”: string,
                               “links”: uri-str
                      }, +
                ]
}
```

5.3.2 Operations

This resource supports the GET operation.

5.3.2.1 GET

Method	URI	Description	Response code
GET	/v1/rack/thermalzones/{TZ_RUID}/fans/	List fan collection information of a specific thermal zone.	Refer to section 1.7 HTTP response codes.

Request:

```
GET /v1/rack/thermalzones/{TZ_RUID }/fans/
Content-Type: application/json
```

Response:

```
{
       “fans”: [
                      {
                               “id”: 1,
                               “uuid”: “25892e17-80f6-415f-9c65-7395632e0232”,
                               “name”: “fan 1”,
                    “links”:
“http://localhost:8080/v1/rack/thermalzones/{TZ_RUID }/fans/1”,
                      },
                      {
                               “id”: 2,
                               “uuid”: “25892e17-80f6-415f-9c65-7395632d0232”,
                                “name”: “fan 2”,
                        “links”:
“http://localhost:8080/v1/rack/thermalzones/{TZ_RUID }/fans/2”,
                        }
        ]
}
```

5.4 Fan

Table 25 Fan attributes

Name	Fan	

Type URI	/v1/rack/thermalzones/{TZ_RUID }/fans/{F_ZUID }
	or
	/v1/rack/thermalzones/{TZ_RUID }/fans/{fan_uuid}	

Attribute	**Type**	**Description**
id	Int64	Index of FAN within one specific thermalzone, equal to F_ZUID. F_ZUID stands for FAN unique index within specific thermalzones. F_ZUID starts from 1 with left-to-right order.
uuid	uid-str	Unique identifier for the thermalzone. UUID Pattern.
name	String	Human readable name for the thermalzone.
description	String	**Editable**. Description of the fan.
createdDate	String	A time/date stamp for when the rack created status.
updatedDate	String	A time/date stamp for when the rack last changed status.
enabledState	RSARack.FANstate	**Editable**. Fan enabled state, refer to CIM_FAN.EnabledState efined in DMTF FAN Profile. 0: Unknown - the fan state is indeterminate. 2: Enabled- the fan shall be turned on. 3: Disabled – the fan shall be turned off. 5: Not applicable – the fan state is indeterminate or fan state management is not supported.
healthState	Int64	Health status of fan unit. This definition is derived from CIM_FAN.healthState defined in DMTF FAN profile. 0: Unknown. 5: Health. 10: Degraded/warning. 25: Critical failure.
desiredSpeed	Int64	**Editable**. Referring to below for format: “#### RPM”: The desired speed of the FAN using RPM as the unit. For example, “5000 RPM”. “#### PWM”: The desired speed of the FAN in percent of maximum speed. For example, “20 PWM”.
presentTachMeterReading	Int64	FAN Tacho Meter reading, using RPM as unit.
meterState	RSARack.TachMeterState	FAN Tacho Meter state. Supported states: “LowerNonCritical” ”UpperNonCritical” ”LowerCritical” ”UpperCritical” ”Normal”
tachMeterThreshold[]	key-value (integer)	**Editable**. Threshold of critical and non-critical state. Supported threshold: “LowerNonCritical” “UpperNonCritical” “LowerCritical”

Name	Fan	

Type URI	/v1/rack/thermalzones/{TZ_RUID }/fans/{F_ZUID }
	or
	/v1/rack/thermalzones/{TZ_RUID }/fans/{fan_uuid}	

Attribute	**Type**	**Description**
		“UpperCritical”
presence	String	The presence information of FAN. Supported states: “Present” ”NotPresent”
rackLocation	RackLocation	
speedUnit	RSARack.FANSpeedUnit	0-RPM, 1-PWM, default is RPM.
assetTag	String	
FRUInfo	RSARack.FRUInfo	

5.4.1 JSON serialization

```
{
        “id”: int64,
        “uuid”: uid-str,
        “name”: string,
        “description”: string,
        “createdDate”: string,
        “updateDate”: string,
        “enabledState”: int64,
        “healthState”: int64,
        “desiredSpeed”: string,
        “presentTachMeterRading”: int64,
        “meterSTate”: TachMeterState,
        “tachMeterThreshold”: [],
        “presence”: string,
        “links”: “string”,
        “rackLocation”: {
                “rackUnits” : int64,
                “xLocation” : int64,
                “uLocation” :  int64,
                “uHeight”: int64
        },
        “speedUnit”: FanSpeedUnit,
        “assetTag”: string,
        “FRUInfo”: FRUInfo
}
```

5.4.2 Operations

This resource supports GET, PUT, and POST operations.

5.4.2.1 GET

Method	**URI**	**Description**	**Response code**
GET	/v1/rack/thermalzones/{TZ_RUID}/fans/{F_ZUID}	List detailed information of a specific FAN.	Refer to section 1.7 HTTP response codes.

Request:

```
GET /v1/rack/thermalzones/{TZ_RUID}/fans/{F_ZUID}
Content-Type: application/json
```

Response:

```
{
       “id”: 1,
       “uuid”: “25892e17-80f6-415f-9c65-7395632e0232”,
       “name”: “fan 1”,
       “description”: “”,
       “createdDate”: “”,
       “updateDate”: “”,
       “enabledState”: 2,
       “healthState”: 5,
       “desiredSpeed”: 5000,
       “presentTachMeterReading”: 0,
       “meterState”: 4,
       “tachMeterThreshold”: [
                      {“lowerNonCritical”: 1000},
                      {“upperNonCritical”: 1500},
                      {“lowerCritical”: 800},
                      {“upperCritical”: 1000}
                      ],
       “presence”: “present”,
       “availableActions”: [
                      {
                               “action”: “requestStateChange”,
                               “capabilities”: [
                                                {
                                                      “propertyName”:
“enabledState”,
                                                      “allowableValues”: [
                                                            2,
                                                            3,
                                                      ]
                                                },
                               ]
                      },
                      {
                               “action”: “setDesiredSpeed”,
                               “capabilities”: [
                                                {
                                                      “propertyName”:
“unitType”,
                                                      “allowableValues”: [
                                                            “RPM”,
                                                            “PWM”,
                                                      ]
                                                },
                                                {
                                                      “propertyName”:
“desiredValue”,
                                                },
                               ]
                      },
       ]
       “links”: http://*:8080/v1/rack/thermalzones/{TZ_RUID }/fans/1,
       “rackLocation”: {
                “rackUnits” : 0,
                “xLocation” : 0,
                 “uLocation” :  1,
                 “uHeight”: 8
       }
}
```

5.4.2.2 PUT

Table 26 Editable properties of a specific fan unit

Name	Editable Properties	
Attribute	**Type**	**Description**
description	String	**Optional**. Description of the FAN unit
tachMeterThreshold[]	key-value (int64)	**Optional**. Threshold of critical and non-critical state. Supported threshold includes: “LowerNonCritical” “UpperNonCritical” “LowerCritical” “UpperCritical”
assetTag	String	**Optional**. String number of the asset tag of a specific FAN unit. Generally, this value is set by the FAN user for inventory management purposes.

Method	**URI**	**Description**	**Response code**
PUT	rack/thermalzones/{TZ_RUID}/fans/{F_ZUID}	Update editable properties of a specific FAN unit.	Refer to section 1.7 HTTP response codes.

Request:

```
PUT /v1/rack/thermalzones/{TZ_RUID }/fans/{F_ZUID }
Content-Type: application/json
{
       “description”: “FAN 1 in zone 1”
    “tachmeterThreshold”: [
                {“lowerNonCritical”: 1000},
                {“upperNonCritical”: 1500},
                {“lowerCritical”: 800},
                {“upperCritical”: 1000}
       ],

}
```

Response:

```
200
{
       “id”: 1,
       “uuid”: “25892e17-80f6-415f-9c65-7395632e0232”,
       “name”: “fan 1”,
       “description”: “FAN 1 in zone 1”,
       “createdDate”: “2014-09-11T01:15:16+08:00”,
       “updateDate”: 2014-09-11T02:11:11+08:00 “”,
       “enabledState”: 2,
       “healthState”: 5,
       “desiredSpeed”: 5000,
       “tachoMeterThreshold”: [
                      {“lowerNonCritical”: 1000},
                      {“upperNonCritical”: 1500},
                      {“lowerCritical”: 800},
                      {“upperCritical”: 1000}
       ],
       “presence”: “present”,
   “assetTag”: “0987654321”,
       “links”: http://*:8080/v1/rack/thermalzones/{TZ_RUID }/fans/1,
       “rackLocation”: {
                “rackUnits” : 0,
      “xLocation” : 0,
                “uLocation” : 1,
                “uHeight”: 8
       }
       “availableActions”: [
                      {
                               “action”: “requestStateChange”,
                               “capabilities”: [
                                                {
                                                      “propertyName”:
“enabledState”,
                                                      “allowableValues”: [
                                                            2,
                                                            3,
                                                      ]
                                                },
                               ]
                      },
                      {
                               “action”: “setDesiredSpeed”,
                               “capabilities”: [
                                                {
                                                      “propertyName”:
“unitType”,
                                                      “allowableValues”: [
                                                            “RPM”,
                                                            “PWM”,
                                                      ]
                                                },
                                                {
                                                      “propertyName”:
“desiredSpeed”,
                                                },
                               ]
                      },
       ]
}
```

5.4.2.3 POST

Table 27 RequestStateChange action

Name	RequestStateChange action	
Attribute	**Type**	**Description**
action	String	Action type. “RequestStateChange”.
enabledState	Int64	Fan enabled state. Refer to CIM_FAN.EnabledState defined in DMTF FAN Profile. 2: Enabled- the fan shall be turned on. 3: Disabled – the fan shall be turned off.

Method	URI	Description	Response code
POST	/v1/rack/thermalzones/{TZ_RUID}/fans/{F_ZUID}	Request enabled state change for a specific FAN.	Refer to section 1.7 HTTP response codes.

Request:

```
POST /v1/rack/thermalzones/{TZ_RUID }/fans/{F_ZUID}
Content-Type: application/json
{
       “action”: “requestStateChange”,
       “enabledState”: 2,
}
```

Response:

Normal code: 202 Accepted. The request has been accepted for processing. This operation does not return a response body.

Table 28 SetDesiredSpeed action

Name	SetDesiredSpeed Action	
Attribute	**Type**	**Description**
action	String	Action type. “SetDesiredSpeed”.
unitType	String	“RPM”: The desired speed of the FAN in RPM. “PWM”: The desired speed of the FAN in percent of maximum speed.
desiredSpeed	Int64	Desired fan speed value.

Method	URI	Description	Response code
POST	/v1/rack/thermalzones/{TZ_RUID}/fans/{F_ZUID}	Request desiredSpeed change for a specific FAN.	Refer to section 1.7 HTTP response codes.

Request:

```
POST /v1/rack/thermalzones/{TZ_RUID }/fans/{F_ZUID}
Content-Type: application/json
{
       “action”: “setDesiredSpeed”,
       “unitType”: “RPM”,
       “desiredSpeed”: “5000”
}
```

Response:

Normal code: 202 Accepted. The request has been accepted for processing. This operation does not return a response body.

§

6 Drawer

6.1 Drawer collections

The Intel® Rack Scale Architecture drawer collection resource provides a collection of all drawers that belong to the rack.

Table 29 Compute system collections attributes

Name	Collections of drawers	
Type URI	/v1/rack/drawers/	
Attribute	**Type**	**Description**
id	Int64	Index of draw within one rack, equal to DR_RUID. RD_RUID stands for draw unique index within rack. It starts from 1.
uuid	uid-str	Unique identifier for the draw. UUID Pattern.
name	String	Human readable name for the draw.
links	uri-str	*http://localhost:8080/v1/rack/draws/{DR_RUID}*.

6.1.1 JSON serialization

```
{
       “drawers”: [
                      {
                              “id”: int64,
                              “uuid”: uid-str,
                              “name”: string,
                              “links”: uri-str,
                      }, +
               ]
}
```

6.1.2 Operations

This resource supports the GET operation.

6.1.2.1 GET

Method	**URI**	**Description**	**Response code**
GET	/v1/rack/drawers	List collection information for all drawers.	Refer to section 1.7 HTTP response codes.

Request:

```
GET /v1/rack/drawers
Content-Type: application/json
```

Response:

```
{
       “drawers”: [
                      {
                              “id”: 1,
                              “uuid”: “25892e17-80f6-415f-9c65-7395632f0223”,
                              “name”: “drawer 1”,
                 "links": "http://localhost:8080/v1/rack/drawers/1"

                      },
                      {
                              "id": 2,
                              "uuid": "25892e17-80f6-415f-9c65-7395632f0224",
                              "name": "drawer 2",
                       "links": "http://localhost:8080/v1/rack/drawers/2"

