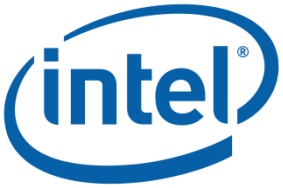


Intel® Rack Scale Design (Intel® RSD) Generic Assets Management Interface (GAMI)

API Specification
Software v2.4

April 2019

Revision 001

Document Number: 608495-001

Contents

Figures

Tables

Revision History

Revision	Description	Date
001	Initial release of Intel® RSD v2.4	April 2019

§

1.0 Introduction

This document contains information about the Intel® Rack Scale Design Generic Assets Management Interface (GAMI) API Specification designed and implemented for Intel® Rack Scale Architecture Software v2.4 Release.

1.1 Intended Audience

The intended audiences for this document include designers and engineers working with the Intel® Software v2.4 Release, porting this software to HW platforms.

1.2 Conventions

The keywords/phrases "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in *Key Words for Use in RFCs to Indicate Requirement Levels, March 1997*, RFC 2119, refer to Table 2.

1.3 Notes and Symbol Convention

Symbol and note convention is similar to typographical conventions used in Cloud Infrastructure Management Interface 6 (CIMI) Model and RESTful HTTP-based Protocol 7, An Interface for Managing Cloud Infrastructure - a DMTF Standard Publication, refer to Table 2.

The notation used in JSON* serialization description:

- Mandatory in italics indicate data types instead of literal Mandatory.
- 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: The lack of ellipses does not mean no extensibility point exists; it is just not explicitly called out.

1.4 Terminology

Table 1. Terminology

Term	Definition
AMC	Asset Management Core
BMC	Baseboard Management Controller
CIMI	Cloud Infrastructure Management Interface
DCB	Data Center Bridging
DCBX	Data Center Bridging Extensions
ETS	Enhanced Transmission Selection
GAM	Generic Asset Management

Term	Definition
GAMI	Generic Assets Management Interface
HAL	Hardware Abstraction Layer
HTTP	Hypertext Transfer Protocol
JSON*	JavaScript Object Notation*
JSON-RPC*	Stateless, lightweight remote procedure call (RPC) protocol. See *http://www.jsonrpc.org*.
LLDP	Link Layer Discovery Protocol
OData	Open Data Protocol
PCIe*	Peripheral Component Interconnect Express*
PNC	Pooled NVMe Controller
POD	A physical collection of multiple racks
PODM	POD Manager
PFC	Priority Flow Control
PSME	Pooled System Management Engine
PTAS	Power Thermal-Aware Solution
PWM	Pulse-width Modulation
RESTful	Representational State Transfer
RMM	Rack Management Module
SDV	Software Development Vehicle
URI	Uniform Resource Identifier
UUID	Universally Unique Identifier
VLAN	Virtual LAN
XML	Extensible Markup Language

1.5 References

Table 2. Reference Documents and Resources

Doc ID	Title	Location
608486	*Intel® Rack Scale Design (Intel® RSD) Pooled System Management Engine (PSME) User Guide Software v2.4*	Note: https://www.intel.com/content/www/us/en/architecture-and-technology/rack-scale-design/rack-scale-design-resources.html
608487	*Intel® Rack Scale Design (Intel® RSD) Conformance and Software Reference Kit Getting Started Guide v2.4*	
608488	*Intel® Rack Scale Design (Intel® RSD) POD Manager (PODM) Release Notes Software v2.4*	
608489	*Intel® Rack Scale Design (Intel® RSD) POD Manager (PODM) User Guide Software v2.4*	
608490	*Intel® Rack Scale Design (Intel® RSD) Pooled System Management (PSME) Release Notes Software v2.4*	
608491	*Intel® Rack Scale Design Storage Services API Specification Software v2.4*	
608492	*Intel® Rack Scale Design (Intel® RSD) Architecture Specification Software v2.4*	
608493	*Intel® Rack Scale Design (Intel® RSD) Pod Manager (PODM) Representational State Transfer (REST) API Specification Software v2.4*	
608494	*Intel® Rack Scale Design (Intel® RSD) Rack Management Module (RMM) Representatinal State Transfer (REST) API Specification Software v2.4*	
608496	*Intel® Rack Scale Design (Intel® RSD) Pooled System Management Engine (PSME) REST API Specification Software v2.4*	
608497	*Intel® Rack Scale Design (Intel® RSD) Conformance Test Suite (CTS) Release Notes*	

Doc ID	Title	Location
608298	*Field Programmable Gate Array (FPGA) over Fabric Protocol Architecture Specification*	https://cdrdv2.intel.com/v1/dl/getContent/608298
596167	*Intel® Rack Scale Design (Intel® RSD) for Cascade Lake Platform Firmware Extension Specification*	https://cdrdv2.intel.com/v1/dl/getContent/596167
DSP0263	*Cloud Infrastructure Management Interface 6 (CIMI) Model and RESTful HTTP-based Protocol 7 An Interface for Managing Cloud Infrastructure*	https://www.dmtf.org/sites/default/files/standards/documents/DSP0263_2.0.0.pdf
DSP0266	*Scalable Platforms Management API Specification v1.0.0*	*https://dmtf.org/sites/default/files/standards/documents/DSP0266_1.0.0.pdf*
DSP8010	*Redfish Schema v2016.3*	*https://www.dmtf.org/sites/default/files/standards/documents/DSP8010_2016.3.zip*
DSP8010	*Redfish* Schema v2018.1*	
DSP8010	*Redfish* Schema v2018.2*	
RFC2119	*Key Words for Use in RFCs to Indicate Requirement Levels, March 1997*	*https://ietf.org/rfc/rfc2119.txt*
DSP0266	*Redfish* Scalable Platforms Management API Specification v1.5.0*	*https://www.dmtf.org/sites/default/files/standards/documents/DSP0266_1.5.0.pdf*
RFC3339	*Date and Time on the Internet: Timestamps*	*https://tools.ietf.org/html/rfc3339*

NOTE: Documents referenced in this table which have a Document ID, but cannot be accessed, can be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm obtain a copy.

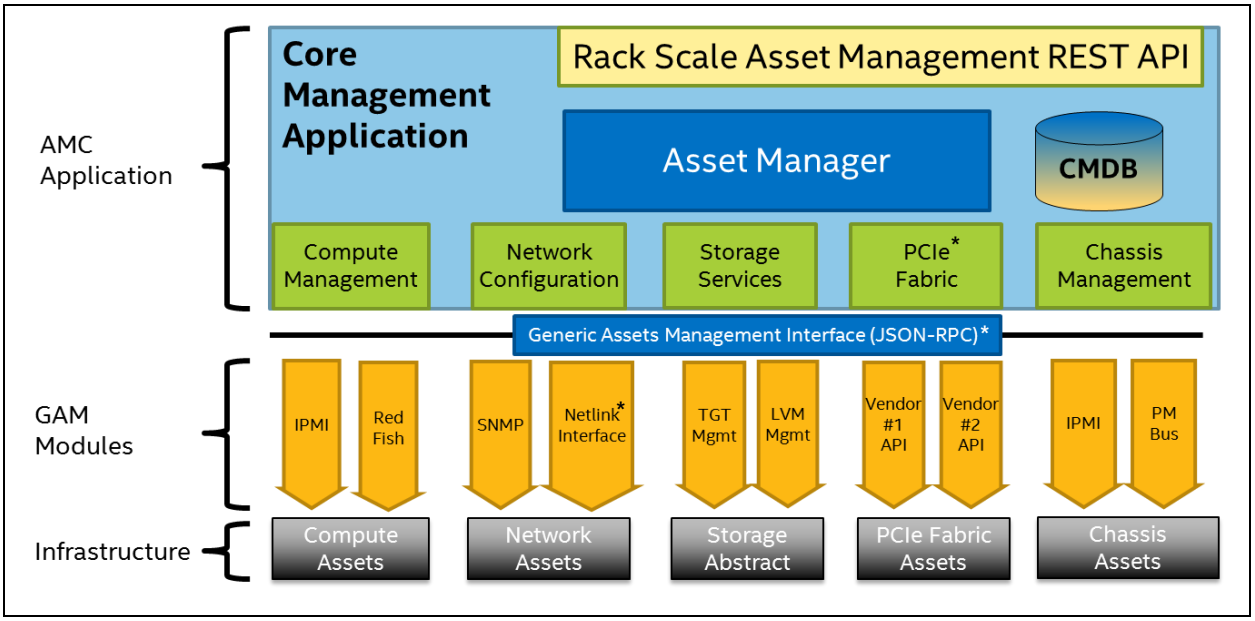
§

2.0 Generic Asset Management Interface

2.1 GAMI API Architecture and Design Principles

This document describes the Asset Management Core (AMC) application to the Generic Asset Management (GAM) module interconnection protocol. The AMC application supports a modular hardware abstraction layer. Each hardware component is handled by a dedicated GAM module managed via JSON-RPC* API sent over an HTTP socket. This document specifies the JSON-RPC API to communicate with GAM modules.

Figure 1. Generic Assets Management Interface in the PSME Architecture



The GAMI JSON-RPC v2.0 protocol is HTTP and uses an HTTP POST command to send the JSON* structures. A sample request and response (below) present JSON structures exchanged by the AMC application acting as a client, and the GAM module acting as a server:

Sample request:

```
POST /<EntryPoint>/ HTTP/1.1
Host: jsonrpc.rackscale.intel.com
Content-Type: application/json
Content-Length: ...
Accept: application/json
{
        "jsonrpc": "2.0",
        "method": "getProcessorInfo",
        "params": {
                "processor": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

Note:

- Content-Type should be `application/json-rpc`, but may be `application/json` or `application/jsonrequest`.
- Content-Length must be specified.
- Accept must be specified and should read application/json-rpc but may be `application/json` or `application/jsonrequest`.
- The host *is* not obligatory.

Sample response:

```
HTTP/1.1 200 OK
Connection: close
Content-Length: ...
Content-Type: application/json
Date: Tue, 17 Feb 2015 15:43:55 CEST

{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "socket": "1",
                "processorType": "CPU",
                "cpuid": {
                        "vendorId": "GenuineIntel",
                        "numericId" 329442,
                        "family": 6,
                        "model": 5,
                        "step": 2,
                        "microcodeInfo": 11
                },
                "maxSpeedMHz": 3700,
                "totalCores": 8,
                "enabledCores": 4,
                "totalThreads": 4,
                "enabledThreads": 2,
                "oem": {}
        },
        "id": 987
}
```

2.2 GAMI API Synchronous and Asynchronous Operations

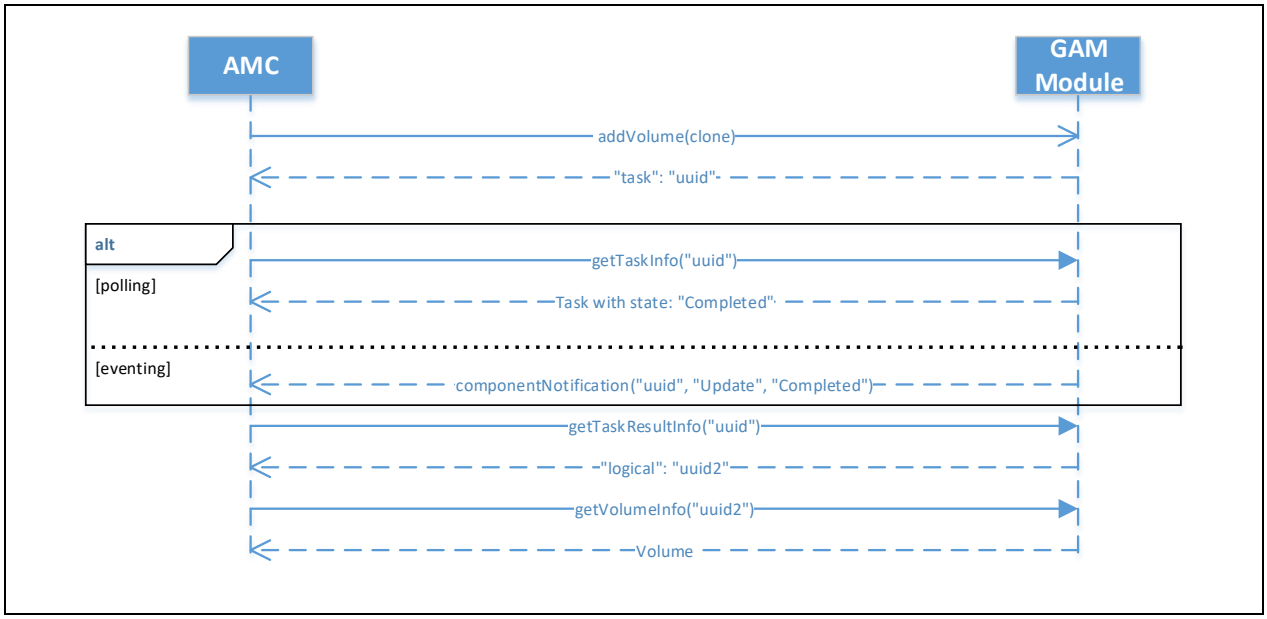
GAM modules can handle commands in two ways: synchronously returning action result immediately or asynchronously providing response containing created Task Universally Unique Identifier (UUID). Task is a special kind of component that controls and monitors the state of asynchronous actions. Synchrony or asynchrony of command depends on GAM module implementation.

When the GAM module receives a command, it decides to finish the action synchronously or asynchronously using the Task mechanism. If an action is completed asynchronously, the GAM module will return Task UUID. The command `getTaskInfo` is for checking the Task state and its additional properties. If Task's state is completed, the `getTaskResultInfo` command will return the UUID and type of component affected by the task.

There is no Add Task command because all Add, Delete, and Set Component Attributes commands can trigger a task mechanism and return created Task UUID instead of a typical response described later in this document - the AMC application must be ready for that.

Additionally, the GAM modules will use the notification mechanism to notify the AMC application about the Task's state change to avoid polling.

Figure 2. Sample Asynchronous Communication



Example of an asynchronous request and response exchange with the `"addVolume"` command.

Request:

```
{
        "jsonrpc": "2.0",
        "method": "addVolume",
        "params": {
                "volumeType": "RawDevice",
                "capacityBytes": 1000000000,
                "capacitySources": {
                        "providingPools": [ "b2fe60a7-33a8-63bb-da16-3e7b1a835ee5" ]
                },
                "accessCapabilities": [ "Read", "Write" ],
                "replicaInfos": [
                        {
                                "replicaType": "Clone",
                                "replica": "4e20b351-6732-ab36-121e-887efa325e4c"
                        }
                ],
                "bootable": true,
                "oem": {}
        },
        "id": 1
}
```

Response:

```
{
        "jsonrpc": "2.0",
        "result": {
```

```
                "task": "123e4567-e89b-12d3-a456-426655440000",
                "oem": {}
        },
        "id": 2
}
```

"getTaskInfo" **Request:**

```
{
        "jsonrpc": "2.0",
        "method": "getTaskInfo",
        "params": {
                "task": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 3
}
```

Response:

```
{
        "jsonrpc": "2.0",
        "result": {
                "name": "CloneVolume",
                "startTime": "2016-02-12T15:19:21+00:00",
                "endTime": "2016-02-12T17:19:21+00:00",
                "state": "Completed",
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "messages": [],
                "oem": {}
        },
        "id": 4
}
```

"getTaskResultInfo" **Result:**

```
{
        "jsonrpc": "2.0",
        "method": "getTaskResultInfo",
        "params": {
                "task": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 5
}
```

Response:

```
{
        "jsonrpc": "2.0",
        "result": {
                "uuid": "987e4567-n46j-34dx-lk78-jg8555440473",
                "type": "Volume",
                "oem": {}
        },
        "id": 6
}
```

"getVolumeInfo" **Request:**

```
{
        "jsonrpc": "2.0",
        "method": "getVolumeInfo",
```

```
        "params": {
                "volume": "987e4567-n46j-34dx-1k78-jg8555440473"
        },
        "id": 7
}
```

Response:

```
{
        "jsonrpc": "2.0",
        "result": {
                "name": "nvme3n0",
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "capacity": {
                        "consumedBytes": 234534527,
                        "allocatedBytes": 1099511627776,
                        "guaranteedBytes": 1099511627776,
                        "provisionedBytes": 1099511627776
                }
                "capacitySources": [
                        {
                                consumedBytes: 100849846,
                                allocatedBytes: 549755813888,
                                guaranteedBytes: 549755813888,
                                providingPools: [
                                                "b2fe60a7-33a8-63bb-da16-
3e7b1a835ee5"
                                ]
                        }
                ],
                "volumeType": "NonRedundant",
                "encrypted": false,
                "encryptionTypes": [],
                "identifiers": [],
                "blockSizeBytes": 512,
                "optimumIoSizeBytes": 4096,
                "accessCapabilities": [ "Read", "Write" ],
                "bootable": false,
                "replicaInfos": [
                        {
                                "replicaReadOnlyAccess": "SourceElement",
                                "replicaType": "Clone",
                                "replicaRole": "Target",
                                "replica": "4e20b351-6732-ab36-121e-887efa325e4c",
                        }
                ],
                "collections": [
                        {
                                "name": "Endpoints",
                                "type": "Endpoints"
                        }
                ]
                "oem": {}
        },
        "id": 987
}
```

§

3.0 GAMI API Error Handling

This chapter contains descriptions of all error codes that may be returned by the JSON-RPC calls implemented in the GAMI interfaces.

3.1 API Error Response

The GAMI JSON-RPC API complies with JSON-RPC v2.0 specification. Each request may return an error response according to the JSON-RPC v2.0 error object given below:

Error:

```
{
        "jsonrpc": "2.0",
        "error": {
                "code": <error code>,
                "data": <additional error data>,
                "message": <error message>
        },
        "id": id
}
```

3.1.1 Example Error JSON Object

Error:

```
{
        "jsonrpc": "2.0",
        "error": {
                "code": -32602,
                "data": <additional error data>,
                "message": "No such system"
        },
        "id": "456"
}
```

3.2 API Error Codes

This section describes the error codes returned by the JSON-RPC API.

3.2.1 General Error Codes

The error codes described by JSON-RPC v2.0 specification are listed in Table 3. Any command of GAMI protocol described in Section 5.0, GAMI API Definition may return any of these codes.

Table 3. General Error Codes

Error Code	Message	Description
-32700	Parse Error	The server received invalid JSON. An error occurred on the server while parsing the JSON text.
-32603	Internal Error	Internal JSON-RPC error.
-32602	Invalid Parameters	Invalid method parameter(s).
-32601	Method not found	The method does not exist/is not available.
-32600	Invalid Request	The JSON sent is not a valid Request object.

Error Code	Message	Description
-1	Unknown Error	Unknown error occurred.
10000	Not found	The requested component could not be found.
11000	Not implemented	The method is not implemented.
12000	Method not allowed	The method is not allowed.
13000	Method conflict	Used when method conflicts with ongoing asynchronous action.
14000	Invalid value	The JSON contains an invalid value for some parameter. This is a generic value related error code used when none of the remaining 14xxx matches.
14100	Unsupported value	The value is not supported.
14200	Invalid collection	The collection given is invalid.
14300	Invalid enum	The value is out of the supported values list.
14400	Invalid value format	The value has an incompatible format.
14500	Invalid UUID	The component with the requested UUID could not be found.
15000	Invalid field	The JSON contains an invalid field name. This is a generic property related error code used when none of the remaining 15xxx matches.
15100	Unsupported field	The field is not supported.
15200	Duplicated field	The field has already been provided.
15300	Invalid field type	The field has an incompatible type.
15400	Unexpected field	The field is unexpected.
15500	Missing field	The field is missing.
20000	Compute Error	Set of errors concerning Compute module.
21000	IPMI Error	Set of errors concerning IPMI.
30000	Network Error	Set of errors concerning Network module.
40000	Storage Error	Set of errors concerning Storage module.
41000	iSCSI Error	Set of errors concerning iSCSI.
42000	LVM Error	Set of errors concerning LVM.
43000	NVMe Error	Set of errors concerning NVMe.
44000	SPDK Error	Set of errors concerning SPDK.
50000	Fabric Error	Set of errors concerning Fabric module.
60000	Chassis Error	Set of errors concerning Chassis module.
61000	Certificate Error	Set of errors concerning certificate problems from Chassis module.
70000	FPGA-oF Error	Set of errors concerning FPGA over Fabrics.

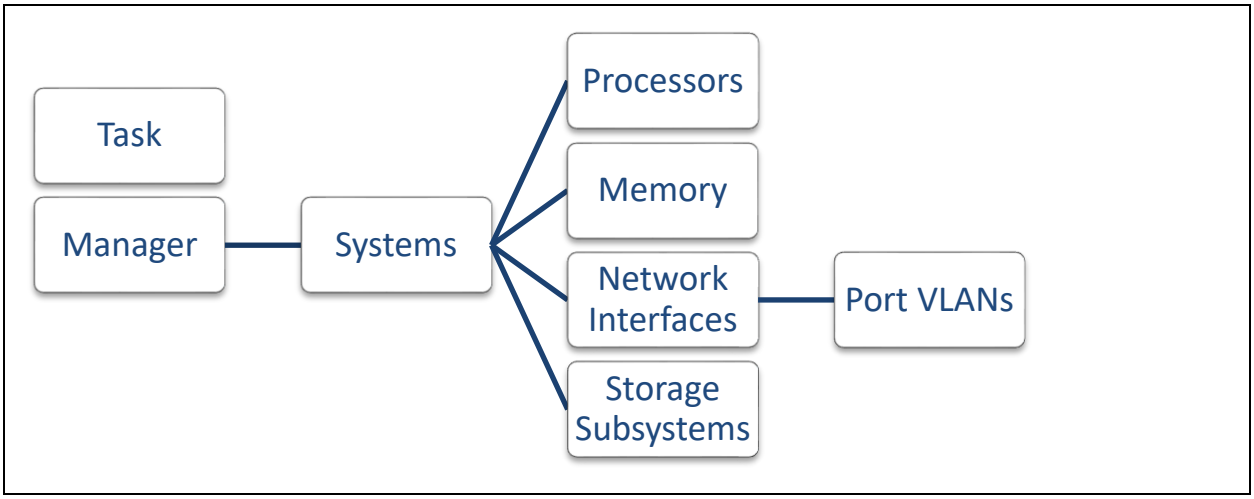
§

4.0 GAMI Data Model

The following sections describe the data models exposed by various GAMI modules.

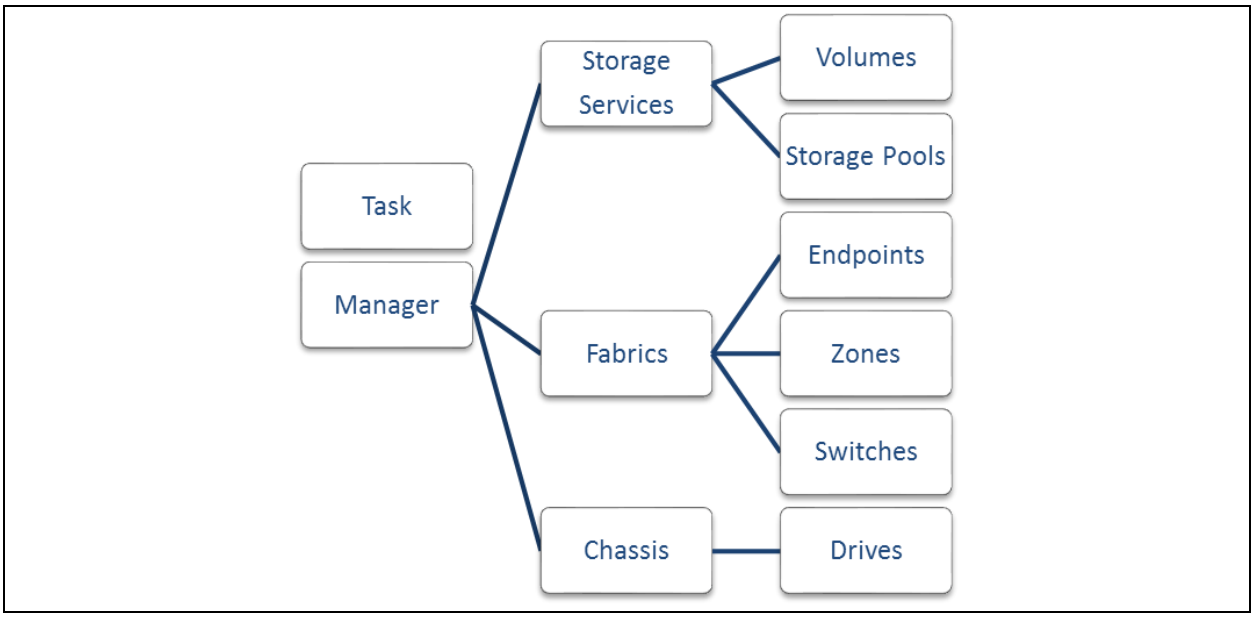
4.1 Computer System Management Module

Figure 3. Computer System Management Data Model



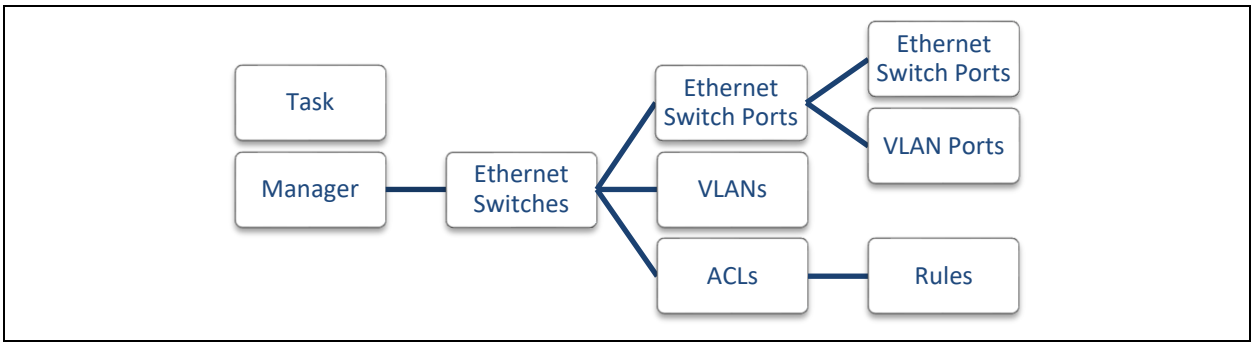
4.2 Storage Services Management Module

Figure 4. Storage Services Management Data Model



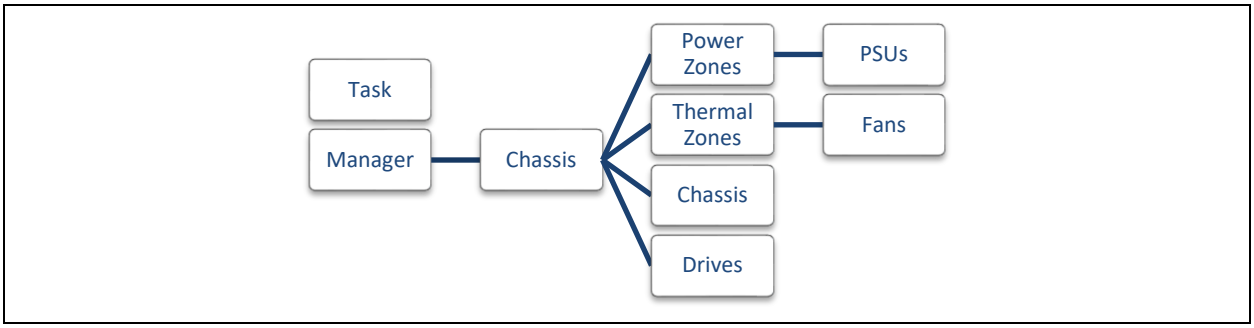
4.3 Network Management Model

Figure 5. Network Management Data Model



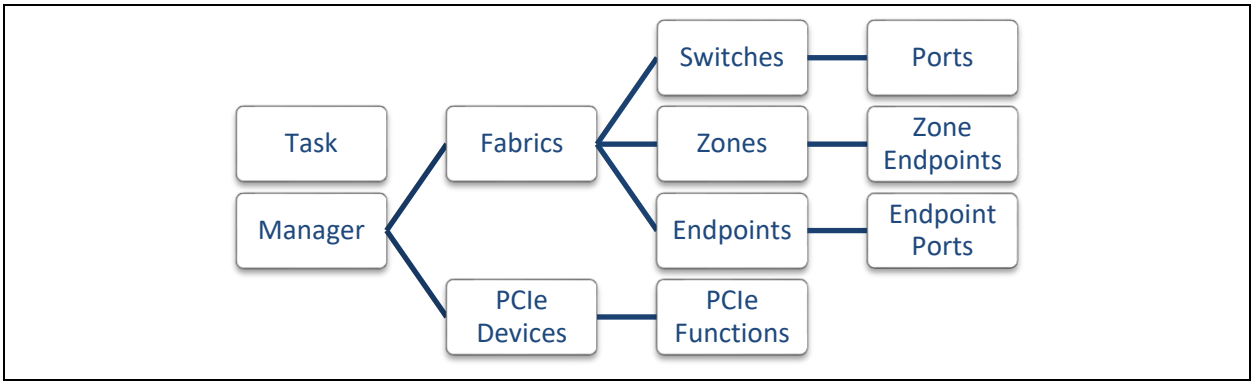
4.4 Chassis Management Model

Figure 6. Chassis Management Data Model



4.5 PCIe* Fabric Management Model

Figure 7. PCIe* Fabric Management Data Model



4.6 Subcomponent Collections

The GAMI data model is a hierarchical model with a strict hierarchy of components and subcomponents. In general, any component may have multiple subcomponents of multiple types. These subcomponents are grouped within the subcomponent collections according to its type.

Every component with at least one collection of subcomponents contains the unique property object that defines these collections. This property is shown in a table with entries defining the single collection. The collection definition consists of two fields as presented in Table 4.

Table 4. GAMI Subcomponent Collection Definition

Attribute	Type	Mandatory	Description
`name`	String	Yes	Collection name; unique collection name used to identify the particular collection of subcomponents; this name is used to retrieve the collection content (identifiers of subcomponents grouped within the collection).
`type`	String	Yes	Collection type; type of the subcomponents building the particular collection.

Every collection is specified as a table with identifiers (such as the UUID) of all subcomponents establishing a given collection. The subcomponent identifier may be used to retrieve its detailed property, using a dedicated command dependent on the collection type. For example, for the collection of `"Processors"` use the `"getProcessorInfo"` command to retrieve detailed information about a given processor identified by the identifier provided as a part of subcomponent collection. The GAMI defines the single command that retrieves a collection. This command accepts the identifier of the component, and the collection name responds with a table of subcomponent identifiers. This is the universal command to get all collection in the GAMI protocol. The `"getCollection"` command is described in Section 5.11, getCollection.

4.7 Mandatory and Optional Properties

The JSON structures exchanged through the GAMI contain the fields defined as `"Mandatory"` or `"Optional."` This information is necessary for the receiver of a JSON structure to accurately interpret the received data.

- If a field is specified as `"Mandatory",` it means that the JSON structure with one or more missing `"Mandatory"` fields is not compliant with the GAMI specification.
- If a field is specified as `"Optional",` it means the JSON stricture without such a field is still compliant with GAMI specification.

Here are a few ways the `"Optional"` field may be specified in the JSON request structure:

1. Field of the `"Number"` type may be:
   a. Specified: <number value>
   b. Set to `"null".`
2. Field of the `"Boolean"` type:
   a. Specified: `"true"` or `"false"`
   b. Set to `"null".`
3. Field of the `"String"` type:
   a. Specified: `<string>`
   b. Specified as an empty string: `""`
   c. Set to `"null".`

Note: The `"Optional"` field in the request JSON structure shall exist; only its **value** is optional.

The `"Optional"` field in the response JSON structure may be either entirely avoided, or specified in the same way as in the request JSON structure.

§

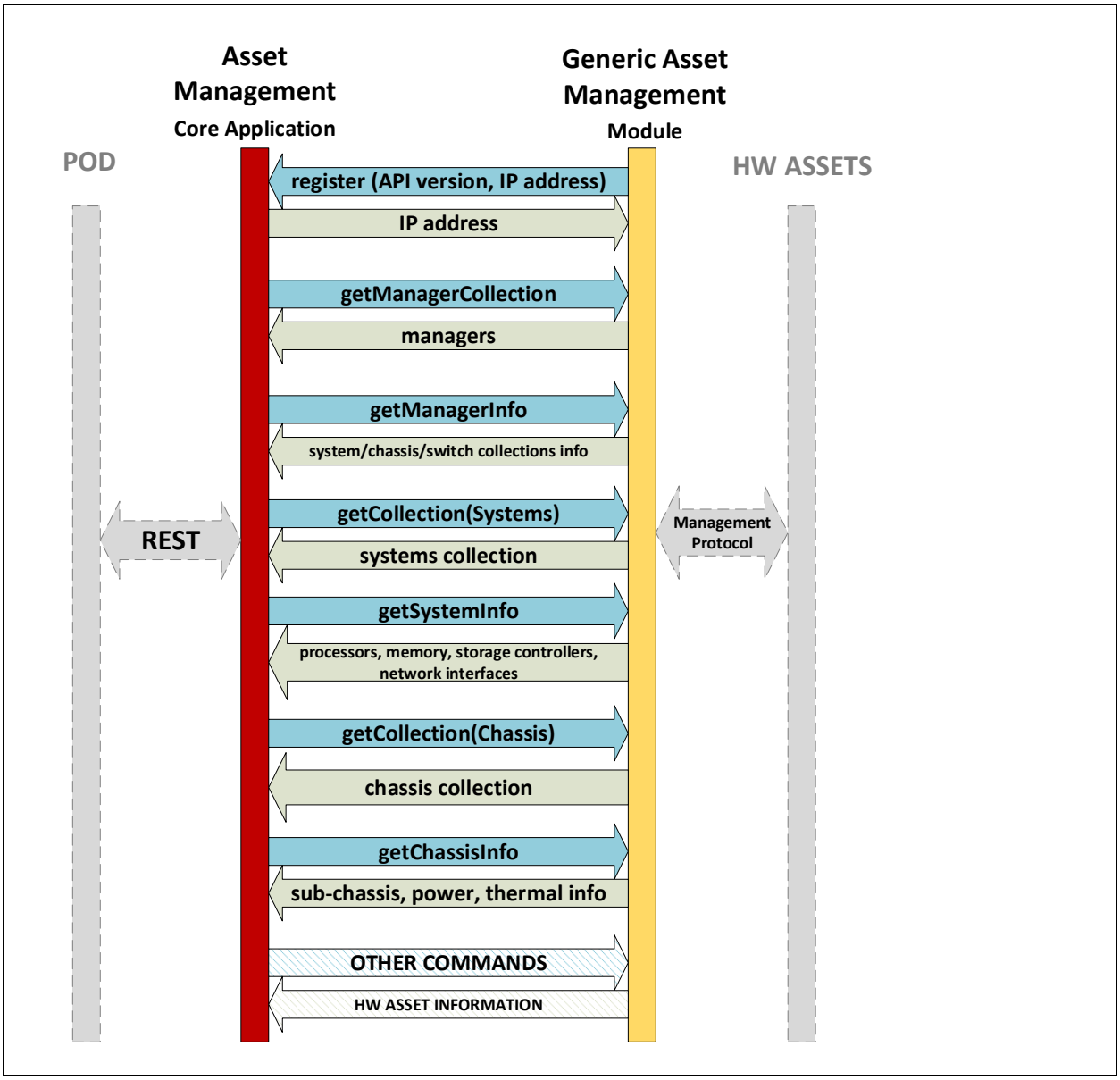
5.0 GAMI API Definition

This section contains the description of the commands provided by the GAMI protocol.

5.1 GAM Module Registration

GAM Module must register to the AMC application service to establish a communication channel. The registration procedure is performed using the JSON-RPC request sent to the preconfigured AMC application TCP/IP port. Figure 8 shows an example of the GAM module registration command sequence.

Figure 8. PSME Generic Asset Management Interface API Communication Overview



5.1.1 Request

The GAM Module "attach"the GAM module sends command to the AMC. The application registers the GAM module to the AMC application.

Table 5. GAM "attach" Registration Request

Parameter	Type	Mandatory	Description
version	String	Yes	GAMI API version
vendor	String	No	Vendor name string
ipv4address	String	Yes	IPv4 address the GAM module is opening for incoming connections
port	Number	Yes	Port number for incoming connections
gamiId	String	Yes	GAM module unique identifier
capabilities	Array: String	Yes	Capability name: "Compute", "Network", "Storage", "Chassis", "Pnc", "Rmm", "NVMe"

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "attach",
        "params": {
                "version": <string>,
                "vendor": <string>,
                "ipv4address": <string>,
                "port": <number>,
                "gamiId": <string>,
                "capabilities": [
                        <{"Compute", "Network", "Storage", "Chassis", "Pnc", "Rmm",
"NVMe"}>,
                        ...
                ]
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "attach",
        "params": {
                "version": "1.0.0",
                "vendor": "Intel Corporation",
                "ipv4address": "127.0.0.1",
                "port": "8080"
                "gamiId": "40bf1aa4-c440-11e5-9202-00a0c923456f",
                "capabilities": [
                        "Compute",
                        "Chassis"
                ]
        },
        "id": 345
}
```

5.1.2 Response

Upon a successful GAM module registration, the AMC application responds with the following response.

Table 6. GAM Module Registration Response

Result	Type	Mandatory	Description
version	String	Yes	GAMI API version
ipv4address	String	Yes	IPv4 address where the AMC Application is listening for requests and notifications from a newly registered GAM module.
port	Number	Yes	TCP port number AMC Application is listening on

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "version": <string>,
                "ipv4address": <string>,
                "port": <port>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "version": "1.0.0",
                "ipv4address": "127.0.0.1",
                "port": 8090
        },
        "id": 345
}
```

5.2 heartBeat

The GAM module sends the "heartBeat" command to the AMC application periodically to detect its status. If the GAM module does not receive a response for the "heartBeat" command, it means the AMC application is not running. If a timestamp returned by the AMC application in response is lower than a timestamp returned in the previous command, this indicates the AMC application has been restarted, since the heartbeat timer is restarted along with AMC application. In both cases, the GAM module will reset its state and start a new registration procedure.

5.2.1 Request

Table 7. heartBeat Command Request

Parameters	Type	Mandatory	Description
gamiId	String	Yes	GAM module unique identifier

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "heartBeat",
        "params": {
                "gamiId": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "heartBeat',
        "params": {
                 "gamiId": "40bf1aa4-c440-11e5-9202-00a0c923456f",
        },
        "id": 345
}
```

5.2.2 Response

Table 8. heartBeat Command Response

Parameters	Type	Mandatory	Description
timeStamp	Number	Yes	Timestamp returned by the AMC Application.
minDelay	Number	Yes	The minimum delay after which the GAM module can send next "heartBeat" command to the AMC Application.

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                 timeStamp: <number>,
                 minDelay: <number>
        },
        "id": 334
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                 timeStamp: 5443323,
                 minDelay: 5000
        },
        "id": 345
}
```

5.3 getManagersCollection

The AMC retrieves collection of main managers supported by the particular GAM Module by sending the "getManagersCollection" command to this GAM Module.

5.3.1 Request

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getManagersCollection",
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getManagersCollection".
        "id": 987
}
```

5.3.2 Response

GAMI Module replies with the following response in a successful case:

Table 9. getManagersCollection Response

Parameters	Type	Mandatory	Description
managers	Array: Object	Yes	

Member	Type	Mandatory	Description
manager	String	Yes	Manager UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": [
                {
                        "manager": <string>
                },
                ...
        ],
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": [
                {
                        "manager": "123e4567-e89b-12d3-a456-426655440000",
                },
                {
                        "manager": "863e4567-e87b-64d3-a489-987656540000",
                }
        ],
        "id": 987
}
```

5.4 getManagerInfo

The "getManagerInfo" command retrieves detailed information about a single manager.

5.4.1 Request

Table 10. getManager Info Request

Parameters	Type	Mandatory	Description
manager	String	Yes	Manager UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getManagerInfo",
        "params": {
                "manager": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getManagerInfo",
        "params": {
                "manager": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.4.2 Response

Table 11. getManagerInfo Response

Result	Type	Mandatory	Description
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare", "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr><tr><td>health</td><td>String</td><td>No</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
type	String	Yes	Manager type: "ManagementController", "EnclosureManager", "BMC", "RackManager", "EthernetSwitchManager", "StorageSystemManager", "AuxiliaryController"
name	String	No	Optional name of the component
description	String	No	Optional description of the component
model	String	No	Manager model
location	String	No	UUID of the chassis being a manager's physical location
guide	String	No	GUID of the management controller

Result	Type	Mandatory	Description
parentId	String	No	Identifier of the parent manager
dateTime	String	No	The current dateTime (with offset) for the manager, used to set or read time compatible with *Date and Time on the Internet: Timestamps, RFC 3339*, refer to Table 2.
dateTimeLocalOffset	String	No	The time offset from UTC that the DateTime property is set to in format: +06:00.
autoDSTEnabled	Boolean	No	Indicates whether automatic Daylight Saving Time (DST) adjustment of the manager's DateTime is enabled
firmwareVersion	String	Yes	Manager firmware version
ipv4Address	String	No	IPv4 address manager is listening on
networkServices	Array: Object	No	

Attribute	Type	Mandatory	Description
name	String	Yes	Protocol: "HTTP", "HTTPS", "SNMP", "VirtualMedia", "Telnet", "SSDP", "IPMI", "SSH", "KVMIP", "DHCP", "NTP"
port	Number	No	Service port number
enabled	Boolean	Yes	Is service enabled

graphicalConsole	Object	No	

Attribute	Type	Mandatory	Description
enabled	Boolean	No	Console availability
maxSessions	Number	No	The number of sessions that can be established at the same time.
typesSupported	Array: String	No	Supported types of connections

serialConsole	Object	Yes	

Attribute	Type	Mandatory	Description
signalType	String	No	Signal type: "Rs232", "Rs485"
bitrate	Number	No	Bitrate: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400
parity	String	No	Parity: "None", "Even", "Odd", "Mark", "Space"
dataBits	Number	No	Data bits: 5, 6, 7, 8
stopBits	Number	No	Stop bits: 1, 2
signalType	String	No	Signal type: "Rs232", "Rs485"

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>bitrate</td><td>Number</td><td>No</td><td>Bitrate:
1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400</td></tr>
<tr><td>parity</td><td>String</td><td>No</td><td>Parity:
"None", "Even", "Odd", "Mark", "Space"</td></tr>
<tr><td>dataBits</td><td>Number</td><td>No</td><td>Data bits:
5, 6, 7, 8</td></tr>
<tr><td>stopBits</td><td>Number</td><td>No</td><td>Stop bits:
1, 2</td></tr>
<tr><td>flowControl</td><td>String</td><td>No</td><td>Flow control:
"None", "Software", "Hardware"</td></tr>
<tr><td>pinOut</td><td>String</td><td>No</td><td>Pin out:
"Cisco", "Cyclades", "Digi"</td></tr>
<tr><td>enabled</td><td>Boolean</td><td>No</td><td>Console availability</td></tr>
<tr><td>maxSessions</td><td>Number</td><td>No</td><td>The number of sessions that can be established at the same time.</td></tr>
<tr><td>typesSupported</td><td>Array: String</td><td>No</td><td>Supported types of connections</td></tr>
</table>
</td></tr>
<tr><td>commandShell</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>enabled</td><td>Boolean</td><td>No</td><td>Console availability</td></tr>
<tr><td>maxSessions</td><td>Number</td><td>No</td><td>The number of sessions that can be established at the same time.</td></tr>
<tr><td>typesSupported</td><td>Array: String</td><td>No</td><td>Supported types of connections.</td></tr>
</table>
</td></tr>
<tr><td>collections</td><td>Array: Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>name</td><td>String</td><td>Yes</td><td>Collection name</td></tr>
<tr><td>type</td><td>String</td><td>Yes</td><td>Collection type:
"Chassis", "Systems", "EthernetSwitches", "StorageServices", "Managers", "Fabrics", "PCIeDevices", "NetworkInterfaces"</td></tr>
</table>
</td></tr>
<tr><td>allowedActions</td><td>Array: String</td><td>No</td><td>Allowable reset types: Allowed values: "On", "ForceOff", "GracefulShutdown", "GracefulRestart", "ForceRestart", "Nmi", "ForceOn", "PushPowerButton"</td></tr>
<tr><td>oem</td><td>Object</td><td>No</td><td>OEM specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "type": <{"ManagementController", "EnclosureManager", "BMC",
"RackManager", "EthernetSwitchManager", "StorageSystemManager",
"AuxiliaryController"}>,
                "model": <string>,
                "location": <string>,
                "guid": <string>,
                "parentId": <string>,
                "dateTime": <string>,
                "dateTimeLocalOffset": <string>,
                "firmwareVersion": <string>,
                "ipv4Address": <string>,
                "networkServices": [
                        {
                                "name": <{"HTTP", "HTTPS", "SNMP", "VirtualMedia",
"Telnet", "SSDP", "IPMI", "SSH", "KVMIP", "DHCP", "NTP"}>,
                                "port": <number>,
                                "enabled": <boolean>
                        },
                        ...
                ],
                "graphicalConsole": {
                        "enabled": <boolean>,
                        "maxSessions": <number>,
                        "typesSupported": [
                                <{"KVMIP"}>
                        ...
                        ],
                }
                "serialConsole": {
                        "signalType": <string>,
                        "bitrate": <number>,
                        "parity": <string>,
                        "dataBits": <number>,
                        "stopBits": <number>,
                        "flowControl": <string>,
                        "pinOut": <string>,
                        "enabled": <boolean>,
                        "maxSessions": <number>,
                        "typesSupported": [
                                <{"Telnet", "IPMI", "SSH"}>
                        ...
                        ],
                },
                "commandShell": {
                        "enabled": <boolean>,
                        "maxSessions": <number>,
                        "typesSupported": [
                                <{"VirtualMedia", "Telnet", "SSH"}>
                        ...
                        ],
                }
                "collections": [
```

```
                        {
                                "name": <string>,
                                "type": <{"Chassis", "Systems", "EthernetSwitches",
"StorageServices", "Managers", "Fabrics", "PCIeDevices", "Drives"}>
                        }
                ...
                ],
                "allowedActions": [{"On", "ForceOff", "GracefulShutdown",
"GracefulRestart", "ForceRestart", "Nmi", "ForceOn", "PushPowerButton"}],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "type": "BMC",
                "model": "Simple IPMI compliant",
                "location": "123e4567-e89b-12d3-a456-426655440000",
                "guid": "030B4A82-1B7C-11CF-9D53-00AA003C9CB6",
                "parentId": "85a53567-349b-d293-4ab6-342655440ab3",
                "dateTime": "2015-03-13T04:14:33+06:00",
                "dateTimeLocalOffset": "+06:00",
                "firmwareVersion": "2.58",
                "ipv4Address": "1.1.2.1",
                "networkServices": [
                        {
                                "name": "IPMI",
                                "port": 623,
                                "enabled": true
                        }
                ],
                "graphicalConsole": {
                        "enabled": true,
                        "maxSessions": 2,
                        "typesSupported": ["KVMIP"]
                }
                "serialConsole": {
                        "signalType": "Rs232",
                        "bitrate": 115200,
                        "parity": "None",
                        "dataBits": 8,
                        "stopBits": 1,
                        "flowControl": "None",
                        "pinOut": "Cisco",
                        "enabled": true,
                        "maxSessions": 1,
                        "typesSupported": ["Telnet", "IPMI", "SSH"]
                },
                "commandShell": {
                        "enabled": true,
                        "maxSessions": 1,
                        "typesSupported": ["Telnet", "SSH"]
                }
```

```
                "collections": [
                        {
                                "name": "Fabrics",
                                "type": "Fabrics"
                        },
                        {
                                "name": "Devices",
                                "type": "PCIeDevices"
                        },
                                "name": "Chassis",
                                "type": "Chassis"
                        }
                ],
                "allowedActions": "ForceOff",
                "oem": {}
        },
        "id": 123
}
```

5.5 setComponentAttributes

The `"setComponentAttributes"` command sets a new value for one or more attributes of a single component. The command is universal and may be used for various components.

Note: If some of the attributes are not set, the response will contain an error message about the command fails.

5.5.1 Request

Table 12. setComponentAttributes Request

Parameters	Type	Mandatory	Description
`component`	String	Yes	Managed component UUID
`attributes`	Object	Yes	Component attributes specified using the structure defined for the component properties retrieving using the Get ***<Component>*** Info command. The set of the configurable attributes is defined on the component basis later in this document.

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "setComponentAttributes",
        "params": {
                "component": <string>,
                "attributes": {
                        ...
                }
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "setComponentAttributes",
        "params": {
                "component": "123e4567-e89b-12d3-a456-426655440000",
                "attributes": {
```

```
                        "initiatorIQN": "iqn.2015-01.com:pod-1,rack-1,drawer-
1,module-3"
                }
        },
        "id": 987
}
```

5.5.2 Response

Table 13. setComponentAttributes Response

Parameters	Type	Mandatory	Description
data	Array: Object	No	If any of the attributes failed to be set, corresponding objects to be returned indicating attribute issue

Member	Type	Mandatory	Description
attribute	String	Yes	Name of the attribute
code	Number	Yes	Error code
message	String	Yes	Additional error message

Serialization:

```
{
        "jsonrpc": "2.0",
        "error": {
                "code": <Number>,
                "message": <String>,
                "data": [
                        {
                                "attribute": <string>,
                                "code": <error code>,
                                "message": <string>
                        },
                        ...
                ],
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "error": {
                "code": 14000,
                "message": "Some of the attributes could not be set",
                "data": [
                        {
                                "attribute": "initiatorIQN",
                                "code": 14001,
                                "message": "Binding already exists!"
                        },
                        {
                                "attribute": "",
                                "code": 14002,
                                "message": ""
                        }
```

```
                ]
        },
        "id": 987
}
```

5.6 Set Manager Attributes

The "setComponentAttributes" described in Section 5.5 setComponentAttributes allows configuration of the Set Manager Attributes listed in the following table.

Table 14. Configurable Manager Attributes

Attribute	Type	Description
reset	Boolean	Any write operation triggers reset action on the managed resource
factoryDefaults	Boolean	Any write operation causes loading factory default settings
packageUrl	String	Software location URL to perform simple update; May not be supported by all components.

5.7 getTasksCollection

AMC retrieves the collection of tasks currently handled by the particular GAM Module by sending the "getTasksCollection" command to the GAM Module.

5.7.1 Request

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getTasksCollection",
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getTasksCollection".
        "id": 987
}
```

5.7.2 Response

The GAMI Module replies with the following response in a successful case:

Table 15. getTasksCollection Response

Parameters	Type	Mandatory	Description
tasks	Array: Object	Yes	<table><tr><th>Member</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>task</td><td>String</td><td>Yes</td><td>UUID of the task.</td></tr></table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": [
                {
                        "task": <string>
                },
                ...
        ],
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": [
                {
                        "task": "123e4567-e89b-12d3-a456-426655440000",
                },
                {
                        "task": "863e4567-e87b-64d3-a489-987656540000",
                }
        ],
        "id": 987
}
```

5.8 getTaskInfo

The "getTaskInfo" command retrieves detailed information about a single task.

5.8.1 Request

Table 16. getTaskInfo Request

Parameters	Type	Mandatory	Description
task	String	Yes	UUID of the task.

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getTaskInfo",
        "params": {
                "task": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getTaskInfo",
        "params": {
                "task": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.8.2 Response

Table 17. getTaskInfo Response

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>name</td><td>String</td><td>Yes</td><td>Task name that identifies the asynchronous operation.</td></tr>
<tr><td>description</td><td>String</td><td>No</td><td>Optional description of the component</td></tr>
<tr><td>endTime</td><td>String</td><td>Yes</td><td>ISO 8601 timestamp string is indicating task end time.</td></tr>
<tr><td>state</td><td>String</td><td>Yes</td><td>Task state indication: "New", "Starting", "Running", "Suspended", "Interrupted", "Pending", "Stopping", "Completed", "Killed", "Exception", "Service".</td></tr>
<tr><td>status</td><td>Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component:
"Enabled",
"Disabled",
"StandbyOffline",
"StandbySpare,",
"InTest",
"Starting",
"Absent,
"UnavailableOffline"</td></tr>
<tr><td>health</td><td>String</td><td>No</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical."</td></tr>
</table>
</td></tr>
<tr><td>messages</td><td>Array: Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>messageId</td><td>String</td><td>Yes</td><td>Key into message registry as described in the Redfish specification, refer to Table 2.</td></tr>
<tr><td>messageContent</td><td>String</td><td>Yes</td><td>Human readable message</td></tr>
<tr><td>severity</td><td>String</td><td>No</td><td>The severity of the error</td></tr>
<tr><td>resolution</td><td>String</td><td>No</td><td>Suggested resolution</td></tr>
<tr><td>relatedProperties</td><td>Object</td><td>No</td><td>Optional JSON pointer array</td></tr>
<tr><td>messageArgs</td><td>Object</td><td>No</td><td>Message arguments</td></tr>
<tr><td>oem</td><td>Object</td><td>No</td><td>OEM specific data</td></tr>
</table>
</td></tr>
<tr><td>oem</td><td>Object</td><td>No</td><td>OEM specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "name": <string>,
                "startTime": <string>,
                "endTime": <string>,
                "state": <{"New", "Starting", "Running", "Suspended", "Interrupted",
"Pending", "Stopping", "Completed", "Killed", "Exception", "Service"}>,
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "name": "ForceShutdownManager",
                "startTime": "2016-02-12T15:19:21+00:00",
                "endTime": "2016-02-12T17:19:21+00:00",
                "state": "Completed",
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "oem": {}
        },
        "id": 123
}
```

5.9 deleteTask

The `"deleteTask"` command deletes existing task.

5.9.1 Request

Table 18. deleteTask Request

Parameters	Type	Mandatory	Description
`task`	String	Yes	Existing task UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "deleteTask",
        "params": {
                "task": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "deleteTask",
        "params": {
                "task": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.9.2 Response

Table 19. deleteTask Response

Result	Type	Mandatory	Description
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 987
}
```

5.10 getTaskResultInfo

The `"getTaskResultInfo"` retrieves the UUID and type of component affected by completed task.

5.10.1 Request

Table 20. getTask ResultInfo Request

Parameters	Type	Mandatory	Description
`task`	String	Yes	Existing task UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getTaskResultInfo",
        "params": {
                "task": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getTaskResultInfo",
        "params": {
                "task": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.10.2 Response

Table 21. getTaskResultInfo Response

Result	Type	Mandatory	Description
uuid	String	Yes	UUID of the component affected by the task.
type	String	Yes	Type of component affected by the task.
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "uuid": <string>,
                "type": <string>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "uuid" "123e4567-e89b-12d3-a456-426655440000",
                "type": "LogicalDrive",
                "oem": {}
        },
        "id": 987
}
```

5.11 getCollection

The AMC retrieves a collection of different types of subcomponents of the particular parent objects by sending the `"getCollection"` command to the appropriate GAM module.

5.11.1 Request

Table 22. getCollection Request

Parameters	Type	Mandatory	Description
component	String	Yes	Component UUID
name	String	No	Subcomponent collection name: The collection name is provided as part of the collection definition that is a property of the component.

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getCollection",
        "params": {
                "component": <string>,
                "name": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getCollection",
        "params": {
                "component": "123e4567-e89b-12d3-a456-426655440000",
                "name": "Systems"
        },
        "id": 987
}
```

5.11.2 Response

The GAM module replies with the following in a successful case:

Table 23. getCollection Response

Parameters	Type	Mandatory	Description
subcomponents	Array: Object	Yes	**Name**: subcomponents **Attribute** / **Type** / **Mandatory** / **Description**: subcomponent / String / Yes / Subcomponent UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": [
                {
                        "subcomponent": <string>
                },
                ...
        ],
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": [
                {
                        "subcomponent": "123e4567-e89b-12d3-a456-426655440734"
                },
                {
                        "subcomponent": "863e4567-e87b-64d3-a489-987656540000",
                }
        ],
```

```
        "id": 987
}
```

5.12 componentNotification

The GAM module sends the "componentNotification" command to the AMC to notify about changes in the managed infrastructure. This command covers notification about:

- Change of the existing component state and/or configuration
- Appearance of new components
- Disappearance of existing components.

This is the "componentNotification" command; no response is expected to be sent by the AMC.

5.12.1 Notification

Table 24. componentNotification

Parameters	Type	Mandatory	Description
gamiId	String	Yes	GAM module unique identifier
notifications	Array: Object	Yes	Notifications for each component.

Attribute	Type	Mandatory	Description
component	String	Yes	Component UUID
notification	String	Yes	Notification type: "Add", "Remove", "Update"
parent			Parent component UUID, (may be empty(null) if no parent component in the hierarchy, e.g., top chassis object)
type	String	Yes	Component type. For example: "Chassis"
timestamp	Number	Yes	Notification timestamp

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "componentNotification",
        "params": {
                "gamiId": <string>,
                "notifications": [
                        {
                                "component": <String>,
                                "notification": <{"Add", "Remove", "Update"}>,
                                "parent": <String>,
                                "type": <String>,
                                "timestamp": <Number>
```

```
                        }
                ]
        }
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "componentNotification",
        "params": {
                "gamiId": "816ddbd5-1919-4cc8-91c0-364016a0e104",
                "notifications": [
                        {
                                "component": "5a3bead2-2162-4746-8dfb-d95891df8c57",
                                "notification": "Update",
                                "parent": "f9db765f-ed60-46da-baff-9bfd0ab1423e",
                                "type": "Chassis",
                                "timestamp": 1506684342
                        },
                        {
                                "component": "0cdab0e9-df36-4ec5-a716-d94e77d6223f",
                                "notification": "Update",
                                "parent": "30b932f2-3ec1-4fbc-9265-2cd6c8a66d58",
                                "type": "Drive",
                                "timestamp": 1506684528
                        }
                ]
        }
}
```

5.13 getComputerSystemInfo

The "getComputerSystemInfo" command retrieves detailed information about a single computer system.

5.13.1 Request

Table 25. getComputerSystemInfo Request

Parameters	Type	Mandatory	Description
system	String	Yes	Managed computer system UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getComputerSystemInfo",
        "params": {
                "system": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getComputerSystemInfo",
        "params": {
                "system": "123e4567-e89b-12d3-a456-426655440000"
        },
```

```
        "id": 234
}
```

5.13.2 Response

Table 26. getComputerSystemInfo Response

Results	Type	Mandatory	Description
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr><tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource: "OK", "Warning", "critical."</td></tr></table>
type	String	No	The type of computer system represented by this resource: "Physical", "Virtual", "OS", "PhysicallyPartitioned", "VirtuallyPartitioned".
name	String	No	Optional name of the component
description	String	No	Optional description of the component
biosVersion	String	No	BIOS version
bootSourceOverrideMode	String	Yes	The BIOS Boot Mode (either Legacy or UEFI) to be used when BootSourceOverrideTarget boot source is booted from: "Legacy", "UEFI"
bootOverride	String	Yes	Boot source override state: "Disabled", "Once", "Continuous"
bootOverrideTarget	String	Yes	Boot source override target: "None", "Pxe", "Floppy", "Cd", "Usb", "Hdd", "BiosSetup", "Utilities", "Diags", "UefiTarget", "RemoteDrive"
bootOverrideSupported	Array: String	Yes	Boot source override supported targets: "None", "Pxe", "Floppy", "Cd", "Usb", "Hdd", "BiosSetup", "Utilities", "Diags", "UefiTarget", "RemoteDrive"
uefiTarget	String	No	Uefi Device Path of the device to boot from when: BootSourceOverrideTarget is UefiTarget.
powerState	String	No	The power state of the system: "On", "Off"
pciDevices	Array: Object	No	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>vendorId</td><td>String</td><td>Yes</td><td>String with 4 digits hex number</td></tr><tr><td>deviceId</td><td>String</td><td>Yes</td><td>String with 4 digits hex number</td></tr></table>

<table>
<tr><th>Results</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>pciFunctions</td><td>Array: Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>vendorId</td><td>String</td><td>Yes</td><td>String with 4 digits hex number</td></tr>
<tr><td>deviceId</td><td>String</td><td>Yes</td><td>String with 4 digits hex number</td></tr>
</table>
</td></tr>
<tr><td>usbDevices</td><td>Array: Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>vendorId</td><td>String</td><td>Yes</td><td>String with 4 digits hex number</td></tr>
<tr><td>deviceId</td><td>String</td><td>Yes</td><td>String with 4 digits hex number</td></tr>
</table>
</td></tr>
<tr><td>trustedExecutionTechnologyEnabled</td><td>Boolean</td><td>No</td><td>Reports if Trusted Execution Technology is enabled or disabled</td></tr>
<tr><td>userModeEnabled</td><td>Boolean</td><td>No</td><td>Determines whether user mode is enabled or not</td></tr>
<tr><td>fruInfo</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>serialNumber</td><td>String</td><td>No</td><td>Module serial number</td></tr>
<tr><td>manufacturer</td><td>String</td><td>No</td><td>Manufacturer name</td></tr>
<tr><td>modelNumber</td><td>String</td><td>No</td><td>Model number</td></tr>
<tr><td>partNumber</td><td>String</td><td>No</td><td>Part number</td></tr>
</table>
</td></tr>
<tr><td>sku</td><td>String</td><td>No</td><td>System SKU</td></tr>
<tr><td>assetTag</td><td>String</td><td>No</td><td>Asset tag</td></tr>
<tr><td>systemGuid</td><td>String</td><td>No</td><td>System GUID</td></tr>
<tr><td>indicatorLED</td><td>String</td><td>No</td><td>The state of the indicator LED used to identify the drive: "Lit", "Blinking", "Off".</td></tr>
<tr><td>currentPerformanceConfiguration</td><td>Number</td><td>No</td><td>index of the currently active configuration from the list of configurations</td></tr>
<tr><td>performanceConfigurations</td><td>Array: Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>configurationId</td><td>Number</td><td>Yes</td><td>Unique Control value that identifies this specific configuration</td></tr>
<tr><td>type</td><td>String</td><td>Yes</td><td>Type of current configuration. Supported values: "StaticSpeedSelect" and "PrioritizedBaseFrequency"</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Results</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>highPriorityCoreCount</td><td>Number</td><td>No</td><td>Identifies the configurable maximum number of priority cores in this configuration</td></tr>
<tr><td>highPriorityBaseFrequency</td><td>Number</td><td>No</td><td>The base Frequency corresponding to the number of High Priority Cores in this configuration</td></tr>
<tr><td>lowPriorityCoreCount</td><td>Number</td><td>No</td><td>Identifies the number of Low Priority Cores in this configuration</td></tr>
<tr><td>lowPriorityBaseFrequency</td><td>Number</td><td>No</td><td>Identifies the base frequency of the remaining lower priority cores</td></tr>
<tr><td>activeCores</td><td>Number</td><td>No</td><td>Number of active cores per CPU</td></tr>
<tr><td>baseCoreFrequency</td><td>Number</td><td>No</td><td>Selected frequency for core</td></tr>
<tr><td>tdp</td><td>Number</td><td>No</td><td>Thermal Design Power in Watts for this configuration Thermal Design Power in Watts for this configuration</td></tr>
<tr><td>maxJunctionTempCelsius</td><td>Number</td><td>No</td><td>Maximum temperature allowed for the CPU in this configuration</td></tr>
</table>
</td></tr>
<tr><td>collections</td><td>Array: Object</td><td>Yes</td><td>Subcomponents collections
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>name</td><td>String</td><td>Yes</td><td>Collection name</td></tr>
<tr><td>type</td><td>String</td><td>Yes</td><td>Collection type:
"Processors", "Memory",
"MemoryDomains",
"StorageSubsystems",
"NetworkInterfaces",
"TrustedModules"
"NetworkDevices"</td></tr>
</table>
</td></tr>
<tr><td>chassis</td><td>String</td><td>Yes</td><td>UUID of the chassis is the physical container for the system</td></tr>
<tr><td>cableIds</td><td>Array: String</td><td>No</td><td>List of cable IDs attached to this computer system.</td></tr>
<tr><td>oem</td><td>Object</td><td>No</td><td>OEM specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
```