                      }
       ]
}
```

6.2 Drawer

Table 30 Drawer attributes

Name	Drawer (a.k.a tray)	
Type URI	/v1/rack/drawers/{DR_UUID}/ or /v1/rack/drawers/{DR_RUID }/ - alias resource URL	
Attribute	**Type**	**Description**
id	Int64	Index of drawer within one rack, equal to DR_RUID. DR_RUID stands for draw unique index within rack. It starts from 1.
uuid	String	Unique identifier for the drawer. UUID Pattern.
name	String	Human readable name for the drawer.
description	String	**Editable.** Description of the drawer.
createdDate	String	A time/date stamp for when the drawer created.
updatedDate	String	A time/date stamp for when the draw last changed status.
trayRUID	Int64	Tray(Compute system) unique id within rack.
trayPUID	String	RackPUID-TrayRUID. "0-0": Default initialization.
location	Location	Location of the drawer.
RMMPresent	Boolean	true or false.
rackSupportsDisaggregatedPowerCooling	Boolean	true or false
powerAndThermal	PowerAndThermal	

Table 31 Drawer location attributes

Name	Location	
Attribute	**Type**	**Description**
pod	String	UUID of pod, default is "N/A".
rack	String	UUID of rack.
rackUnits	Int64	RSARack.Units: indicates the Rack unit number.
uLocation	Int64	The index of the top-most U of the component, from top to bottom (1..MAXIMUM). "0" indicates not available .
uHeight	Int64	The height of managed zone. For example, 8 for 8U, 16 for 16U.
xLocation	Int64	The horizontal location within uLocation, from left to right (1.. MAXIMUM). "0" indicates not available.

Table 32 Drawer power and thermal attributes

Name	PowerAndThermal	
Attribute	**Type**	**Description**
powerConsumedWatts	Int64	The total power consumption of the computer system. The unit is measured in watts.
presentTemperature	Int64	Current temperature of the computer system.

6.2.1 JSON serialization

```
{
                “id”: int64,
                “uuid”: uid-str,
                “name”: string,
                “description”: string,
                “createdDate”: string,
                “updatedDate”: string,
          “trayRUID”: int64,
          “trayPUID”: string,
          “thermalState”: int64,
                “location”: {
                       “pod” : string,
                          “rack” : string,
             “rackUnits”: int64,
             “xLocation”: int64,
             “uLocation”: int64,
             “uHeight”: int64
                },
       “powerAndThermal”: {
            “powerConsumedWatts”: int64,
            “presentTemperature”: int64
       },
                “RMMPresent”: Boolean,
                “rackSupportsDisaggregatedPowerCooling”: Boolean
}
```

6.2.2 Operations

This resource supports the GET, PUT, and POST operations.

6.2.2.1 GET

Query information of specific drawer.

Method	**URI**	**Description**	**Response code**
GET	/v1/rack/drawers/{DR_UUID}	Query information of a drawer.	Refer to section 1.7 HTTP response codes.

Request:

```
GET /v1/rack/drawers/{DR_UUID}
Content-Type: application/json
```

Response:

```
{
                “id”: 1,
                “uuid”: “25892e17-80f6-415f-9c65-7395632f0232”,
                “name”: “drawer 1”,
                “description”: “drawer 1”,
                “createDate”: “2014-09-11T01:15:16+08:00”,
                “udpateDate”: “2014-09-11T02:15:11+08:00”,
          “trayPUID”: “1-1”,
          “trayRUID”: 1,
          “thermalState”: 3,
                “location”: {
                       “pod” : ”N/A”,
                       “rack” :  “34892e17-80f6-415f-9c65-7395632f0232”,
             “rackUnits”: 0,
             “xLocation”: 0,
             “uLocation”: 3,
             “uHeight”: 2
                },
       “powerAndThermal”: {
            “powerConsumedWatts”: 1000,
            “presentTemperature”: 80
       },
                “RMMPresent”: true,
                “rackSupportsDisaggregatedPowerCooling”: false
}
```

6.2.2.2 PUT

Table 33 Editable properties of a specific drawer

Name	Editable Properties	
Attribute	**Type**	**Description**
description	string	**Editable:** description of the drawer.

Method	URI	Description	Response code
PUT	/v1/rack/drawers/{DR_UUID}	Update editable properties of a specific drawer.	Refer to section 1.7 HTTP response codes.

Request:

```
PUT /v1/rack/drawers/{DR_UUID}
Content-Type: application/json
{
       “description”: “drawer 1”
}
```

Response:

```
{
                “id”: 1,
                “uuid”: “25892e17-80f6-415f-9c65-7395632f0232”,
                “name”: “drawer x”,
                “description”: “drawer 1”,
                “createDate”: “2014-09-11T01:15:16+08:00”,
                “udpateDate”: “2014-09-11T02:15:11+08:00”,
           “trayPUID”: “1-1”,
           “trayRUID”: 1,
           “thermalState”: 3,
                “location”: {
                       “pod” : “N/A”,
                       “rack” :  ” 34892e17-80f6-415f-9c65-7395632f0232”,
             “rackUnits”: 0,
             “xLocation”: 0,
             “uLocation”: 3,
             “uHeight”: 2
                },
       “powerAndThermal”: {
            “powerConsumedWatts”: 1000,
            “presentTemperature”: 80
       },
                “RMMPresent”: true,
                “rackSupportsDisaggregatedPowerCooling”: false
}
```

6.2.2.3 POST

Table 34 Reset action

Name	Reset Action	
Attribute	**Type**	**Description**
action	String	Action type. “reset”.
resetType	String	The type of reset. Allowable values: “soft” “hard”

Method	URI	Description	Response code
POST	/v1/rack/drawers/{DR_UUID} or /v1/rack/drawers/{DR_UUID}	Reset specific drawer.	Refer to section 1.7 HTTP response codes.

Request:

```
POST /v1/rack/drawers/{DR_UUID}
Content-Type: application/json
{
       “action”: “reset”,
       “resetType” “soft”
}
```

Response:

Normal code: 202 Accepted. The request has been accepted for processing,.

This operation does not return a response body.

§

7 Management Backplane

The management backplane is part of the RMM. This component is platform-specific, like the Control Module (CM) defined in the BDC-A management backplane.

7.1 Management backplane collections

This Intel® Rack Scale Architecture system collection resource provides collection of all management backplanes that belong to the rack.

Table 35 Management backplane collections attributes

Name	Collection of management backplanes	
Type URI	/v1/rack/mbps/	
Attribute	**Type**	**Description**
mbp_id	Int64	The index of the management backplane system that is unique within the rack. Generally, it equals the Backplane ID.
uuid	uid-str	Unique identifier for the management backplane collection. UUID Pattern.
name	String	Human readable name for the management backplane system.
links	uri-str	Reference to the management backplane resource. (URI) "http://localhost:8080/v1/rack/{mbp_id}"

7.1.1 JSON serialization

```
{
       “mbps”: [
                      {
                              “id”: int64,
                              “uuid”: uid-str,
                              “name”: string,
                “links”: uri-str,
                      }, +
                ]
}
```

7.1.2 Operations

This resource supports the GET operation.

7.1.2.1 GET

Method	**URI**	**Description**	**Response code**
GET	/v1/rack/mbps	List collection information for all the management backplane.	Refer to section 1.7 HTTP response codes.

Request:

```
GET /v1/rack/mbps
Content-Type: application/json
```

Response:

```
{
                “mbps”: [
                {
                “id”: 1,
                “uuid”: “25892e17-80f6-415f-9c65-7395632f0223”,
                “name”: “middle backplane 1”,
                “links”: “http://localhost:8080/v1/rack/mbps/1”
                },
                {
                “id”: 2,
                “uuid”: “25892e17-80f6-415f-9c65-7395632f0223”,
                “name”: “middle backplane 2”,
                “links”: “http://localhost:8080/v1/rack/mbps/2”
                },
                ]
}
```

7.2 Management backplane

Table 36 Management backplane attributes

Name	Management Backplane
Type URI	/v1/rack/mbps/{mbp_id}/ or /v1/rack/mbps/{mbp_uuid}

Attribute	**Type**	**Description**
id	Int64	The index of the management backplane that is unique within the rack. Generally, it equals the Backplane ID (mbp_id), starting from 1.
uuid	uid-str	Unique identifier for the management backplane system. UUID Pattern.
name	String	Human readable name for the management backplane system.
description	String	**Editable**. Description of the management backplane system.
createdDate	String	Timestamp when the management backplane entry was created (ISO 8601). ISO 8601: YYYY-MM-DDThh:mm:ss.SSS'Z' Year, Month, Date (UTC) 1134 Hour, Minute, Second 1135 Relative Time Zone Shift (Hour and Minute) Ex: "2013-01-10T01:09:57+08:00".
updatedDate	String	Timestamp when the management backplane status was changed (ISO 8601).
MBPID	Int64	Management backplane unique id within the rack.
MBPHWAddress	String	The physical address of the management backplane. The format is as follows: “physical_protocol:physical_address” BDC-A uses USB slot to indicate physical address of the management backplane. For example, “usb:2-1” indicates usb device address is 2, and the usb port is 1. or “ipmb:1-20” indicates ipmb channel 1, smbus device address 20.
MBPIPAddress	String	The IP address of the management backplane.
firmwareState	String	“up”,”down”,”updating”.
rackLocation	RackLocation	The location information of the management backplane.
assetTag	String	**Editable**. String number of the asset tag of the management back plane. Generally, this value is set by the rack user for inventory management purposes.
FRUInfo	RSARack.FRUInfo	MBP FRU (Field Replacement Unit) Information. Generally, FRUInfo is set by the manufacture and normally not changeable.
availableActions	Action Array	Refer to section 1.12 Available actions (RF 5.1).

7.2.1 JSON serialization