```
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "type": <{"Physical", "Virtual", "OS", "PhysicallyPartitioned",
"VirtuallyPartitioned".}>,
                "biosVersion" <string>,
                "bootSourceOverrideMode": <{"Legacy", "UEFI"}>,
                "bootOverride": <{"Disabled", "Once", "Continuous"}>,
                "bootOverrideTarget": <{"None", "Pxe", "Floppy", "Cd", "Usb", "Hdd",
"BiosSetup", "Utilities", "Diags", "UefiTarget", "RemoteDrive"}>,
                "bootOverrideSupported": [
                        <{"Pxe", "Floppy", "Cd", "Usb", "Hdd", "BiosSetup",
"Utilities", "Diags", "UefiTarget",  "RemoteDrive"}>,
                        ...
                ],
                "uefiTarget": <string>,
                "powerState": <{"On", "Off"}>,
                "pciDevices": [
                        {
                                "vendorId" : <string>,
                                "deviceId" : <string>
                        },
                        ...
                ],
                "pciFunctions": [
                        {
                                "vendorId" : <string>,
                                "deviceId" : <string>
                        },
                        ...
                ],

                "usbDevices": [
                        {
                                "vendorId" : <string>,
                                "deviceId" : <string>
                        },
                        ...
                ],
                "trustedExecutionTechnologyEnabled": <Boolean>,
                "userModeEnabled": <Boolean>,
                "fruInfo": {
                        "serialNumber": <string>,
                        "manufacturer": <string>,
                        "modelNumber": <string>,
                        "partNumber": <string>
                },
                "sku": <string>,
                "assetTag": <string>,
                "indicatorLED": <{"Lit", "Blinking", "Off"}>,
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"Processors", "Memory", "MemoryDomains",
"StorageSubsystems", "NetworkInterfaces", "TrustedModules", "NetworkDevices"}>
                        }
                        ...
                ],
        "chassis": <string>,
        "cableIds": [
```

```
                <string>
        ],
        "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "type": "Physical",
                "biosVersion": "A20F21_A0",
                "bootSourceOverrideMode": "Legacy",
                "bootOverride": "Continuous",
                "bootOverrideTarget": "Pxe",
                "bootOverrideSupported": [
                        "Pxe", "Cd", "Usb", "Hdd", "BiosSetup", "Utilities", "Diags"
                ]
                "powerState": "On",
                "pciDevices": [
                        {
                                "vendorId" : "0x8086"
                                "deviceId" : "0x1234"
                        }
                ],
                "usbDe                                                vices": [
                        {
                                "vendorId" : "0x8086",
                                "deviceId" : "0x5678"
                        }
                ],
                "trustedExecutionTechnologyEnabled": false,
                "userModeEnabed": true,
                "fruInfo": {
                        "serialNumber": "123fed3029c-b23394-12",
                        "manufacturer": "Intel Corporation",
                        "modelNumber": "E323",
                        "partNumber": "29ee2220939"
                },
                "sku": "System SKU",
                "assetTag": "User defined asset tag",
                "indicatorLED": "Lit",
                "collections": [
                        {
                                "name": "Processors",
                                "type": "Processors"
                        },
                        {
                                "name": "Memory",
                                "type": "Memory"
                        },
                        {
                                "name": "MemoryDomains",
                                "type": "MemoryDomains"
                        },
```

```
                        {
                                "name": "StorageSubsystems",
                                "type": "StorageSubsystems"
                        },
                        {
                                "name": "NetworkInterfaces",
                                "type": "NetworkInterfaces"
                        }
                ],
                "chassis": "123e4567-e89b-12d3-a456-426655440000",
                "cableIds": ["189237", "hg7234"],
                "oem": {}
        },
        "id": 234
}
```

5.14 Set Computer System Attributes

The "setComponentAttributes" described in Section 5.5, setComponentAttributes, allows configuration of the Computer System Attributes listed in the following table.

Table 27. Configurable Computer System Attributes

Attribute	Type	Description
bootOverride	String	Boot source override state: "Disabled", "Once", "Continuous"
bootOverrideTarget	String	Boot source override target: "None", "Pxe", "Cd", "Usb", "Hdd", "BiosSetup", "Utilities", "Diags" The Pxe indicates to the PXE boot; CD, USB, and Hdd to boot from their devices respectively; BIOS Setup indicates to boot to the native BIOS screen setup; Utilities and Diagnostics indicate to boot from the local utilities or diagnostic partitions.
bootOverrideMode	String	Boot source override mode: "Legacy", "UEFI".
reset	String	Power state of the system: "On", "ForceOff", "GracefulShutdown", "ForceRestart", "Nmi", "ForceOn", "PushPowerButton"
assetTag	String	Tag assigned to the asset.
trustedExecutionTechnologyEnabled	Boolean	Enables or disables TXT on the platform
userModeEnabled	Boolean	Allows to switch between user mode FW upgrade of system components disabled) and admin mode (FW upgrade enabled)
currentPerformanceConfiguration	String	Unique Control value that identifies a specific configuration to be set
resetConfiguration	Boolean	Indicates that the Intel Optane memory Platform Configuration Data should be overwritten.
eraseConfigurationKeys	Boolean	Indicates that the Intel Optane memory configuration keys should be securely erased.
oem	Object	OEM specific data

5.15 getProcessorInfo

The "getProcessorInfo" command retrieves detailed information about a single processor.

5.15.1 Request

Table 28. getProcessorInfo Request

Parameters	Type	Mandatory	Description
processor	String	Yes	Managed processor UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getProcessorInfo",
        "params": {
                "processor": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getProcessorInfo",
        "params": {
                "processor": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.15.2 Response

Table 29. getProcessorInfo Response

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>status</td><td>Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component:
"Enabled",
"Disabled",
"StandbyOffline",
"StandbySpare,"<br>"InTest",
"Starting", "Absent,
"UnavailableOffline"</td></tr>
<tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource:
"OK", "Warning",
"Critical"</td></tr>
</table>
</td></tr>
<tr><td>socket</td><td>String</td><td>No</td><td>Socket identifier where the processor is installed, e.g., "CPU 1"</td></tr>
<tr><td>name</td><td>String</td><td>No</td><td>Optional name of the component</td></tr>
<tr><td>description</td><td>String</td><td>No</td><td>Optional description of the component</td></tr>
<tr><td>processorType</td><td>String</td><td>Yes</td><td>Processor type:
"CPU", "GPU", "FPGA", "DSP", "Accelerator", "OEM"</td></tr>
</table>

Result	Type	Mandatory	Description
processorArchitecture	String	No	Processor architecture: "x86", "IA-64", "ARM", "MIPS", "OEM"
instructionSet	String	No	Supported CPU instruction sets such as: "x86", "x86-64", "IA-64", "ARM-A32", "ARM-A64", "MIPS32", "MIPS64", "OEM"
capabilities	Array: String	No	An array of strings describing processor capabilities (like reported in /proc/cpuinfo flags), such as "sse" - Streaming SIMD Extensions "avx" - Advanced Vector Extensions etc.
manufacturer	String	No	Processor manufacturer e.g. "Intel Corporation"
model	String	No	Processor model (based on CPUID): Intel® Xeon® family: "E3", "E5", "E7" SoC family: "X3" (Avoton), "X5" (Broadwell-DE), "X7" Core family: "I3", "I5", "I7" "Silver", "Gold", "Platinum" Unknown model: "Unknown"
modelName	String	Yes	Processor model, e.g., "Multi-Core Intel® Xeon® processor 7xxx Series."
cpuId	Object	No	

Attribute	Type	Mandatory	Description
vendorId	String	No	Numeric vendor id: "GenuineIntel"
numericId	String	No	cpuid.1.eax[31:0]
family	String	No	CPU family
model	String	No	CPU model
step	String	No	CPU stepping
microcodeInfo	String	No	Microcode version

extendedCpuId	Object	No	

Attribute	Type	Mandatory	Description
eax00h	String	No	Register value
eax01h	String	No	Register value
eax02h	String	No	Register value
eax03h	String	No	Register value
eax04h	String	No	Register value
eax05h	String	No	Register value
eax06h	String	No	Register value
eax07h	String	No	Register value
eax09h	String	No	Register value
eax0ah	String	No	Register value
eax0bh	String	No	Register value
eax0dh	String	No	Register value
eax0fh	String	No	Register value
eax10h	String	No	Register value
eax14h	String	No	Register value
eax15h	String	No	Register value

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>eax16h</td><td>String</td><td>No</td><td>Register value</td></tr>
<tr><td>eax17hecx00h</td><td>String</td><td>No</td><td>Register value</td></tr>
<tr><td>eax17hecx01h</td><td>String</td><td>No</td><td>Register value</td></tr>
<tr><td>eax17hecx02h</td><td>String</td><td>No</td><td>Register value</td></tr>
<tr><td>eax17hecx03h</td><td>String</td><td>No</td><td>Register value</td></tr>
<tr><td>eax80000000h</td><td>String</td><td>No</td><td>Register value</td></tr>
<tr><td>eax80000001h</td><td>String</td><td>No</td><td>Register value</td></tr>
<tr><td>eax80000002h</td><td>String</td><td>No</td><td>Register value</td></tr>
<tr><td>eax80000003h</td><td>String</td><td>No</td><td>Register value</td></tr>
<tr><td>eax80000004h</td><td>String</td><td>No</td><td>Register value</td></tr>
<tr><td>eax80000006h</td><td>String</td><td>No</td><td>Register value</td></tr>
<tr><td>eax80000007h</td><td>String</td><td>No</td><td>Register value</td></tr>
<tr><td>eax80000008h</td><td>String</td><td>No</td><td>Register value</td></tr>
</table>
</td></tr>
<tr><td>maxSpeedMHz</td><td>Number</td><td>No</td><td>The maximum supported frequency of CPU</td></tr>
<tr><td>totalCores</td><td>Number</td><td>No</td><td>Number of available cores</td></tr>
<tr><td>enabledCores</td><td>Number</td><td>No</td><td>Number of enabled cores</td></tr>
<tr><td>totalThreads</td><td>Number</td><td>No</td><td>Number of available threads</td></tr>
<tr><td>enabledThreads</td><td>Number</td><td>No</td><td>Number of enabled threads</td></tr>
<tr><td>thermalDesignPowerWatt</td><td>Number</td><td>No</td><td>CPU thermal design power in watts</td></tr>
<tr><td>onPackageMemory</td><td>Array: Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>capacity</td><td>Number</td><td>No</td><td>The capacity of on-package memory</td></tr>
<tr><td>type</td><td>String</td><td>No</td><td>Memory type. Possible values: "EDRAM", "HBM", "HBM2"</td></tr>
<tr><td>speedMHz</td><td>Number</td><td>No</td><td>Memory speed in MHz</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>fpga</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>type</td><td>String</td><td>No</td><td>FPGA type. Supported values:
"Integrated",
"Discrete".</td></tr>
<tr><td>model</td><td>String</td><td>No</td><td>Model of the FPGA device. Supported values:
"Arria10",
"Stratix10",
"Stratix11", "OEM"</td></tr>
<tr><td>fwId</td><td>String</td><td>No</td><td>FPGA firmware identifier</td></tr>
<tr><td>fwManufacturer</td><td>String</td><td>No</td><td>FPGA firmware manufacturer</td></tr>
<tr><td>fwVersion</td><td>String</td><td>No</td><td>FPGA firmware version</td></tr>
<tr><td>hostInterface</td><td>String</td><td>No</td><td>Type of the FPGA interface to the host. Supported values:
"QPI", "UPI",
"2xPCIe-4",
"4xPCIe-4",
"8xPCIe-4",
"16xPCIe-4",
"2xPCIe-5",
"4xPCIe-5",
"8xPCIe-5",
"16xPCIe-5",
"1x10G", "2x10G",
"4x10G", "10x10G",
"1x25G",
"2x25G",
"4x25G",
"1x40G",
"2x40G",
"1x50G",
"2x50G",
"1X100G",
"SPI",
"SMBus",
"I2C",
"OEM"</td></tr>
<tr><td>externalInterfaces</td><td>Array: String</td><td>No</td><td>An array of the FPGA external interfaces</td></tr>
<tr><td>sidebandInterface</td><td>String</td><td>No</td><td>Type of the FPGA sideband interface</td></tr>
<tr><td>pcieVirtualFunctions</td><td>Number</td><td>No</td><td>Number of PCIe Virtual Functions configured within the FPGA</td></tr>
</table>
</td></tr>
</table>

Result	Type	Mandatory	Description
			programmableFromHost — Boolean — No — Indicates whether the FPGA FW can be reprogrammed from the host - if set to false, an FPGA FW can be programmed through the sideband interface only reconfigurationSlotsDetails — Array: Object — No — Refer to Table 30 bitStreamVersion — String — No — The version of the bitstream hssiConfiguration — String — No — Configuration string representing configuration of the HSSI, supported values: "4x10G", "2x40G", "10x10G", "PCIe" hssiSideband — String — No — Sideband configuration supported mechanism. Valid values: "SPI", "I2C-0", "I2C-1", "I2C" reconfigurationSlots — Number — No — Number of Partial Reconfiguration Slots available in this FPGA erased — Boolean — No — Indicates whether the FPGA was cleared after assignment to a Composed Node.
collections	Array: Object	Yes	Subcomponents collections: Attribute — Type — Mandatory — Description name — String — Yes — Collection name type — String — Yes — Collection type: "AccelerationFunctions", "Processor"
oem	Object	No	OEM specific data

Table 30. Reconfiguration Slot Details

Result	Type	Mandatory	Description
slotId	String	No	Slot ID of reconfiguration
uuid	String	No	UUID for the reconfiguration slot
programmableFromHost	Number	No	Indicates whether the configuration slot can be reprogrammed from the host
accelerationFunction	String	No	Acceleration function UUID reconfigured in this slot

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
```

```
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "socket": <string>,
                "processorType": <{"CPU", "GPU", "FPGA", "DSP", "Accelerator",
"OEM"}>,
                "processorArchitecture": <{"x86", "IA-64", "ARM", "MIPS", "OEM"}>,
                "instructionSet": <{"x86", "x86-64", "IA-64", "ARM-A32", "ARM-A64",
"MIPS32", "MIPS64", "OEM"}>,
                "capabilities": [
                        <{"sse", "sse2", "sse3", "sse4", "sse5", "avx","avx2",
"vt-d", "vt-x", "aes", "mmx", "em64t"}>
                ],
                "manufacturer": <string>,
                "model": <{"E3", "E5", "E7", "X3", "X5", "X7", "I3", "I5", "I7",
"Unknown", "Silver", "Gold", "Platinum"}>,
                "modelName": <string>,
                "cpuid": {
                        "vendorId": <string>,
                        "numericId" <string>,
                        "family": <string>,
                        "model": <string>,
                        "step": <string>,
                        "microcodeInfo": <string>
                },
                "maxSpeedMHz": <number>,
                "totalCores": <number>,
                "enabledCores": <number>,
                "totalThreads": <number>,
                "enabledThreads": <number>,
                "thermalDesignPowerWatt": <number>,
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"AccelerationFunctions", "Processor"}>
                        }],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "socket": "1",
                "processorType": "CPU",
                "processorArchitecture": "x86",
                "instructionSet": "x86-64",
                "capabilities": [
                        "sse",
                        "sse2"
                ],
                "manufacturer": "Intel(R) Corporation",
                "model": "E7",
```

```
                "modelName": "Multi-Core Intel(R) Xeon(R) processor 7xxx Series",
                "cpuId": {
                        "vendorId": "GenuineIntel",
                        "numericId" "329442",
                        "family": "6",
                        "model": "5",
                        "step": "2",
                        "microcodeInfo": "11"
                },
                "maxSpeedMHz": 3700,
                "totalCores": 8,
                "enabledCores": 4,
                "totalThreads": 4,
                "enabledThreads": 2,
                "collections": [
                        {
                                "name": "AccelerationFunctions",
                                "type": "AccelerationFunctions"
                        }],
                "oem": {}
        },
        "id": 987
}
```


5.16 Set Processor Attributes

The "setComponentAttributes" described in Section 5.5, setComponentAttributes allows configuration of the Set Processor Attributes in the following table.

Table 31. Set Drive Attributes

Attribute	Type	Description
erased	Boolean	Current indicator of whether an FPGA processor was erased.
securelyErase	Boolean	Triggers a secure erase of an FPGA processor.

5.17 getAccelerationFunctionInfo

The "getAccelerationFunctionInfo" command retrieves detailed information about a single acceleration function.

5.17.1 Request

Table 32. getAccelerationFunctionInfo Request

Parameters	Type	Mandatory	Description
accelerationFunction	String	Yes	Acceleration function UUID

Serialization:


```
{
        "jsonrpc": "2.0",
        "method": "getAccelerationFunctionInfo",
        "params": {
                "accelerationFunction": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getAccelerationFunctionInfo",
        "params": {
                "accelerationFunction": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.17.2 Response

Table 33. getAccelerationFunctionInfo Response

Result	Type	Mandatory	Description
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent”, "UnavailableOffline"</td></tr><tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
uuid	String	Yes	UUID of the acceleration function
name	String	No	Optional name of the component
description	String	No	Optional description of the component
reconfigurationSlots	Array: String	No	The array of the FPGA reconfiguration slots identifiers which implement this function
accelerationFunctionType	String	No	Identifies the type of acceleartion function. Supported values: "Encryption", "Compression", "PacketInspection", "PacketSwitch", "Scheduler", "VideoProcessing", "OEM"
manufacturer	String	No	Acceleration function code manufacturer
version	String	No	Acceleration function code version
powerWatts	Number	No	Acceleration function power consumption in watts
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "uuid": <string>,
```

```
                "reconfigurationSlots": [
                        <string>,
                        …
                ],
                "accelerationFunctionType": <{"Encryption", "Compression",
"PacketInspection", "PacketSwitch", "Scheduler", "VideoProcessing", "OEM"}>,
                "manufacturer": <string>,
                "version": <string>,
                "powerWatts": <number>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "uuid": "00000000-0000-0000-0000-000000000000",
                "reconfigurationSlots": [
                        "AFU0"
                ],
                "accelerationFunctionType": "Compression",
                "manufacturer": "Intel (R) Corporation",
                "version": "Green Compression Type 1 v.1.00.86",
                "powerWatts": 15,
                "oem": {}
        },
        "id": 987
}
```

5.18 getMemoryInfo

The `"getMemoryInfo"` command retrieves detailed information about a single memory module.

5.18.1 Request

Table 34. getMemoryInfo Request

Parameters	Type	Mandatory	Description
`memory`	String	Yes	Managed memory UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getMemoryInfo",
        "params": {
                "memory": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getMemoryInfo",
        "params": {
                  "memory": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.18.2 Response

Table 35. getMemoryInfo Response

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>status</td><td>Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component:
"Enabled",
"Disabled",
"StandbyOffline",
"StandbySpare,"<br>"InTest",
"Starting",
"Absent",
"UnavailableOffline"</td></tr>
<tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource:
"OK", "Warning", "Critical"</td></tr>
</table>
</td></tr>
<tr><td>name</td><td>String</td><td>No</td><td>Optional name of the component</td></tr>
<tr><td>description</td><td>String</td><td>No</td><td>Optional description of the component</td></tr>
<tr><td>memoryType</td><td>String</td><td>No</td><td>Memory type:
"DRAM","NVDIMM_N", "NVDIMM_F", "NVDIMM_P", "IntelOptane"</td></tr>
<tr><td>deviceType</td><td>String</td><td>No</td><td>DIMM type:
"DDR", "DDR2", "DDR3", "DDR4", "DDR4_SDRAM", "DDR4E_SDRAM", "LPDDR4_SDRAM", "DDR3_SDRAM", "LPDDR3_SDRAM", "DDR2_SDRAM", "DDR2_SDRAM_FB_DIMM", "DDR2_SDRAM_FB_DIMM_PROBE", "DDR_SGRAM", "DDR_SDRAM", "ROM", "SDRAM", "EDO", "FastPageMode", "PipelinedNibble"</td></tr>
<tr><td>moduleType</td><td>String</td><td>No</td><td>The base module type of DIMM:
"RDIMM", "UDIMM", "SO_DIMM", "LRDIMM", "Mini_RDIMM", "Mini_UDIMM", "SO_RDIMM_72b", "SO_UDIMM_72b", "SO_DIMM_16b", "SO_DIMM_32b"</td></tr>
<tr><td>media</td><td>Array:
String</td><td>No</td><td>Memory module media types:
"DRAM", "NAND", "Proprietary"</td></tr>
<tr><td>capacityMiB</td><td>Number</td><td>Yes</td><td>Memory module size in MiB</td></tr>
<tr><td>logicalSizeMiB</td><td>Number</td><td>No</td><td>Memory logical size in MiB</td></tr>
<tr><td>dataWidthBits</td><td>Number</td><td>No</td><td>Data Width in bits</td></tr>
<tr><td>busWidthBits</td><td>Number</td><td>No</td><td>Bus Width in bits</td></tr>
</table>

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>fruInfo</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>serialNumber</td><td>String</td><td>No</td><td>Serial number</td></tr>
<tr><td>manufacturer</td><td>String</td><td>No</td><td>Manufacturer name</td></tr>
<tr><td>modelNumber</td><td>String</td><td>No</td><td>Model number</td></tr>
<tr><td>partNumber</td><td>String</td><td>No</td><td>Part number</td></tr>
</table>
</td></tr>
<tr><td>firmwareRevision</td><td>String</td><td>No</td><td>Memory module firmware revision</td></tr>
<tr><td>firmwareApiVersion</td><td>String</td><td>No</td><td>Memory module firmware API version</td></tr>
<tr><td>maxTDPMilliWatts</td><td>Array: Number</td><td>No</td><td>The maximum power budgets supported by the Memory in milliwatts.</td></tr>
<tr><td>moduleManufacturerID</td><td>String</td><td>No</td><td>Model manufacturer ID</td></tr>
<tr><td>moduleProductID</td><td>String</td><td>No</td><td>Model product ID</td></tr>
<tr><td>memorySubsystemControllerManufacturerID</td><td>String</td><td>No</td><td>The manufacturer ID of the memory subsystem controller for the memory module</td></tr>
<tr><td>memorySubsystemControllerProductID</td><td>String</td><td>No</td><td>The product ID of the memory subsystem controller for the memory module</td></tr>
<tr><td>subsystemVendorID</td><td>String</td><td>No</td><td>Memory subsystem controller manufacturer ID</td></tr>
<tr><td>subsystemDeviceID</td><td>String</td><td>No</td><td>Memory subsystem controller product ID</td></tr>
<tr><td>allowedSpeedsMHz</td><td>Array: Number</td><td>No</td><td>Speed bins supported by this memory module</td></tr>
<tr><td>voltageVolt</td><td>Number</td><td>No</td><td>Typical memory module voltage</td></tr>
<tr><td>minimumVoltageVolt</td><td>Number</td><td>No</td><td>Minimum memory module voltage</td></tr>
<tr><td>maximumVoltageVolt</td><td>Number</td><td>No</td><td>Maximum memory module voltage</td></tr>
<tr><td>deviceLocator</td><td>String</td><td>No</td><td>Location of the DIMM in the platform, typically marked in the silkscreen</td></tr>
<tr><td>location</td><td>Object</td><td>No</td><td>Property describing DIMM location concerning processor and memory controller
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>socket</td><td>Number</td><td>No</td><td>Socket number</td></tr>
<tr><td>controller</td><td>Number</td><td>No</td><td>Memory controller number</td></tr>
<tr><td>channel</td><td>Number</td><td>No</td><td>Channel number</td></tr>
<tr><td>slot</td><td>Number</td><td>No</td><td>Slot number</td></tr>
</table>
</td></tr>
<tr><td>rankCount</td><td>Number</td><td>No</td><td>Number of ranks available in the memory module</td></tr>
<tr><td>errorCorrection</td><td>String</td><td>No</td><td>Error correction scheme supported for this memory: "NoECC", "SingleBitECC", "MultiBitECC", "AddressParity"</td></tr>
<tr><td>volatileRegionSizeLimitMiB</td><td>Number</td><td>No</td><td>The total size of volatile regions in MiB</td></tr>
<tr><td>persistentRegionSizeLimitMiB</td><td>Number</td><td>No</td><td>The total size of persistent regions in MiB</td></tr>
<tr><td>volatileSizeMiB</td><td>Number</td><td>No</td><td>Volatile memory size in MiB</td></tr>
<tr><td>nonVolatileSizeMiB</td><td>Number</td><td>No</td><td>Non-volatile memory size in MiB</td></tr>
</table>

Result	Type	Mandatory	Description
operatingSpeedMHz	Number	No	Operating speed of the memory in MHz
operatingMemoryModes	Array: String	No	Memory modes supported by the DIMM. Available values: "Volatile" – Volatile memory, "PMEM" – Persistent memory, byte accessible through system address space, "Block" – Block accessible system memory
regions	Array: Object	No	Memory regions configured within the memory module

Attribute	Type	Mandatory	Description
regionId	String	No	Unique region ID representing a specific region within the memory module
memoryType	String	No	Region memory classification: "Volatile", "Block", "Persistent"
offsetMiB	Number	No	Offset within the memory module corresponding to the region begin
sizeMiB	Number	No	Region size in MiB

volatileRegionSizeLimitMiB	Number	No	The total size of volatile regions in MiB
persistentRegionSizeLimitMiB	Number	No	The total size of persistent regions in MiB
volatileSizeMiB	Number	No	Volatile memory size in MiB
nonVolatileSizeMiB	Number	No	Non-volatile memory size in MiB
operatingSpeedMHz	Number	No	Operating speed of the memory in MHz

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>securityCapabilities</td><td>Object</td><td>No</td><td>Properties which describe the security capabilities of a memory module
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>passphraseCapable</td><td>Boolean</td><td>No</td><td>Memory passphrase set capability</td></tr>
<tr><td>maxPassphraseCount</td><td>Number</td><td>No</td><td>Maximum number of passphrases supported for Memory</td></tr>
<tr><td>securityStates</td><td>Array: String</td><td>No</td><td>Security states supported b the Memory:
"Enabled", "Disabled"
"Unlocked",
"Locked",
"Frozen",
"Passphraselimit"</td></tr>
</table></td></tr>
<tr><td>spareDeviceCount</td><td>Number</td><td>No</td><td>The number of unused spare devices available in the Memory</td></tr>
<tr><td>powerManagementPolicy</td><td>Object</td><td>No</td><td>Properties which describe the power management policy for a memory module
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>policyEnabled</td><td>Boolean</td><td>No</td><td>Power management policy enabled status</td></tr>
<tr><td>maxTDPMilliWatts</td><td>Number</td><td>No</td><td>Maximum TDP in milliwatts</td></tr>
<tr><td>peakPowerBudgetMilliWatts</td><td>Number</td><td>No</td><td>Peak power budget in milliwatts</td></tr>
<tr><td>averagePowerBudgetMilliWatts</td><td>Number</td><td>No</td><td>Average power budget in milliwatts</td></tr>
</table></td></tr>
<tr><td>oem</td><td>Object</td><td>No</td><td>OEM specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                  "status": {
                         "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline"}>,
                         "health": <{"OK", "Warning", "Critical"}>
                  },
                  "memoryType": <{"DRAM","NVDIMM_N", "NVDIMM_F", "NVDIMM_P",
"IntelOptane"}>,
                  "deviceType": <{"DDR", "DDR2", "DDR3", "DDR4", "DDR4_SDRAM",
"DDR4E_SDRAM", "LPDDR4_SDRAM", "DDR3_SDRAM", "LPDDR3_SDRAM", "DDR2_SDRAM",
"DDR2_SDRAM_FB_DIMM", "DDR2_SDRAM_FB_DIMM_PROBE", "DDR_SGRAM", "DDR_SDRAM", "ROM",
"SDRAM", "EDO", "FastPageMode", "PipelinedNibble"}>,
```

```
                "moduleType": <{"RDIMM", "UDIMM", "SO_DIMM", "LRDIMM", "Mini_RDIMM",
"Mini_UDIMM", "SO_RDIMM_72b", "SO_UDIMM_72b", "SO_DIMM_16b", "SO_DIMM_32b"}>,
                "media": [
                        <{"DRAM", "NAND", "Proprietary"}>,
                        ...
                ],
                        ...
                ],
                "capacityMiB": <number>,
                "dataWidthBits": <number>,
                "busWidthBits": <number>,
                "fruInfo": {
                        "serialNumber": <string>,
                        "manufacturer": <string>,
                        "modelNumber": <string>,
                        "partNumber": <string>
                },
                "firmwareRevision": <string>,
                "firmwareApiVersion": <string>,
                "maxTDPMilliWatts":
                [
                        <number>,
                        ...
                ],
                "functionClasses": [
                        <{"Volatile", "Block", "Persistent"}>,
                        ...
                ],
                "moduleManufacturerID": <string>,
                "moduleProductID": <string>,
                "operatingSpeedMHz": <number>,
                "allowedSpeedsMHz": [
                        <number>,
                        ...
                ],
                "voltageVolt": <number>,
                "minimumVoltageVolt": <number>,
                "maximumVoltageVolt": <number>,
                "deviceLocator": <string>
                "location: {
                        "socket": <number>,
                        "controller": <number>,
                        "channel": <number>,
                        "slot": <number>
                },
                "rankCount": <number>,
                "errorCorrection": <{"NoECC", "SingleBitECC", "MultiBitECC",
                                "AddressParity"}>,
                "regions": [{
                        "regionId": <string>,
                        "memoryType": <{"Volatile", "Block", "Persistent"}>,
                        "offsetMiB": <number>,
                        "sizeMiB": <number>
                        }
                ...
                ],
                "operatingMemoryModes": [
                        <String>,
                        …
                ],
                "oem": <object>
```

```
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "memoryType": "DRAM",
                "deviceType": "DDR4",
                "moduleType": "RDIMM",
                "media": [
                        "DRAM"
                ],
                "capacityMiB": 16384,
                "dataWidthBits": 64,
                "busWidthBits": 72,
                "fruInfo": {
                        "serialNumber": "123fed3029c-b23394-12",
                        "manufacturer": "Intel Corporation",
                        "modelNumber": "E323",
                        "partNumber": "29ee2220939"
                },
                "firmwareRevision": "RevAbc",
                "firmwareApiVersion": "ApiAbc",
                "maxTDPMilliWatts": [ 2.5 ],
                "functionClasses": [
                        "Volatile"
                ],
                "moduleManufacturerID": "0x8086",
                "moduleProductID": "0xAB245f",
                "operatingSpeedMHz": 2400,
                "allowedSpeedsMHz": [
                        2133,
                        2400,
                        2667
                ],
                "voltageVolt": 1.35,
                "minimumVoltageVolt": 1.3,
                "maximumVoltageVolt": 1.4,
                "deviceLocator": "PROC 1 DIMM 1"
                "location: {
                        "socket": 1,
                        "controller": 1,
                        "channel": 1,
                        "slot": 1
                },
                "rankCount": 1,
                "errorCorrection": "MultiBitECC",
                "regions": [{
                        "regionId": "1",
                        "memoryType": "Volatile",
                        "offsetMiB": 0,
                        "sizeMB": 16384
                        }
                ],
```

```
                "operatingMemoryModes": [
                        "Volatile"
                ]
                "oem": {}
        },
        "id": 987
}
```

5.19 Set Memory Attributes

The "setComponentAttributes" described in the Section 5.5, setComponentAttributes, allows configuration of the Memory Attributes listed in the following table.

Table 36. Configurable Memory Attributes

Attribute	Type	Description
regionId	String	Memory region ID for which action is to be applied
passphrase	String	The passphrase of a given Memory region, the action will be applied if the passphrase is correct
oem	Object	OEM specific data

5.20 getMemoryDomainInfo

The "getMemoryDomainInfo" command retrieves detailed information about a single memory domain.

5.20.1 Request

Table 37. getMemoryDomainInfo Request

Parameters	Type	Mandatory	Description
memoryDomain	String	Yes	Managed memory domain UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getMemoryDomainInfo",
        "params": {
                "memoryDomain": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getMemoryDomainInfo",
        "params": {
                "memoryDomain": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.20.2 Response

Table 38. getMemoryDomainInfo Response

Result	Type	Mandatory	Description
name	String	No	Optional name of the component
description	String	No	Optional description of the component
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare", "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr><tr><td>health</td><td>String</td><td>No</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
interleavableMemorySets	Array: Array	Yes	The array of interleaving memory UUID sets (arrays) for a memory chunk
allowsMemoryChunkCreation	Boolean	No	Indicates if this Memory Domain supports the creation of Memory Chunks
allowsBlockProvisioning	Boolean	No	Indicates if this Memory Domain supports the creation of Blocks of memory
allowsMirroring	Boolean	No	Indicates if this Memory Domain supports the creation of Memory Chunks with mirroring enabled
allowsSparing	Boolean	No	Indicate if this Memory Domain supports the creation of Memory Chunks with sparing enabled
collections	Array: Object	Yes	Subcomponents collections <table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>name</td><td>String</td><td>Yes</td><td>Collection name</td></tr><tr><td>type</td><td>String</td><td>Yes</td><td>Collection type: "MemoryChunks"</td></tr></table>
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "interleavableMemorySets": [
                        [<uuid>, <uuid>, …],
                        …
                ],
                "allowsMemoryChunkCreation": <boolean>,
                "allowsBlockProvisioning": <boolean>,
```

```
                "allowsMirroring": <boolean>,
                "allowsSparing": <boolean>,
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"MemoryChunks"}>
                        }],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "interleavableMemorySets": [
                        ["123e4567-e89b-12d3-a456-426655440000", "123e4567-e89b-12d3-
a456-426655440001"],
                        ["123e4567-e89b-12d3-a456-426655440003", "123e4567-e89b-12d3-
a456-426655440004"]
                ],
                "allowsMemoryChunkCreation": true,
                "allowsBlockProvisioning": true,
                "allowsMirroring": false,
                "allowsSparing": false,
                "collections": [
                        {
                                "name": "MemoryChunks",
                                "type": "MemoryChunks"
                        }],
                "oem": {}
        },
        "id": 987
}
```

5.21 getMemoryChunksInfo

The "getMemoryChunskInfo" command retrieves detailed information about a single memory chunks.

5.21.1 Request

Table 39. getMemoryChunksInfo Request

Parameters	Type	Mandatory	Description
memoryChunks	String	Yes	Memory chunks UUID.

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getMemoryChunksInfo",
        "params": {
                "memoryChunks": <string>
        },
        "id": <id>
}
```

Example:

```
{
    "jsonrpc": "2.0",
    "method": "getMemoryChunksInfo",
    "params": {
        "memoryChunks": "123e4567-e89b-12d3-a456-426655440000"
    },
    "id": 987
}
```

5.21.2 Response

Table 40. getMemoryChunksInfo Response

Result	Type	Mandatory	Description
name	String	No	Optional name of the component
description	String	No	Optional description of the component
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare", "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr><tr><td>health</td><td>String</td><td>No</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
addressRangeType	String		A memory type of this memory chunk. Available values: "Volatile" – Volatile memory, "PMEM" – Persistent memory, byte accessible through system address space, **"Block" – Block accessible system memory**
interleavableSets	Array: Object	Yes	Properties which describe the interleavable Memory Chunk <table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>memory</td><td>String</td><td>Yes</td><td>Memory UUID</td></tr><tr><td>regionId</td><td>String</td><td>Yes</td><td>Memory region ID</td></tr><tr><td>offsetMiB</td><td>Number</td><td>Yes</td><td>Offset within the DIMM that corresponds to the starting of this memory region in MiB</td></tr><tr><td>sizeMiB</td><td>Number</td><td>Yes</td><td>Size of this memory region in MiB</td></tr><tr><td>memoryLevel</td><td>Number</td><td>Yes</td><td>The level of this interleave set for multi-level tiered memory</td></tr></table>
isMirrorEnabled	Boolean	No	Indicates if this Memory Chunk has mirroring enabled

Result	Type	Mandatory	Description
isSpare	Boolean	No	Indicates if this Memory Chunk has sparing enabled
memoryChunkSizeMiB	Number	No	Size of the memory chunk
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "interleavableSets": [{
                        "memory": <string>,
                        "regionId": <string>,
                        "offsetMiB": <number>,
                        "sizeMiB": <number>,
                        "memoryLevel": <number>
                },
                …
                ],
                "addressRangeType": <number>,
                "isMirrorEnabled": <boolean>,
                "isSpare": <boolean>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "interleavableSets": [{
                        "memory": "123e4567-e89b-12d3-a456-426655440000",
                        "regionId": "2",
                        "offsetMiB": 0,
                        "sizeMiB": 16384,
                        "memoryLevel": 0
                }],
                "addressRangeType": "PMEM",
                "isMirrorEnabled": false,
                "isSpare": false,
                "oem": {}
        },
        "id": 987
}
```

5.22 getStorageSubsystemInfo

The "getStorageSubsystemInfo" command retrieves detailed information about a single storage subsystem.

5.22.1 Request

Table 41. getStorageSubsystemInfo Request

Parameters	Type	Mandatory	Description
storage	String	Yes	Managed storage subsystem UUID.

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getStorageSubsystemInfo",
        "params": {
                "storage": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getStorageSubsystemInfo",
        "params": {
                "storage": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.22.2 Response

Table 42. getStorageSubsystemInfo Response

Result	Type	Mandatory	Description
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr><tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
name	String	No	Optional name of the component
description	String	No	Optional description of the component
collections	Array: Object	Yes	Subcomponents collections <table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>name</td><td>String</td><td>Yes</td><td>Collection name</td></tr><tr><td>type</td><td>String</td><td>Yes</td><td>Collection type: "Drives", "StorageControllers"</td></tr></table>
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"Drives", "StorageControllers"}>
                        }
                ...
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "collections": [
                        {
                                "name": "Storage",
                                "type": "StorageControllers",
                        },
                        {
                                "name": "Drives",
                                "type": "Drives",
                        },

                ],
                "oem": {}
        },
        "id": 987
}
```

5.23 Set Storage Subsystem Attributes

The `"setComponentAttributes"` described in Section 5.5, setComponentAttributes, allows configuration of the Storage Controller Attributes listed in the following table.

Table 43. Configurable Storage Subsystem Attributes

Attribute	Type	Description
`encryptionKey`	String	The encryption key to set on the storage subsystem.
`assetTag`	String	Tag assigned to the asset.
`oem`	Object	OEM specific data

5.24 getStorageControllerInfo

The "getStorageControllerInfo" command retrieves detailed information about a single storage controller.

5.24.1 Request

Table 44. getStorageControllerInfo Request

Parameters	Type	Mandatory	Description
controller	String	Yes	Managed controller UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getStorageControllerInfo",
        "params": {
                "controller": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getStorageControllerInfo",
        "params": {
                "controller": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.24.2 Response

Table 45. getStorageControllerInfo Response

Result	Type	Mandatory	Description
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr><tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
name	String	No	Optional name of the component
description	String	No	Optional description of the component

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>fruInfo</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>serialNumber</td><td>String</td><td>No</td><td>Module serial number</td></tr>
<tr><td>manufacturer</td><td>String</td><td>No</td><td>Manufacturer name</td></tr>
<tr><td>modelNumber</td><td>String</td><td>No</td><td>Model number</td></tr>
<tr><td>partNumber</td><td>String</td><td>No</td><td>Part number</td></tr>
</table>
</td></tr>
<tr><td>physicalId</td><td>String</td><td>No</td><td>Storage controller physical Id (bus type and location)</td></tr>
<tr><td>speedGbps</td><td>Number</td><td>No</td><td>This resource shall be used to represent a resource that represents a storage controller in the Redfish specification.</td></tr>
<tr><td>firmwareVersion</td><td>String</td><td>No</td><td>The firmware version of this storage Controller.</td></tr>
<tr><td>sku</td><td>String</td><td>No</td><td>This is the model number for the storage controller.</td></tr>
<tr><td>assetTag</td><td>String</td><td>No</td><td>The user assigned asset tag for this storage controller.</td></tr>
<tr><td>identifiers</td><td>Array: Object</td><td>No</td><td>This property shall contain a list of all known durable names for the associated drive:
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>durableName</td><td>String</td><td>Yes</td><td>This indicates the worldwide, the persistent name of the resource.</td></tr>
<tr><td>durableNameFormat</td><td>String</td><td>Yes</td><td>Represents the format of the durableName property: "NAA", "iQN", "FC_WWN", "UUID", "EUI", "NQN", "NSID".</td></tr>
</table>
</td></tr>
<tr><td>supportedControllerProtocols</td><td>Array: String</td><td>No</td><td>This represents the protocols by which this storage controller can be communicated to. Array contains of elements: "PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB", "NVMe", "FC", "iSCSI", "FCoE", "NVMeOverFabrics", "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS", "FTP", "FTPS"</td></tr>
<tr><td>supportedDeviceProtocols</td><td>Array: String</td><td>No</td><td>This represents the protocols which the storage controller can use to communicate with attached devices. Array contains of elements: "PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB", "NVMe", "FC", "iSCSI", "FCoE", "NVMeOverFabrics", "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS", "FTP", "FTPS"</td></tr>
<tr><td>oem</td><td>Object</td><td>No</td><td>OEM specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "fruInfo": {
                        "serialNumber": <string>,
                        "manufacturer": <string>,
                        "modelNumber": <string>,
                        "partNumber": <string>
```

```
                },
                "physicalId" : <string>,
                "speedGbps": <number>,
                "firmwareVersion": <string>,
                "sku": <string>,
                "assetTag": <string>,
                "identifiers": [
                        {
                                "durableName": <string>,
                                "durableNameFormat": <{"NAA", "iQN", "FC_WWN",
"UUID", "EUI", "NQN", "NSID"}>
                        },
                        …
                ],
                "supportedControllerProtocols":[
                        <{"PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB", "NVMe", "FC",
"iSCSI", "FCoE", "NVMeOverFabrics", "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS", "FTP",
"FTPS".}>
                ],
                "supportedDeviceProtocols":[
                        <{"PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB", "NVMe", "FC",
"iSCSI", "FCoE", "NVMeOverFabrics", "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS", "FTP",
"FTPS"}>
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "fruInfo": {
                        "serialNumber": "123fed3029c-b23394-12",
                        "manufacturer": "Intel Corporation",
                        "modelNumber": "E323",
                        "partNumber": "29ee2220939"
                },
                "physicalId" : "pci@0000:01:00.0",
                "speedGbps": 2.0,
                "firmwareVersion": "ss123fr",
                "sku": "sk123xp123",
                "assetTag": "best asset",
                "identifiers": [
                        {
                                "durableName": "123e4567-e89b-12d3-a456-
426655440000",
                                "durableNameFormat": "UUID",
                        }
                ],
                "supportedControllerProtocols":[
                         "AHCI", "UHCI
                ],
                "supportedDeviceProtocols":[
                         "SATA", "SAS"
```

```
                ],
                "oem": {}
        },
        "id": 987
}
```

5.25 getDriveInfo

The "getDriveInfo" command retrieves detailed information about a single drive.