```
{
        “id”: int64,
        “uuid”: uid-str,
        “name”: string,
        “description”: string,
        “createdDate”: string,
        “updatedDate”: string,
        “MBPID”: int64,
        “MBPHWAddress”: string,
        “MBPIPAddress”: string,
        “firmwareState”: string,
        “rackLocation”: {
                “rackUnits” : int64,
      “xLocation” : int64,
                “uLocation” :  int64,
                “uHeight”: int64
        },
        “assetTag”: string,
        “FRUInfo” : {
                “serialNumber”: string,
                “manufacture”: string,
                “modelNumber”: string,
                “partNumber”: string,
                “firmwareRevision”: string
        },
        “availableActions”: [],
}
```

7.2.2 Operations

This resource supports the GET, PUT, and POST operations.

7.2.2.1 GET

Method	URI	Description	Response code
GET	/v1/rack/mbps/{mbp_id}	Query information of a specific management backplane.	Refer to section 1.7 HTTP response codes.

Request:

```
GET /v1/rack/mbps/{mbp_id}
Content-Type: application/json
```

Response:

```
{
        “id”: 1,
        “uuid”: “25892e17-80f6-415f-9c65-7395632f0223”,
        “name”: “middle backplane 1”,
        “description”: “middle backplane 1”,
        “createdDate”: “2014-09-10T01:09:26+08:00”,
        “updatedDate”: “2014-09-11T11:11:16+08:00”,
        “MBPID”: “1”,
        “MBPHWAddress”: “usb:2-1”,
        “MBPIPAddress”: “192.168.1.120”,
        “firmwareState”: “up”,
        “rackLocation”: {
        “rackUnits” : 0,
                “xLocation” : 0,
                “uLocation” : 8,
                “uHeight”: 8,
        },
        “assetTag”: “”,
        “FRUInfo”: {
                “serialNumber”: “”,
                “manufacture”: “”,
                “modelNumber”: “”,
                “partNumber”: “”,
                “firmwareRevision”: “”,
        },
        “availableActions”: [
                        {
                                “action”: “reset”,
                                “capabilities”: [
                                                {
                                                        “propertyName”:
“resetType”,
                                                        “allowableValues”: [
                                                              “soft”,
                                                        ]
                                                },
                                ]
                        },
                        {
                                “action”: “setUartTarget”,
                                “capabilities”: [
                                                {
                                                        “propertyName”:
“targetIndex”,
                                                },
                                ]
                        },
                        {
                                “action”: “update”,
                                “capabilities”: [
                                                {
                                                        “propertyName”:
“image”,
                                                },
                                ]
                        },
        ]
}
```

7.2.2.2 PUT

Table 37 Editable properties of a specific management backplane

Name	Editable Properties	
Attribute	**Type**	**Description**
description	String	**Optional**. Description of the specific management backplane.
assetTag	String	**Optional**. String number of the asset tag of the management back plane. Generally,

		this value is set by the rack user for inventory management purposes.

Method	URI	Description	Response code
PUT	/v1/rack/mbps/{mbp_id }	Update editable properties of a specific management backplane.	Refer to section 1.7 HTTP response codes.

Request:

```
PUT /v1/rack/mbps/{mbp_id}
Content-Type: application/json
{
        “description”: “middle plane 1”
        “assetTag”: “11223344”}
```

Response:

```
{
        “id”: 1,
        “uuid”: “25892e17-80f6-415f-9c65-7395632f0223”,
        “name”: “middle backplane 1”,
        “description”: “middle backplane 1”,
        “createdDate”: “2014-09-10T01:09:26+08:00”,
        “updatedDate”: “2014-09-11T11:11:16+08:00”,
        “MBPID”: “1”,
        “MBPHWAddress”: “usb:2-1”,
        “MBPIPAddress”: “192.168.1.120”,
        “firmwareState”: “up”,
        “rackLocation”: {
        “rackUnits” : 0,
                “xLocation” : 0,
                “uLocation” : 8,
                “uHeight”: 8,
        },
        “assetTag”: “11223344”,
        “FRUInfo”: {
                “serialNumber”: “”,
                “manufacture”: “”,
                “modelNumber”: “”,
                “partNumber”: “”,
                “firmwareRevision”: “”,
        },
        “availableActions”: [
                        {
                                “action”: “reset”,
                                “capabilities”: [
                                                {
                                                        “propertyName”:
“resetType”,
                                                        “allowableValues”: [
                                                                “soft”,
                                                        ]
                                                },
                                ]
                        },
                        {
                                “action”: “setUartTarget”,
                               “capabilities”: [
                                                {
                                                      “propertyName”:
“targetIndex”,
                                                },
                               ]
                      },
                      {
                               “action”: “update”,
                               “capabilities”: [
                                                {
                                                      “propertyName”:
“image”,
                                                },
                               ]
                      },
       ]}
```

7.2.2.3 POST

Table 38 Reset action

Name	Reset Action	
Attribute	**Type**	**Description**
action	String	Action type. “reset”.
resetType	String	The type of reset. Allowable values: “soft”

Method	**URI**	**Description**	**Response code**
POST	/v1/rack/mbps/{mbp_id }	Reset management backplane.	Refer to section 1.7 HTTP response codes.

Request:

```
POST /v1/rack/mbps/{mbp_id}
Content-Type: application/json
{
       “action”: “reset”,
       “resetType”: “soft”
}
```

Response:

Normal code: 202 Accepted. The request has been accepted for processing.

This operation does not return a response body.

Table 39 SetUartTarget action

Name	SetUartTarget Action	
Attribute	**Type**	**Description**
action	String	Action type. “SetUartTarget”
targetIndex	Int64	The target UART index. (0-8). 0 indicates CM. 1-8 indicates CPP.

Method	**URI**	**Description**	**Response code**
POST	/v1/rack/mbps/{mbp_id }	Set UART Target of the	Refer to section 1.7 HTTP response codes.

		management backplane.	

Request:

```
POST /v1/rack/mbps/{mbp_id}
Content-Type: application/json
{
       “action”: “setUartTarget”,
       “targetIndex”: 1
}
```

Response:

Normal code: 202 Accepted. The request has been accepted for processing.

Table 40 Update action

Name	Update Action	
Attribute	**Type**	**Description**
action	String	Action type. “Update”
image	String	CM/MBP FW package content encoded with base64.

Method	**URI**	**Description**	**Response code**
POST	/v1/rack/mbps/{mbp_id }	Update CM FW.	Success: 202 Failure: 406

Request:

```
POST /v1/rack/mbps/{mbp_id}
Content-Type: application/json
{
   “action”: “update”,
       “image”: “XAXX111ddxx###.....”
}
```

Successful Response:

Response code: 202(Accepted).
Response body:

```
{
       “result”: “updating”,
       “rev”: “0.9”
}
```

The request has been accepted for processing. The client should query the status and firmware version to verify if the update is successful or not.

Failure Response:

Response code: 406(Not acceptable).
Response body:

```
{
       “result”: “unmatched version”,
}
Or
{
       “result”: “invalid image”,
}
```

§

8 Event and Logging

This section defines the event and logging functionalities of RMM which are aligned with the definition of event service and logging service of Redfish.

8.1 Event service

Table 41 Event service

Name	Rack event service	
Attribute	**Type**	**Description**
oem	Object	OEM extension object which contains properties defined by the manufacturer. Pattern: OEM info combined with version number which is split by 3 dots. For example, "Intel.1.0.1".
name	String	The name of the resource or array element.
modified	String	The date-time stamp that the object was last modified. ISO 8601: YYYY-MM-DDThh:mm:ss.SSS'Z' Year, Month, Date (UTC) 1134 Hour, Minute, Second 1135 Relative Time Zone Shift (Hour and Minute) For example, "2013-01-10T01:09:57+08:00".
type	String	Type represents the schema used for the resource and indicates not only the schema but the version of that schema. Pattern: Type info combined with version number which is split by 3 dots. For example, "RMM.1.0.0"
description	String	Description of the resource and is used for commonality in the schema definitions.
subscription	Array	**Editable**. A collection of Listener Destination resources.
self	String	URI of rack event service resource.

8.1.1 Event URIs

Table 42 Event URIs

Name	Event URIs	
Category	**Type**	**Description**
Rack	String	/v1/rack/rf_event
Power zone collection	String	/v1/rack/powerzones/rf_event
Power zone	String	/v1/rack/powerzones/{zone_id}/ rf_event
Psu collection	String	/v1/rack/powerzones/{zone_id}/psus/ rf_event
Psu	String	/v1/rack/powerzones/{zone_id}/psus/{psu_id}/ rf_event
Thermal zone collection	String	/v1/rack/thermalzones/ rf_event
Thermal zone	String	/v1/rack/thermalzones/{zone_id}/ rf_event
Fan collection	String	/v1/rack/thermalzones/{zone_id}/fans/ rf_evnet
Fan	String	/v1/rack/thermalzones/{zone_id}/fans/{fan_id}/ rf_event
Drawer collection	String	/v1/rack/drawers/ rf_event
Drawer	String	/v1/rack/drawers/{drawer_uuid}/ rf_event
Mbp collection	String	/v1/rack/mbps/ rf_event
Mbp	String	/v1/rack/mbps/{mbp_id}/ rf_event

8.1.2 JSON serialization