5.25.1 Request

Table 46. getDriveInfo Request

Parameters	Type	Mandatory	Description
drive	String	Yes	Managed drive UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getDriveInfo",
        "params": {
                "drive": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getDriveInfo",
        "params": {
                "drive": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.25.2 Response

Table 47. getDriveInfo Response

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>status</td><td>Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component:
"Enabled",
"Disabled",
"StandbyOffline",
"StandbySpare,"<br>"InTest",
"Starting",
"Absent",
"UnavailableOffline"</td></tr>
<tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource:
"OK", "Warning",
"Critical"</td></tr>
</table>
</td></tr>
<tr><td>name</td><td>String</td><td>No</td><td>Optional name of the component</td></tr>
<tr><td>description</td><td>String</td><td>No</td><td>Optional description of the component</td></tr>
<tr><td>interface</td><td>String</td><td>No</td><td>Drive interface:
"PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB", "NVMe", "FC", "iSCSI", "FCoE", "NVMeOverFabrics", "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS", "FTP", "SFTP"</td></tr>
<tr><td>mediaType</td><td>String</td><td>No</td><td>Drive type:
"HDD", "SSD", "SMR"</td></tr>
<tr><td>rpm</td><td>Number</td><td>No</td><td>Rotation per minute (HDD only)</td></tr>
<tr><td>capacityGB</td><td>Number</td><td>Yes</td><td>Drive capacity in GB</td></tr>
<tr><td>fruInfo</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>serialNumber</td><td>String</td><td>No</td><td>Module serial number</td></tr>
<tr><td>manufacturer</td><td>String</td><td>No</td><td>Manufacturer name</td></tr>
<tr><td>modelNumber</td><td>String</td><td>No</td><td>Model number</td></tr>
<tr><td>partNumber</td><td>String</td><td>No</td><td>Part number</td></tr>
</table>
</td></tr>
<tr><td>firmwareVersion</td><td>String</td><td>No</td><td>Drive firmware version</td></tr>
<tr><td>physicalId</td><td>String</td><td>No</td><td>Drive physical location (JBOD)</td></tr>
<tr><td>indicatorLED</td><td>String</td><td>No</td><td>The state of the indicator LED used to identify the drive:
"Lit", "Blinking", "Off"</td></tr>
<tr><td>assetTag</td><td>String</td><td>No</td><td>The value of this property shall be an identifying string used to track the drive for inventory purposes.</td></tr>
<tr><td>capableSpeedGbps</td><td>Number</td><td>No</td><td>The speed which this drive can communicate to a storage controller in ideal conditions in Gigabits per second.</td></tr>
<tr><td>negotiatedSpeedGbps</td><td>Number</td><td>No</td><td>The speed which this drive is currently communicating to the storage controller in Gigabits per second.</td></tr>
</table>

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>location</td><td>Array: Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>info</td><td>String</td><td>Yes</td><td>This indicates the location of the resource.</td></tr>
<tr><td>infoFormat</td><td>String</td><td>Yes</td><td>This represents the format of the Info property.</td></tr>
</table>
</td></tr>
<tr><td>statusIndicator</td><td>String</td><td>No</td><td>The state of the status indicator, used to communicate status information about this drive:
"Ok", "Fail", "Rebuild", "PredictiveFailureAnalysis", "Hotspare", "InACriticalArray", "InAFailedArray"</td></tr>
<tr><td>revision</td><td>String</td><td>No</td><td>The revision of this Drive.</td></tr>
<tr><td>failurePredicted</td><td>Boolean</td><td>No</td><td>Is this drive currently predicting a failure shortly.</td></tr>
<tr><td>sku</td><td>String</td><td>No</td><td>This is the SKU for this drive.</td></tr>
<tr><td>identifiers</td><td>Array: Object</td><td>No</td><td>This property shall contain a list of all known durable names for the associated drive:
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>durableName</td><td>String</td><td>Yes</td><td>This indicates the worldwide, the persistent name of the resource.</td></tr>
<tr><td>durableNameFormat</td><td>String</td><td>Yes</td><td>Represents the format of the durableName property:
"NAA", "iQN", "FC_WWN", "UUID", "EUI", "NQN", "NSID".</td></tr>
</table>
</td></tr>
<tr><td>hotspareType</td><td>String</td><td>No</td><td>Type of hot spare this drive is currently serving as:
"None", "Global", "Chassis", "Dedicated"</td></tr>
<tr><td>encryptionAbility</td><td>String</td><td>No</td><td>The encryption abilities of this drive:
"None", "SelfEncryptingDrive", "Other"</td></tr>
<tr><td>encryptionStatus</td><td>String</td><td>No</td><td>The status of the encryption of this drive:
"Unencrypted","Unlocked","Locked","Foreign"</td></tr>
<tr><td>blockSizeBytes</td><td>Number</td><td>No</td><td>The size of the smallest unit (Block) of this drive in bytes.</td></tr>
<tr><td>predictedMediaLifeLeftPercent</td><td>Number</td><td>No</td><td>The percentage of reads and writes that are predicted to still be available for the media.</td></tr>
<tr><td>erased</td><td>Boolean</td><td>No</td><td>Indicates whether the drive was cleared after assignment to a Composed Node.</td></tr>
<tr><td>collections</td><td>Array: Object</td><td>No</td><td>Subcomponents collections
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>name</td><td>String</td><td>Yes</td><td>Collection name</td></tr>
<tr><td>type</td><td>String</td><td>Yes</td><td>Collection type:
"PCIeFunctions"</td></tr>
</table>
</td></tr>
<tr><td>oem</td><td>Object</td><td>No</td><td>OEM specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "interface": <{"PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB", "NVMe",
"FC", "iSCSI", "FCoE", "NVMeOverFabrics", "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS",
"FTP", "SFTP"}>
                "mediaType": <{"HDD", "SSD", "SMR"}>,
                "rpm": <number>,
                "capacityGB": <number>,
                "fruInfo": {
                        "serialNumber": <string>,
                        "manufacturer": <string>,
                        "modelNumber": <string>,
                        "partNumber": <string>
                },
                "firmwareVersion": <string>,
                "physicalId" : <string>,
                "indicatorLED": <{"Lit", "Blinking", "Off"}>,
                "assetTag": <string>,
                "capableSpeedGbps": <number>,
                "negotiatedSpeedGbps": <number>,
                "location": [
                        {
                                "info": <string>,
                                "infoFormat: <string>
                        },
                        …
                ],
                "statusIndicator": <{"Ok", "Fail", "Rebuild",
"PredictiveFailureAnalysis", "Hotspare", "InACriticalArray", "InAFailedArray"}>,
                "revision": <string>,
                "failurePredicted": <boolean>,
                "sku": <string>,
                "identifiers": [
                        {
                                "durableName": <string>,
                                "durableNameFormat": <{"NAA", "iQN", "FC_WWN",
"UUID", "EUI", "NQN", "NSID"}>
                        },
                        …
                ],
                "hotspareType": <{"None", "Global", "Chassis", "Dedicated"}>,
                "encryptionAbility": <{"None", "SelfEncryptingDrive", "Other }>,
                "encryptionStatus": <{"Unencrypted","Unlocked","Locked","Foreign"}>,
                "blockSizeBytes": <number>,
                "predictedMediaLifeLeftPercent": <number>,
                "collections": [
                        {
                                "name": "PCIeFunctions",
                                "type": "PCIeFunctions",
                        }
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:


```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "interface": "SATA"
                "meditaType": "HDD",
                "rpm": 7200,
                "capacityGB": 500,
                "fruInfo": {
                        "serialNumber": "123fed3029c-b23394-12",
                        "manufacturer": "Intel Corporation",
                        "modelNumber": "E323",
                        "partNumber": "29ee2220939"
                },
                "firmwareVersion": "0002",
                "physicalId" : "0.1.0",
                "indicatorLED": "Lit",
                "assetTag": "A12345",
                "capableSpeedGbps": 10.0,
                "negotiatedSpeedGbps": 8.0,
                "location": [
                        {
                        "info": "4",
                        "infoFormat: "HDD index"
                        }
                ],
                "statusIndicator": "Ok",
                "revision": "rev123po",
                "failurePredicted": false,
                "sku": "sku123po",
                "identifiers": [
                        {
                                "durableName": "123e4567-e89b-12d3-a456-
426655440000",
                                "durableNameFormat": "UUID",
                        }
                ],
                "hotspareType": "None",
                "encryptionAbility": "None",
                "encryptionStatus": "Unencrypted",
                "blockSizeBytes": 1,
                "predictedMediaLifeLeftPercent": 60.0,
                "collections": [
                        {
                                "name": "PCIeFunctions",
                                "type": "PCIeFunctions",
                        }
                ],
                "oem": {}
        },
        "id": 987
}
```

5.26 deleteDrive

The "deleteDrive" command deletes existing drive.

5.26.1 Request

Table 48. deleteDrive Request

Result	Type	Mandatory	Description
drive	String	Yes	Drive UUID.

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "deleteDrive",
        "params": {
                "drive": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "deleteDrive",
        "params": {
                "drive": "7ec52201-bd23-4cc2-1256-6262e544fda8"
        },
        "id": 1218
}
```

5.26.2 Response

Table 49. deleteDrive Response

Result	Type	Mandatory	Description
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 1218
}
```

5.27 Set Drive Attributes

The "setComponentAttributes" described in Section 5.5, setComponentAttributes allows configuration of the Set Drive Attributes in the following table.

Table 50. Set Drive Attributes

Attribute	Type	Description
assetTag	String	Tag assigned to the asset.
erased	Boolean	Current indicator of whether the drive was erased.
securelyErase	Boolean	Triggers a secure drive erase.
oem	Object	OEM specific data

5.28 getNetworkInterfaceInfo

The "getNetworkInterfaceInfo" commands retrieves detailed information about a network interface.

5.28.1 Request

Table 51. getNetworkInterfaceInfo Request

Parameters	Type	Mandatory	Description
interface	String	Yes	Managed network interface UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getNetworkInterfaceInfo",
        "params": {
                "interface": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getNetworkInterfaceInfo",
        "params": {
                "interface": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.28.2 Response

Table 52. getNetworkInterfaceInfo Response

Result	Type	Mandatory	Description
status	Object	Yes	

Attribute	Type	Mandatory	Description
state	String	Yes	Known state of the component: "Enabled", "Disabled",

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td></td><td></td><td></td><td>"StandbyOffline",
"StandbySpare,"<br>"InTest",
"Starting",
"Absent",
"UnavailableOffline"</td></tr>
<tr><td>health</td><td>String</td><td>No</td><td>Overall health state from the view of this resource:
"OK", "Warning",
"Critical"</td></tr>
</table>
</td></tr>
<tr><td>name</td><td>String</td><td>No</td><td>Optional name of the component</td></tr>
<tr><td>description</td><td>String</td><td>No</td><td>Optional description of the component</td></tr>
<tr><td>frameSize</td><td>Number</td><td>No</td><td>MAC Frame size in bytes</td></tr>
<tr><td>speedMbps</td><td>Number</td><td>No</td><td>The current speed in Mbps of this network interface</td></tr>
<tr><td>autoSense</td><td>Boolean</td><td>No</td><td>Indicates if the NIC automatically configures the speed and duplex</td></tr>
<tr><td>fullDuplex</td><td>Boolean</td><td>No</td><td>Indicates if the NIC is in Full Duplex mode or not</td></tr>
<tr><td>macAddress</td><td>String</td><td>No</td><td>This is the currently assigned MAC address for this NIC</td></tr>
<tr><td>factoryMacAddress</td><td>String</td><td>No</td><td>Factory assigned MAC address</td></tr>
<tr><td>ipv4Addresses</td><td>Array: Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>address</td><td>String</td><td>Yes</td><td>Ipv4 address</td></tr>
<tr><td>subnetMask</td><td>String</td><td>No</td><td>Ipv4 subnet mask</td></tr>
<tr><td>addressOrigin</td><td>String</td><td>No</td><td>Ipv4 address origin
"DHCP",
"Static"</td></tr>
<tr><td>gateway</td><td>String</td><td>No</td><td>Ipv4 gateway for this address</td></tr>
</table>
</td></tr>
<tr><td>ipv4StaticAddresses</td><td>Array: Object</td><td>No</td><td>The array of static IPv4 addresses assigned to this interface
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>address</td><td>String</td><td>Yes</td><td>Ipv4 address</td></tr>
<tr><td>subnetMask</td><td>String</td><td>No</td><td>Ipv4 subnet mask</td></tr>
<tr><td>addressOrigin</td><td>String</td><td>No</td><td>Ipv4 address origin
"DHCP",
"Static"</td></tr>
<tr><td>gateway</td><td>String</td><td>No</td><td>Ipv4 gateway for this address</td></tr>
</table>
</td></tr>
<tr><td>ipv6Addresses</td><td>Array: Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>address</td><td>String</td><td>Yes</td><td>Ipv6 address</td></tr>
<tr><td>prefixLength</td><td>Number</td><td>No</td><td>Ipv6 Address Prefix Length</td></tr>
<tr><td>addressOrigin</td><td>String</td><td>No</td><td>Ipv4 address origin
"DHCP",
"Static",
"SLAAC"</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>addressState</td><td>String</td><td>No</td><td>Ipv6 address state: "Preferred", "Deprecated", "Tentative", "Failed"</td></tr>
</table>
</td></tr>
<tr><td>ipv6StaticAddresses</td><td>Array: Object</td><td>No</td><td>The array of static IPv6 addresses assigned to this interface
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>address</td><td>String</td><td>Yes</td><td>Ipv6 address</td></tr>
<tr><td>prefixLength</td><td>Number</td><td>No</td><td>Ipv6 Address Prefix Length</td></tr>
<tr><td>addressOrigin</td><td>String</td><td>No</td><td>Ipv6 address origin "DHCP", "Static", "SLAAC"</td></tr>
<tr><td>addressState</td><td>String</td><td>No</td><td>Ipv6 address state: "Preferred", "Deprecated", "Tentative", "Failed"</td></tr>
</table>
</td></tr>
<tr><td>ipv6DefaultGateway</td><td>String</td><td>No</td><td>The ipv6 default gateway address</td></tr>
<tr><td>ipv6DefaultStaticGateways</td><td>Array: String</td><td>No</td><td>The array of Ipv6 default gateway addresses</td></tr>
<tr><td>staticNameServers</td><td>Array: String</td><td>No</td><td>The array of DNS server IP addresses to be used when DHCP provisioning is not in enabled for name server configuration.</td></tr>
<tr><td>supportedProtocols</td><td>Array: String</td><td>Yes</td><td>List of supported protocols. Allowed values: "RoCEv2", "iWARP", "iSCSI"</td></tr>
<tr><td>collections</td><td>Array: Object</td><td>Yes</td><td>Subcomponents collections
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>name</td><td>String</td><td>Yes</td><td>Collection name</td></tr>
<tr><td>type</td><td>String</td><td>Yes</td><td>Collection type "EthernetSwitchPortVlans"</td></tr>
</table>
</td></tr>
<tr><td>firmwareVersion</td><td>String</td><td>No</td><td>A version of network interface firmware</td></tr>
<tr><td>Oem</td><td>Object</td><td>No</td><td>OEM specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "frameSize": <number>,
                "speedMbps": <number>,
                "autoSense": <boolean>,
                "fullDuplex": <boolean>,
```

```
                "macAddress": <string>,
                "factoryMacAddress": <string>,
                "ipv4Addresses": [
                        {
                                "address": <string>,
                                "subnetMask": <string>,
                                "addressOrigin": <string>,
                                "gateway": <string>
                        }
                …
                ]
                "ipv6Addresses": [
                        {
                                "address": <string>,
                                "prefixLength": <number>,
                                "addressOrigin": <{"DHCP", "Static", "SLAAC"}>,
                                "addressState": <{"Preferred", "Deprecated",
"Tentative", "Failed" }>
                        }
                ...
                ],
                "ipv6DefaultGateway": <string>,
                "maxIPv6StaticAddresses": <number>,
                "supportedProtocols": [ <{"RoCEv2", "iWARP", "iSCSI"}> ],
                "defaultVlan": <number>,
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"EthernetSwitchPortVlans"}>
                        }
                ...
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "frameSize": 1520,
                "speedMbps": 1000,
                "autoSense": true,
                "fullDuplex":false,
                "macAddress": "AA:BB:CC:DD:EE:FF",
                "factoryMacAddress": "AA:BB:CC:DD:EE:FF",
                "ipv4Addresses": [
                        {
                                "address": "10.0.2.10",
                                "subnetMask": "255.255.255.0",
                                "addressOrigin": "DHCP",
                                "gateway": "10.0.2.1"
                        }
                ],
                "ipv6Addresses": [
```

```
                        {
                                "address": "fe80::1ec1:deff:fe6f:1c37",
                                "prefixLength": 16,
                                "addressOrigin": "DHCP",
                                "addressState": "Preferred"
                        }
                ],
                "ipv6DefaultGateway": "fe80::1ec1:deff:febd:67e3",
                "maxIPv6StaticAddresses": 1,
                "supportedProtocols": [ "RoCEv2" ],
                "defaultVlan": 4090,
                "collections": [
                        {
                                "name": "EthernetSwitchPortVlans",
                                "type": "EthernetSwitchPortVlans"
                        }
                ],
                "oem": {}
        },
        "id": 987
}
```

5.29 Set Network Interface Attributes

The `"setComponentAttributes"` described in Section 5.5, setComponentAttributes, allows configuration of the network interface attributes listed in the following table.

Table 53. Configurable Network Interface Attributes

Attribute	Type	Description
`frameSize`	Number	MAC frame size in bytes
`speedMbps`	Number	The network interface speed in Mbps
`autosense`	Boolean	Indicates if the interface automatically configures the speed and duplex
`Oem`	Object	OEM specific data

5.30 getTrustedModuleInfo

The `"getTrustedModuleInfo"` command retrieves detailed information about a trusted module.

5.30.1 Request

Table 54. getTrustedModuleInfo Request

Parameters	Type	Mandatory	Description
`trustedModule`	String	Yes	Managed trusted module UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getTrustedModuleInfo",
        "params": {
                "trustedModule": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getTrustedModuleInfo",
        "params": {
                "trustedModule": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.30.2 Response

Table 55. getTrustedModuleInfo Response

Result	Type	Mandatory	Description
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr><tr><td>health</td><td>String</td><td>No</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
name	String	No	Optional name of the component
description	String	No	Optional description of the component
interfaceType	String	No	Trusted module interface type. Supported values: "TPM1_2", "TPM2_0"
firmwareVersion	String	No	Firmware version of the trusted module
Oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,","InTest", "Starting", "Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "interfaceType": <{"TPM1_2","TPM2_0"}>,
                "firmwareVersion": <String>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "interfaceType": "TPM2_0",
                "firmwareVersion": "1.0",
                "oem": {}
        },
        "id": 987
}
```

5.31 Set Trusted Module Attributes

The "setComponentAttributes" described in Section 5.5, setComponentAttributes, allows configuration of the trusted module attributes listed in the following table.

Table 56. Configurable Trusted Module Attributes

Attribute	Type	Description
deviceEnabled	Boolean	Enables or disables trusted module
clearOwnership	Boolean	Clears trusted module ownership
oem	Object	OEM specific data

5.32 getFabricInfo

The "getFabricInfo" command retrieves detailed information about single fabric.

5.32.1 Request

Table 57. getFabricInfo Request

Parameters	Type	Mandatory	Description
fabric	String	Yes	Managed fabric UUID.

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getFabricInfo",
        "params": {
                "fabric": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getFabricInfo",
        "params": {
                "fabric": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.32.2 Response

Table 58. getFabricInfo Response

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>status</td><td>Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component:
"Enabled",
"Disabled",
"StandbyOffline",
"StandbySpare,"<br>"InTest",
"Starting",
"Absent”,
"UnavailableOffline"</td></tr>
<tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource:
"OK", "Warning",
"Critical"</td></tr>
</table>
</td></tr>
<tr><td>name</td><td>String</td><td>No</td><td>Optional name of the component</td></tr>
<tr><td>description</td><td>String</td><td>No</td><td>Optional description of the component</td></tr>
<tr><td>collections</td><td>Array: Object</td><td>Yes</td><td>Subcomponents collections
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>name</td><td>String</td><td>Yes</td><td>Collection name</td></tr>
<tr><td>type</td><td>String</td><td>Yes</td><td>Collection type:
"Zones",
"Switches",
"Endpoints"</td></tr>
</table>
</td></tr>
<tr><td>protocol</td><td>String</td><td>Yes</td><td>The value of this property shall contain the protocol used by the switch:
"PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB", "NVMe", "FC", "iSCSI", "FCoE", "NVMeOverFabrics", "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS", "FTP", "FTPS"</td></tr>
<tr><td>oemProtocol</td><td>String</td><td>No</td><td>Can be specified for OEM-defined protocol such as "FPGA-oF"</td></tr>
<tr><td>oem</td><td>Object</td><td>No</td><td>OEM specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,","InTest", "Starting", "Absent”, "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"Zones", "Switches", "Endpoints"}>
                        }
                ...
```

```
                ],
                "protocol": <{"PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB", "NVMe",
"FC", "iSCSI", "FCoE", "NVMeOverFabrics", "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS",
"FTP", "FTPS"}>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "collections": [
                        {
                                "name": "Zones",
                                "type": "Zones"
                        },
                        {
                                "name": "Switches",
                                "type": "Switches"
                        },
                        {
                                "name": "Endpoints",
                                "type": "Endpoints"
                        },
                ],
                "protocol": "PCIe",
                "oem": {}
        },
        "id": 987
}
```

5.33 getSwitchInfo

The "getSwitchInfo" command retrieves detailed information about a single switch.

5.33.1 Request

Table 59. getSwitchInfo Request

Parameters	Type	Mandatory	Description
switch	String	Yes	Managed switch UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getSwitchInfo",
        "params": {
                "switch": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getSwitchInfo",
        "params": {
                "switch": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.33.2 Response

Table 60. getSwitchInfo Response

Result	Type	Mandatory	Description
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent, "UnavailableOffline"</td></tr><tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
name	String	No	Optional name of the component
description	String	No	Optional description of the component
switchId	String	Yes	Switch identifier
chassis	String	Yes	UUID of the chassis is the physical container for the PCIe switch
assetTag	String	No	Asset tag
fruInfo	Object	No	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>manufacturer</td><td>String</td><td>No</td><td>Manufacturer name</td></tr><tr><td>modelNumber</td><td>String</td><td>No</td><td>Model</td></tr><tr><td>serialNumber</td><td>String</td><td>No</td><td>Serial number</td></tr><tr><td>partNumber</td><td>String</td><td>No</td><td>Part number</td></tr></table>
sku	String	No	PCIe switch SKU
collections	Array: Object	Yes	Subcomponents collections <table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>name</td><td>String</td><td>Yes</td><td>Collection name</td></tr><tr><td>type</td><td>String</td><td>Yes</td><td>Collection type: "Ports"</td></tr></table>
indicatorLED	String	No	The state of the indicator LED used to identify the drive: "Lit", "Blinking", "Off"
powerState	String	No	The value of this property shall contain the power state of the switch: "On", "Off", "PoweringOn", "PoweringOff"

Result	Type	Mandatory	Description
allowedActions	Array: String	Yes	List of allowed actions: "On", "ForceOff", "GracefulShutdown", "ForceRestart", "Nmi", "GracefulRestart", "ForceOn", "PushPowerButton"
protocol	String	Yes	The value of this property shall contain the protocol used by the switch: "PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB", "NVMe", "FC", "iSCSI", "FCoE", "NVMeOverFabrics", "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS", "FTP", "FTPS"
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "switchId": <string>,
                "chassis": <string>,
                "assetTag": <string>,
                "fruInfo": {
                        "manufacturer": <string>,
                        "modelNumber": <string>,
                        "serialNumber": <string>,
                        "partNumber": <string>
                },
                "sku": <string>,
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"Ports"}>
                        }
                ...
                ],
                "indicatorLED": <{"Lit", "Blinking", "Off"}>,
                "powerState": <{"On", "Off", "PoweringOn", "PoweringOff"}>,
                "allowedActions": [
                        <{"On", "ForceOff", "GracefulShutdown", "ForceRestart",
"Nmi", "GracefulRestart", "ForceOn", "PushPowerButton"}>,
                        …
                ],
                "protocol": <{"PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB", "NVMe",
"FC", "iSCSI", "FCoE", "NVMeOverFabrics", "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS",
"FTP", "FTPS"}>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "switchId": "PCIe Switch #1",
                "chassis": "123e4567-e89b-12d3-a456-426655440000",
                "assetTag": "PCie switch #2",
                "fruInfo": {
                        "manufacturer": "Switch Manufacturer",
                        "modelNumber": "Switch Model",
                        "serialNumber": "123fed3029c-b23394-12",
                        "partNumber": "29ee2220939"
                },
                "sku": "B1"
                "collections": [
                        {
                                "name": "Ports",
                                "type": "Ports"
                        }
                ],
                "indicatorLED": "Lit",
                "powerState": "On",
                "allowedActions": ["On", "ForceOff"],
                "protocol": "PCIe",
                "oem": {}
        },
        "id": 987
}
```

5.34 Set Switch Attributes

The `"setComponentAttributes"` described in Section 5.5, setComponentAttributes, allows configuration of the switch attributes listed in the following table.

Table 61. Configurable Switch Attributes

Attribute	Type	Description
`reset`	String	Type of reset to perform. Currently supported values listed in allowedActions field. Possible values: `"On"`, `"ForceOff"`, `"GracefulShutdown"`, `"ForceRestart"`, `"Nmi"`, `"GracefulRestart"`, `"ForceOn"`, `"PushPowerButton"`
`oem`	Object	OEM specific data

5.35 getPortInfo

The `"getPortInfo"` command retrieves detailed information about a single port.

5.35.1 Request

Table 62. getPortInfo Request

Parameters	Type	Mandatory	Description
`port`	String	Yes	Managed port UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getPortInfo",
        "params": {
                  "port": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getPortInfo",
        "params": {
                  "port": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.35.2 Response

Table 63. getPortInfo Response

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>status</td><td>Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component:
"Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent, "UnavailableOffline"</td></tr>
<tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource:
"OK", "Warning", "Critical"</td></tr>
</table>
</td></tr>
<tr><td>name</td><td>String</td><td>No</td><td>Optional name of the component</td></tr>
<tr><td>description</td><td>String</td><td>No</td><td>Optional description of the component</td></tr>
<tr><td>portId</td><td>String</td><td>Yes</td><td>Port Identifier (port index, name, etc.)</td></tr>
<tr><td>portType</td><td>String</td><td>Yes</td><td>Port type:
"UpstreamPort", "DownstreamPort", "InterswitchPort", "ManagementPort", "BidirectionalPort", "UnconfiguredPort"</td></tr>
<tr><td>cableIds</td><td>Array: String</td><td>No</td><td>Cable identifiers</td></tr>
<tr><td>speedGBps</td><td>Number</td><td>Yes</td><td>Port speed in Gbps</td></tr>
<tr><td>width</td><td>Number</td><td>Yes</td><td>Port width (PCIe lanes number)</td></tr>
<tr><td>maxSpeedGbps</td><td>Number</td><td>Yes</td><td>Max port speed in Gbps</td></tr>
<tr><td>maxWidth</td><td>Number</td><td>Yes</td><td>Max port width (PCIe lines number)</td></tr>
</table>

Result	Type	Mandatory	Description
allowedActions	Array: String	Yes	List of allowed actions: "ForceOff", "ForceRestart", "ForceOn"
protocol	String	No	Protocol name used by port: "PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB", "NVMe", "FC", "iSCSI", "FCoE", "NVMeOverFabrics", "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS", "FTP", "FTPS"
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare," "InTest", "Starting", "Absent, "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "portId": <string>,
                "portType": <{"UpstreamPort", "DownstreamPort", "InterswitchPort",
"ManagementPort", "BidirectionalPort", "UnconfiguredPort"}>,
                "cableIds": [<string>]
                "speedGBps": <number>,
                "width": <number>,
                "maxSpeedGBps": <number>,
                "maxWidth": <number>,
                "allowedActions: [
                        <{"ForceOff", "ForceRestart", "ForceOn"}>,
                        …
                ],
                "protocol": <{"PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB", "NVMe",
"FC", "iSCSI", "FCoE", "NVMeOverFabrics", "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS",
"FTP", "FTPS"}>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "portId": "1"
                "portType": "DownstreamPort",
                "cableIds": ["Host1:3-Switch1:1"]
                "speedGbps ": 4,
                "width": 2,
                "maxSpeedGTps": 8,
                "maxWidth": 4,
                "allowedActions": [ "ForceRestart" ],
                "protocol" "PCIe",
                "oem": {}
```

```
        },
        "id": 987
}
```

5.36 Set Port Attributes

The "setComponentAttributes" described in Section 5.5, setComponentAttributes, allows configuration of the Switch Port Attributes listed in the following table.

Table 64. Configurable Port Attributes

Attribute	Type	Description
oem	Object	OEM specific data
powerState	String	Trigger power action on port: "ForceOff", "ForceRestart", "ForceOn".

5.37 getEndpointInfo

The "getEndpointInfo" command retrieves detailed information about a single endpoint.

5.37.1 Request

Table 65. getEndpointInfo Request

Parameters	Type	Mandatory	Description
endpoint	String	Yes	Managed endpoint UUID.

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getEndpointInfo",
        "params": {
                "endpoint": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getEndpointInfo",
        "params": {
                "endpoint": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.37.2 Response

Table 66. getEndpointInfo Response

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>status</td><td>Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component:
"Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent, "UnavailableOffline"</td></tr>
<tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource:
"OK", "Warning", "Critical"</td></tr>
</table>
</td></tr>
<tr><td>name</td><td>String</td><td>No</td><td>Optional name of the component</td></tr>
<tr><td>description</td><td>String</td><td>No</td><td>Optional description of the component</td></tr>
<tr><td>ipTransportDetail</td><td>Array: Object</td><td>Yes</td><td>This property shall contain IP transport data of endpoint. Refer to Table 56 for details.</td></tr>
<tr><td>identifiers</td><td>Array: Object</td><td>No</td><td>This property shall contain a list of all known durable names for the associated drive. Refer to Table 68 for details.</td></tr>
<tr><td>connectedEntities</td><td>Array: Object</td><td>No</td><td>List of the entities connected to the endpoint.
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>type</td><td>String</td><td>Yes</td><td>This indicates entity type:
"RootComplex", "StorageInitiator", "NetworkController", "Drive", "StorageExpander", "DisplayController", "Bridge", "Processor", "Volume", "System"</td></tr>
<tr><td>role</td><td>String</td><td>Yes</td><td>Represents the entity role:
"Initiator", "Target", "Both"</td></tr>
<tr><td>accessMode</td><td>String</td><td>No</td><td>The allowed access mode of the entity. Allowed values:
"ReadWrite", "ReadOnly"</td></tr>
<tr><td>entity</td><td>String</td><td>No</td><td>UUID of the entity component.</td></tr>
<tr><td>identifiers</td><td>Array: Object</td><td>No</td><td>This resource should represent a connected entity identifier. Refer to Table 68 for details.</td></tr>
<tr><td>lun</td><td>Number</td><td>No</td><td>Logical unit number used in iSCSI storage.</td></tr>
</table>
</td></tr>
<tr><td>protocol</td><td>String</td><td>No</td><td>Protocol name used by port:
"PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB", "NVMe", "FC", "iSCSI", "FCoE", "NVMeOverFabrics", "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS", "FTP", "FTPS"</td></tr>
<tr><td>username</td><td>String</td><td>No</td><td>Username for endpoint access control</td></tr>
</table>

Result	Type	Mandatory	Description
password	String	No	Password for endpoint access control – must not be displayed for security reasons
collections	Array: Object	Yes	Subcomponents collections **Attribute** / **Type** / **Mandatory** / **Description** name / String / Yes / Collection name type / String / Yes / Collection type: "Ports"
oemProtocol	String	No	Can be specified for OEM-defined protocol such as "FPGA-oF"
oem	Object	No	OEM specific data

Table 67. IP Transport Details

Attribute	Type	Mandatory	Description
protocol	String	Yes	Specifies the transport protocol. Possible values: "PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB", "NVMe", "FC", "iSCSI", "FCoE", "FCP", "FICON", "NVMeOverFabrics", "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS", "FTP", "SFTP", "iWARP", "RoCE", "RoCEv2"
ipv4Address	Object	Yes	The IPv4 address assigned to the associated Endpoint. **Attribute** / **Type** address / String subnetMask / String addressOrigin / String gateway / String Address origin values are: "DHCP", "Static".
ipv6Address	String	Yes	The IPv6 address assigned to the associated Endpoint. **Attribute** / **Type** address / String prefixLength / Number addressOrigin / String addressState / String Address origin values are: "DHCP", "Static", "SLAAC". Address state values are: "Preferred", "Deprecated", "Tentative", "Failed".
port	Number	Yes	This property specifies the UDP or TCP port used by the associated Endpoint.
interface	String	No	Specifies the interface on which transport is available.
protocol	String	Yes	Specifies the transport protocol. Possible values: "PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB", "NVMe", "FC", "iSCSI", "FCoE", "FCP", "FICON", "NVMeOverFabrics", "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS", "FTP", "SFTP", "iWARP", "RoCE", "RoCEv2"

Table 68. Identifiers

Attribute	Type	Mandatory	Description
durableName	String	Yes	This indicates the worldwide, a persistent name of the resource.

durableNameFormat	String	Yes	Represents the format of the durableName property: "NAA", "iQN", "FC_WWN", "UUID", "EUI", "NQN", "NSID".

Serialization:

```
{
        "jsonrpc": "2.0",
        "params": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
                                  "StandbySpare", "InTest", "Starting",
                                  "Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "protocol": <{"PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB",
                              "NVMe", "FC", "iSCSI", "FCoE", "NVMeOverFabrics",
                              "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS",
                              "FTP", "FTPS"}>,
                "connectedEntities": [
                {
                        "type": <{"RootComplex", "StorageInitiator",
                                  "NetworkController", "Drive", "StorageExpander",
                                  "DisplayController", "Bridge", "Processor",
                                  "Volume", "System"}>,
                        "role": <{"Initiator", "Target", "Both"}>,
                        "entity": <string>,
                        "identifiers": [{
                                "durableName": <string>,
                                "durableNameFormat": <{"NAA", "iQN", "FC_WWN",
                                                       "UUID", "EUI", "NQN",
                                                       "NSID"}>
                        },
                        …
                        ],
                        "lun": <number>
                },
                …
                ],
                "identifiers": [{
                        "durableName": <String>,
                        "durableNameFormat": <{"NAA", "iQN", "FC_WWN", "UUID",
                                               "EUI", "NQN", "NSID"}>
                },
                …
                ],
                "ipTransportDetails": [
                {
                        "protocol": <{"PCIe", "AHCI", "UHCI", "SAS", "SATA",
                                      "USB", "NVMe", "FC", "iSCSI", "FCoE",
                                      "FCP", "FICON", "NVMeOverFabrics",
                                      "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS",
                                      "FTP", "SFTP", "iWARP", "RoCE", "RoCEv2" }>,
                        "ipv4Address": {
                                "address": <String>,
                                "subnetMask": <String>,
                                "addressOrigin": <{"DHCP", "Static"}>,
                                "gateway": <String>
                        },
                        "ipv6Address": {
                                "address": <String>,
                                "prefixLength": <Number>,
```

```
                                    "addressOrigin": <{"DHCP", "Static", "SLAAC"}>,
                                    "addressState": <{"Preferred", "Deprecated",
                                                      "Tentative", "Failed"}>
                            },
                            "port": <Number>,
                            "interface": <String>
                    },
                    …
                    ],
                    "oem": <Object>
            },
            "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addEndpoint",
        "params": {
                "fabric": "41d22dd0-a6dc-4fea-a789-1e34db12d881",
                "ipTransportDetails": [
                        {
                                "protocol": "RoCEv2",
                                "ipv4Address": {
                                        "address": "192.168.1.128",
                                        "subnetMask": "255.255.255.0",
                                        "addressOrigin": "DHCP",
                                        "gateway": "192.168.1.1"
                                },
                                "port": 6623,
                                "interface": "ed364562-f011-5c28-deca-14d3a6741238"
                        }
                ],
                "identifiers": [
                        {
                                "durableName": "123e4567-e89b-12d3-a456-
426655440000",
                                "durableNameFormat": "UUID"
                        }
                ],
                "connectedEntities": [
                        {
                                "type": "Volume",
                                "role": "Target",
                                "entity": "123e4567-e89b-12d3-a456-426655440000" ,
            "lun": null
                        },
                        …
                ],
                "oem": {}
        },
        "id": 987
}
```

5.38 addEndpoint

The "addEndpoint" command creates new endpoint with associated resources.

5.38.1 Request

Table 69 addEndpoint Request

Parameters	Type	Mandatory	Description
fabric	String	Yes	Fabric UUID to add the endpoint to
ipTransportDetails	Array: Object	No	This array shall be empty or contain objects with the following attributes. Refer to Table 70.
identifiers	Array: Object	Yes	This property shall contain a list of all known durable names for the associated drive. Refer to Table 68 for details.
connectedEntities	Array: Object	Yes	List of the entities connected to the endpoint. **Attribute** / **Type** / **Mandatory** / **Description** role / String / Yes / Represents the entity role: "Initiator", "Target", "Both" entity / String / No / UUID of the entity component. identifiers / Array: Object / No / This resource should represent a connected entity identifier. Refer to Table 54 for details. lun / Number / No / Logical unit number used in iSCSI storage.
username	String	No	Username for endpoint access control
password	String	No	Password for endpoint access control
Oem	Object	No	OEM-specific data

Table 70. IP Transport Details

Attribute	Type	Mandatory	Description
protocol	String	Yes	Specifies the transport protocol. Possible values: "PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB", "NVMe", "FC", "iSCSI", "FCoE", "FCP", "FICON", "NVMeOverFabrics", "SMB", "NFSv3", "NFSv4", "HTTP", "HTTPS", "FTP", "SFTP", "iWARP", "RoCE", "RoCEv2"
ipv4Address	Object	Yes	The IPv4 address assigned to the associated Endpoint. **Attribute** / **Type** address / String subnetMask / String addressOrigin / String gateway / String Address origin values are: "DHCP", "Static".
ipv6Address	String	Yes	The IPv6 address assigned to the associated Endpoint. **Attribute** / **Type** address / String prefixLength / Number addressOrigin / String addressState / String Address origin values are: "DHCP", "Static", "SLAAC". Address state values are: "Preferred", "Deprecated", "Tentative", "Failed".
port	Number	Yes	This property specifies the UDP or TCP port used by the associated Endpoint.

Attribute	Type	Mandatory	Description
interface	String	No	Specifies the interface on which transport is available

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "addEndpoint",
        "params": {
                "fabric": <String>,
                "ipTransportDetails": [
                {
                        "protocol": <{"PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB",
                                "NVMe", "FC", "iSCSI", "FCoE", "FCP", "FICON",
                                "NVMeOverFabrics", "SMB", "NFSv3", "NFSv4",
                                "HTTP", "HTTPS", "FTP", "SFTP", "iWARP",
                                "RoCE", "RoCEv2" }>,
                        "ipv4Address": {
                                "address": <String>,
                                "subnetMask": <String>,
                                "addressOrigin": <{"DHCP", "Static"}>,
                                "gateway": <String>
                        },
                        "ipv6Address": {
                                "address": <String>,
                                "prefixLength": <Number>,
                                "addressOrigin": <{"DHCP", "Static", "SLAAC"}>,
                                "addressState": <{"Preferred", "Deprecated",
                                                "Tentative", "Failed"}>
                                },
                                "port": <Number>,
                                "interface": <String>
                        }
                ],
                "identifiers": [
                {
                        "durableName": <String>,
                        "durableNameFormat": <{"NAA", "iQN", "FC_WWN", "UUID",
                                                "EUI", "NQN", "NSID"}>
                },
                …
                ],
                "connectedEntities": [
                {
                        "role": <{"Initiator", "Target", "Both"}>,
                        "entity": <String>,
                        "identifiers": [
                        {
                                "durableName": <String>,
                                "durableNameFormat": <{"NAA", "iQN", "FC_WWN",
                                                        "UUID", "EUI", "NQN",
                                                        "NSID"}>
                        },
                        …
                        ],
                        "lun": <number>
                },
                …
                ],
                "oem": <Object>
        },
```

```
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addEndpoint",
        "params": {
                "fabric": "41d22dd0-a6dc-4fea-a789-1e34db12d881",
                "ipTransportDetails": [
                        {
                                "protocol": "RoCEv2",
                                "ipv4Address": {
                                        "address": "192.168.1.128",
                                        "subnetMask": "255.255.255.0",
                                        "addressOrigin": "DHCP",
                                        "gateway": "192.168.1.1"
                                },
                                "port": 6623,
                                "interface": "ed364562-f011-5c28-deca-14d3a6741238"
                        }
                ],
                "identifiers": [
                        {
                                "durableName": "123e4567-e89b-12d3-a456-
426655440000",
                                "durableNameFormat": "UUID"
                        }
                ],
                "connectedEntities": [
                        {
                                "role": "Target",
                                "entity": "123e4567-e89b-12d3-a456-426655440000",
                                "lun": null
                        },
                        …
                ],
                "oem": {}
        },
        "id": 987
}
```