```
{
       "oem": string,
       “name”: string,
       "modified": string,
       "type": string,
       "description": string,
       "subcsription": [
               {“listener”:string},
       ],
       "self": string,http://192.168.10.1:8090/rack/event
}
```

8.1.3 Operations

8.1.3.1 GET

Method	URI	Description	Response code
GET	Reference to 8.1.1	List redfish event service information.	Normal: 200 Error: Refer to section 1.7 HTTP response codes.

Request:

```
GET ${URI}
Content-Type: application/json
```

Response:

```
{
       "oem": "Intel.1.0.1",
       “name”: "RSA_RMM",
       "modified": "2015-04-01T12:38:47+01:00",
       "type": "RMM1.0.0",
       "description": "For internal use only",
       "subscription": [
               {“listener”: {"listener":
"http://localhost:8092/${listener}"}},
       ],
       "self":
"http://localhost:8090/${URI}",http://192.168.10.1:8090/rack/event
}
```

8.1.3.2 POST

Table 43 Subscribe/unsubscribe events

Name	Subscribe events	
Attribute	**Type**	**Description**
action	String	Action type. Allowable values: “subscribe” “unsubscribe”
type	String	The type of subscription. Allowable values: “rack”
destination	String	URL of listener destination.
protocol	String	Protocol of subscription, allowable values: “redfish”

Method	URI	Description	Response code
POST	Reference to 8.1.1	Subscribe/unsubscribe event	Normal: 200 Error: Refer to section 1.7 HTTP response codes.

Request:

```
POST /v1/rack/rf_event
Content-Type: application/json
{
       “action”:”subscribe”,
       “type”:”rack”,
       “destination”:”http://localhost:8092/${listener}”,
       “eventTypes”:
               {"event":"add"},
               {"event":"alert"}
       ],
       “protocol”:”redfish”
}
```

Response:

Normal code: 202 Accepted. The request has been accepted for processing,.

This operation does not return a response body.

8.2 Logging Service

Table 44 Log Entry

Name	Log entry	
Type URI	/v1/rack/rflog	
Attribute	**Type**	**Description**
count	Int64	OEM extension object which contains properties defined by the manufacturer. Pattern: OEM info combined with version number which is split by 3 dots, for example "Intel.1.0.1".
length	Int64	The name of the resource or array element.
loginfo	String	The date-time stamp that the object was last modified. ISO 8601: YYYY-MM-DDThh:mm:ss.SSS'Z' Year, Month, Date (UTC) 1134 Hour, Minute, Second 1135 Relative Time Zone Shift (Hour and Minute) For example, "2013-01-10T01:09:57+08:00".

8.2.1 JSON serialization

```
{
       "count": int64,
       "length": int64,
       "loginfo": [
               String,
               String,
       ]
}
```

8.2.2 Operations

8.2.2.1 GET

Method	URI	Description	Response code
GET	/v1/rack/rflog?count=${count}	Get redfish specific logs.	Normal: 200 Refer to section 1.7 HTTP response codes.

Request:

```
GET /v1/rack/rflog?count=${count}
Content-Type: application/json
```

Response:

```
{
       "count": 1,
       "length": 256,
       "loginfo": [
                "4 2015/04/02 15:10:37.040870 [info] rf_record_log: 79 4,PSU
remove event...\n"
       ]
}
```