5.38.2 Response

Table 71 addEndpoint Response

Parameters	Type	Mandatory	Description
endpoint	String	Yes	Created endpoint UUID
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "endpoint": <String>,
                "oem": <Object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "endpoint": "653d2a64-1c97-4ed3-f773-6ca6454a037b",
                "oem": {}
        },
        "id": 1127
}
```

5.39 deleteEndpoint

The "deleteEndpoint" command deletes existing endpoint.

5.39.1 Request

Table 72 deleteEndpoint Request

Parameters	Type	Mandatory	Description
endpoint	String	Yes	Managed endpoint UUID.

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "deleteEndpoint",
        "params": {
                "endpoint": <String>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "deleteEndpoint",
        "params": {
                "endpoint": "7ec52201-bd23-4cc2-1256-6262e544fda8"
        },
        "id": 1218
}
```

5.39.2 Response

Table 73 deleteEndpoint Response

Result	Type	Mandatory	Description
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
        },
```

```
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 1218
}
```

5.40 Set Endpoint Attributes

The "setComponentAttributes" described in Section 5.5, setComponentAttributes, allows configuration of the Endpoint Attributes listed in the following table.

Table 74. Configurable Endpoint Attributes

Attribute	Type	Description
username	String	Username for endpoint access control
password	String	Password for endpoint access control
oem	Object	OEM specific data

5.41 getPCIeDeviceInfo

The "getPCIeDeviceInfo" command retrieves detailed information about a single PCIe* device.

5.41.1 Request

Table 75. getPCIeDeviceInfo Request

Parameters	Type	Mandatory	Description
device	String	Yes	Managed PCIe device UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getPCIeDeviceInfo",
        "params": {
                "device": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getPCIeDeviceInfo",
        "params": {
                "device": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.41.2 Response

Table 76. getPCIeDeviceInfo Response

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>status</td><td>Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component:
"Enabled",
"Disabled",
"StandbyOffline",
"StandbySpare,"<br>"InTest",
"Starting",
"Absent",
"UnavailableOffline"</td></tr>
<tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource:
"OK", "Warning", "Critical"</td></tr>
</table>
</td></tr>
<tr><td>name</td><td>String</td><td>No</td><td>Optional name of the component</td></tr>
<tr><td>description</td><td>String</td><td>No</td><td>Optional description of the component</td></tr>
<tr><td>deviceId</td><td>String</td><td>Yes</td><td>Device Identifier</td></tr>
<tr><td>deviceClass</td><td>String</td><td>Yes</td><td>Device type:
"SingleFunction", "MultiFunction"</td></tr>
<tr><td>speedGbps</td><td>Number</td><td>Yes</td><td>Link speed in Gbps</td></tr>
<tr><td>width</td><td>Number</td><td>Yes</td><td>Link width</td></tr>
<tr><td>maxSpeedGbps</td><td>Number</td><td>No</td><td>Max link speed in Gbps</td></tr>
<tr><td>maxWidth</td><td>Number</td><td>No</td><td>Max link width</td></tr>
<tr><td>firmwareVersion</td><td>String</td><td>No</td><td>Firmware version</td></tr>
<tr><td>chassis</td><td>String</td><td>Yes</td><td>UUID of the chassis is the physical container for the PCIe device</td></tr>
<tr><td>assetTag</td><td>String</td><td>No</td><td>Asset tag</td></tr>
<tr><td>fruInfo</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>manufacturer</td><td>String</td><td>No</td><td>Manufacturer name</td></tr>
<tr><td>modelNumber</td><td>String</td><td>No</td><td>Model</td></tr>
<tr><td>serialNumber</td><td>String</td><td>No</td><td>Serial number</td></tr>
<tr><td>partNumber</td><td>String</td><td>No</td><td>Part number</td></tr>
</table>
</td></tr>
<tr><td>sku</td><td>String</td><td>No</td><td>PCIe device SKU</td></tr>
<tr><td>collections</td><td>Array: Object</td><td>Yes</td><td>Subcomponents collections
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>name</td><td>String</td><td>Yes</td><td>Collection name</td></tr>
<tr><td>type</td><td>String</td><td>Yes</td><td>Collection type:
"PCIeFunctions"</td></tr>
</table>
</td></tr>
<tr><td>oem</td><td>Object</td><td>No</td><td>OEM specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,"
                                                                "InTest", "Starting",
"Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "deviceId": <string>,
                "deviceClass": <{"SingleFunction", "MultiFunction"}>,
                "speedGbps": <number>,
                "width": <number>,
                "maxSpeedGbps": <number>,
                "maxWidth": <number>,
                "firmwareVersion": <string>,
                "chassis": <string>,
                "assetTag": <string>,
                "fruInfo": {
                        "manufacturer": <string>,
                        "modelNumber": <string>,
                        "serialNumber": <string>,
                        "partNumber": <string>
                },
                "sku": <string>,
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"PCIeFunctions"}>
                        }
                …
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "deviceId": "Device 3-12",
                "deviceType": "Drive",
                "speedGbps": 8,
                "width": 4,
                "maxSpeedGbps": 8,
                "maxWidth": 4,
                "firmwareVersion": "ABC123debug",
                "chassis": "123e4567-e89b-12d3-a456-426655440000",
                "assetTag": "NVMe SSD #1"
                "fruInfo": {
                        "manufacturer": "Manufacturer Name",
                        "modelNumber": "Model",
                        "serialNumber": "123fed3029c-b23394-12",
```

```
                        "partNumber": "29ee2220939",
                },
                "sku": "Rev.1",
                "collections": [
                        {
                                "name": "Functions",
                                "type": "PCIeFunctions"
                        }
                ],
                "oem": {}
        },
        "id": 987
}
```

5.42 Set PCIe* Device Attributes

The "setComponentAttributes" described in Section 5.5, setComponentAttributes, allows configuration of the PCIe* device in the following table.

Table 77. Set PCIe* Device Attributes

Attribute	Type	Description
assetTag	String	Tag assigned to the asset.
oem	Object	OEM specific data

5.43 getPCIeFunctionInfo

The "getPCIeFunctionInfo" command retrieves detailed information about a single PCIe* function.

5.43.1 Request

Table 78. getPCIeFunctionInfo Request

Parameters	Type	Mandatory	Description
function	String	Yes	Managed PCIe function UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getPCIeFunctionInfo",
        "params": {
                "function": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getPCIeFunctionInfo",
        "params": {
                "function": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.43.2 Response

Table 79. getPCIeFunctionInfo Response

Result	Type	Mandatory	Description
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr><tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
name	String	No	Optional name of the component
description	String	No	Optional description of the component
functionId	String	Yes	Function identifier
functionType	String	Yes	Function type: "Physical", "Virtual"
deviceClass	String	Yes	Function device class: "UnclassifiedDevice", "MassStorageController", "NetworkController", "DisplayController", "MultimediaController", "MemoryController", "Bridge", "CommunicationController", "GenericSystemPeripheral", "InputDeviceController", "DockingStation", "Processor", "SerialBusController", "WirelessController", "IntelligentController", "SatelliteCommunicationsController", "EncryptionController", "SignalProcessingController", "ProcessingAccelerators", "NonEssentialInstrumentation", "Coprocessor", "UnassignedClass"
pciDeviceId	String	No	PCIe device identifier
pciVendorId	String	No	PCIe device vendor identifier
pciClassCode	String	No	PCIe device class code
pciRevisionId	String	No	PCIe device revision identifier
pciSubsystemId	String	No	PCIe device subsystem identifier
pciSubsystemVendorId	String	No	PCIe device subsystem vendor identifier
functionalDevice	String	Yes	UUID of the functional device (drive, network interface, etc.) corresponding to the PCIe function
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
```

```
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "functionId": <string>,
                "functionType": <{"Physical", "Virtual"}>,
                "deviceClass": <{"UnclassifiedDevice", "MassStorageController",
"NetworkController", "DisplayController", "MultimediaController", "MemoryController",
"Bridge", "CommunicationController", "GenericSystemPeripheral",
"InputDeviceController", "DockingStation", "Processor", "SerialBusController",
"WirelessController", "IntelligentController", "SatelliteCommunicationsController",
"EncryptionController", "SignalProcessingController", "ProcessingAccelerators",
"NonEssentialInstrumentation", "Coprocessor", "UnassignedClass"}>,
                "pciDeviceId": <string>,
                "pciVendorId": <string>,
                "pciClassCode": <string>,
                "pciRevisionId": <string>,
                "pciSubsystemId": <string>,
                "pciSubsystemVendorId": <string>,
                "functionalDevice": <string>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "functionId": "1",
                "functionType": "Physical",
                "deviceClass": "MassStorageController",
                "pciDeviceId": "0xABCD",
                "pciVendorId": "0xABCD",
                "pciClassCode": "0x10802",
                "pciRevisionId": "0x01",
                "pciSubsystemId": "0xABCD",
                "pciSubsystemVendorId": "0xABCD",
                "functionalDevice": "863e4567-e87b-64d3-a489-987656540000",
                "oem": {}
        },
        "id": 987
}
```

5.44 getZoneInfo

The `"getZoneInfo"` command retrieves detailed information about a single zone.

5.44.1 Request

Table 80. getZoneInfo Request

Parameters	Type	Mandatory	Description
`zone`	String	Yes	Managed zone UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getZoneInfo",
        "params": {
                "zone": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getZoneInfo",
        "params": {
                "zone": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.44.2 Response

Table 81. getZoneInfo Response

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>status</td><td>Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component:
"Enabled",
"Disabled",
"StandbyOffline",
"StandbySpare,"<br>"InTest",
"Starting",
"Absent",
"UnavailableOffline"</td></tr>
<tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource:
"OK", "Warning",
"Critical"</td></tr>
</table>
</td></tr>
<tr><td>name</td><td>String</td><td>No</td><td>Optional name of the component</td></tr>
<tr><td>description</td><td>String</td><td>No</td><td>Optional description of the component</td></tr>
<tr><td>collections</td><td>Array:
Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>name</td><td>String</td><td>Yes</td><td>Collection name</td></tr>
<tr><td>type</td><td>String</td><td>Yes</td><td>Collection type:
"Endpoints"</td></tr>
</table>
</td></tr>
<tr><td>oem</td><td>Object</td><td>No</td><td>OEM specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
```

```
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,"
                                                                        "InTest",
"Starting", "Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"Endpoints"}>
                        }
                        ...
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "collections": [
                        {
                                "name": "Endpoints",
                                "type": "Endpoints"
                        }
                ],
                "oem": {}
        },
        "id": 987
}
```

5.45 addZone

The "addZone" command creates a new zone.

5.45.1 Request

Table 82. addZone Request

Result	Type	Mandatory	Description
fabric	String	Yes	Parent fabric UUID.
endpoints	Array: String	Yes	UUID of endpoints assigned to the zone
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "addZone",
        "params": {
                "fabric": <string>,
```

```
                "endpoints": [
                        <string>,
                        ...
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addZone",
        "params": {
                "fabric": "123e4567-e89b-12d3-a456-426655440000",
                "endpoints": [
                        "863e4567-e87b-64d3-a489-987656540000"
                ],
                "oem": {}
        },
        "id": 987
}
```

5.45.2 Response

Table 83. addZone Response

Parameters	Type	Mandatory	Description
zone	String	Yes	Created zone UUID
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "zone": <string>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "zone": "123e4567-e89b-12d3-a456-426655440000",
                "oem": {}
        },
        "id": 987
}
```

5.46 deleteZone

The "deleteZone" command deletes an existing zone.

5.46.1 Request

Table 84. deleteZone Request

Parameters	Type	Mandatory	Description
zone	String	Yes	Managed zone UUID.

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "deleteZone",
        "params": {
                "zone": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "deleteZone",
        "params": {
                "zone": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.46.2 Response

Table 85. deleteZone Response

Result	Type	Mandatory	Description
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 987
}
```

5.47 addZoneEndpoints

The "addZoneEndpoints" command attaches existing endpoints to a zone.

5.47.1 Request

Table 86. addZoneEndpoints Request

Parameters	Type	Mandatory	Description
zone	String	Yes	Managed zone UUID
endpoints	Array: String	Yes	List of UUIDs of existing endpoints to be attached to this zone.
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "addZoneEndpoints",
        "params": {
                "zone": <string>,
                "endpoints": [
                        <string>,
                        …
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addZoneEndpoints",
        "params": {
                "zone": "123e4567-e89b-12d3-a456-426655440000",
                "endpoints": ["123e4567-e89b-12d3-a456-426655441234"],
                "oem": {}
        },
        "id": 987
}
```

5.47.2 Response

Table 87. addZoneEndpoints Response

Result	Type	Mandatory	Description
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
```

```
        },
        "id": 987
}
```

5.48 deleteZoneEndpoints

The "deleteZoneEndpoints" command detaches existing endpoints from a zone.

5.48.1 Request

Table 88. deleteZoneEndpoints Request

Parameters	Type	Mandatory	Description
zone	String	Yes	Managed zone UUID
endpoints	Array: String	Yes	List of UUIDs of existing endpoints to be removed from this zone.
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "deleteZoneEndpoints",
        "params": {
                "zone": <string>,
                "endpoints": [
                        <string>,
                        …
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "deleteZoneEndpoints",
        "params": {
                "zone": "123e4567-e89b-12d3-a456-426655440000",
                "endpoints": ["123e4567-e89b-12d3-a456-426655441234"],
                "oem": {}
        },
        "id": 987
}
```

5.48.2 Response

Table 89. deleteZoneEndpoints Response

Result	Type	Mandatory	Description
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
```

```
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 987
}
```

5.49 getEthernetSwitchInfo

The "getEthernetSwitchInfo" command retrieves detailed information about a single Ethernet switch.

5.49.1 Request

Table 90. getEthernetSwitchInfo Request

Parameters	Type	Mandatory	Description
switch	String	Yes	Managed Ethernet switch UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getEthernetSwitchInfo",
        "params": {
                "switch": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getEthernetSwitchInfo",
        "params": {
                "switch": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.49.2 Response

Table 91. getEthernetSwitchInfo Response

Result	Type	Mandatory	Description
status	Object	Yes	

Attribute	Type	Mandatory	Description
state	String	Yes	Known state of the component:

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td></td><td></td><td></td><td>"Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr>
<tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource:
"OK", "Warning", "Critical"</td></tr>
</table>
</td></tr>
<tr><td>name</td><td>String</td><td>No</td><td>Optional name of the component</td></tr>
<tr><td>description</td><td>String</td><td>No</td><td>Optional description of the component</td></tr>
<tr><td>switchIdentifier</td><td>String</td><td>Yes</td><td>Switch identifier</td></tr>
<tr><td>technology</td><td>String</td><td>No</td><td>Switch technology:
"Ethernet", "PCIe"</td></tr>
<tr><td>macAddress</td><td>String</td><td>Yes</td><td>Switch MAC Address</td></tr>
<tr><td>firmwareName</td><td>String</td><td>No</td><td>Name of switch FW</td></tr>
<tr><td>firmwareVersion</td><td>String</td><td>No</td><td>The version of switch FW</td></tr>
<tr><td>role</td><td>String</td><td>No</td><td>Switch role in the network:
"TOR", "EOR", "Drawer", "Unknown"</td></tr>
<tr><td>fruInfo</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>serialNumber</td><td>String</td><td>No</td><td>Module serial number</td></tr>
<tr><td>manufacturer</td><td>String</td><td>No</td><td>Manufacturer name</td></tr>
<tr><td>modelNumber</td><td>String</td><td>No</td><td>Model number</td></tr>
<tr><td>partNumber</td><td>String</td><td>No</td><td>Part number</td></tr>
</table>
</td></tr>
<tr><td>manufacturingDate</td><td>String</td><td>No</td><td>Manufacturing date</td></tr>
<tr><td>location</td><td>Number</td><td>Yes</td><td>Switch location within the chassis (aka. socket)</td></tr>
<tr><td>chassis</td><td>String</td><td>Yes</td><td>UUID of the chassis containing the Ethernet switch</td></tr>
<tr><td>maxAclNumber</td><td>Number</td><td>No</td><td>The maximum number of ACLs that can be handled by the switch</td></tr>
<tr><td>lldpEnabled</td><td>Boolean</td><td>No</td><td>Indicates whether Link Layer Discovery Protocol (LLDP) IEEE 802.1AB is enabled on this switch</td></tr>
<tr><td>pfcEnabled</td><td>Boolean</td><td>No</td><td>Indicates whether Priority Flow Control (PFC) defined in IEEE 802.1Qbb is enabled on this switch</td></tr>
<tr><td>etsEnabled</td><td>Boolean</td><td>No</td><td>Indicates whether Enhanced Transmission Selection (ETS) defined in IEEE 802.1Qaz is enabled on this switch</td></tr>
<tr><td>dcbxEnabled</td><td>Boolean</td><td>No</td><td>Indicates whether Data Center Bridging Extensions (DCBX) is enabled on this switch</td></tr>
<tr><td>qosApplicationProtocol</td><td>Array: Object</td><td>Yes</td><td>Properties which specify the configuration of application protocol of DCB node
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>protocol</td><td>String</td><td>No</td><td>Protocol supported by DCB node: "TCP", "UDP", "L2"</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>port</td><td>Number</td><td>No</td><td>UDP/TCP socket number or EtherType for protocol set to L2. The maximum value of this field is 65535</td></tr>
<tr><td>priority</td><td>Number</td><td>No</td><td>Supported values: 0, 1, 2, 3, 4, 5, 6, 7</td></tr>
</table>
</td></tr>
<tr><td>qosPriorityToPriorityGroupMapping</td><td>Array: Object</td><td>Yes</td><td>Provides configuration of priority to priority group mapping
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>priorityGroup</td><td>Number</td><td>No</td><td>Priority group ID. Supported range [0, 15]</td></tr>
<tr><td>priority</td><td>Number</td><td>No</td><td>Supported values: 0, 1, 2, 3, 4, 5, 6, 7</td></tr>
</table>
</td></tr>
<tr><td>qosBandwidthAllocation</td><td>Array: Object</td><td>Yes</td><td>Provides configuration of bandwidth allocation on converged links in end stations and switches in a DCB (Data Center Bridging) environment
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>priorityGroup</td><td>Number</td><td>No</td><td>Priority group ID. Supported range [0, 15]</td></tr>
<tr><td>bandwidthPercent</td><td>Number</td><td>No</td><td>Percentage of guaranteed bandwidth. Supported range [0, 100]</td></tr>
</table>
</td></tr>
<tr><td>collections</td><td>Array: Object</td><td>Yes</td><td>Subcomponents collections
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>name</td><td>String</td><td>Yes</td><td>Collection name</td></tr>
<tr><td>type</td><td>String</td><td>Yes</td><td>Collection type: "EthernetSwitchPorts", "Acls", "NeighborSwitches", "VxLANs"</td></tr>
</table>
</td></tr>
<tr><td>services</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>ntp</td><td>Object</td><td>No</td><td>Network Time Protocol
<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>enabled</td><td>Boolean</td><td>Is enabled</td></tr>
<tr><td>pollInterval</td><td>Number</td><td>Interval in secs</td></tr>
<tr><td>timeServers</td><td>Array: Object</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>name</td><td>String</td><td>User-defined name</td></tr>
<tr><td>address</td><td>String</td><td>Hostname</td></tr>
<tr><td>port</td><td>Number</td><td>Service port number</td></tr>
<tr><td>preferred</td><td>Boolean</td><td>Is preferred</td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th colspan="4">Description</th></tr>
<tr><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td><td>lldp</td><td>Object</td><td>No</td><td>Link Layer Discovery Protocol
<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>enabled</td><td>Boolean</td><td>Is enabled</td></tr>
<tr><td>holdTimeSeconds</td><td>Number</td><td>Hold time in seconds</td></tr>
<tr><td>intervalSeconds</td><td>Number</td><td>Interval in seconds</td></tr>
</table></td></tr>
<tr><td>stp</td><td>Object</td><td>No</td><td>Spanning Tree Protocol
<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>enabled</td><td>Boolean</td><td>Is enabled</td></tr>
</table></td></tr>
<tr><td>oem</td><td>Object</td><td>No</td><td colspan="4">OEM-specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "switchIdentifier": <string>
                "technology": <string>,
                "macAddress": <string>,
                "firmwareName": <string>,
                "firmwareVersion": <string>,
                "role": <{"TOR", "EOR", "Drawer", Unknown"}>,
                "fruInfo": {
                        "serialNumber": <string>,
                        "manufacturer": <string>,
                        "modelNumber": <string>,
                        "partNumber": <string>
                },
                "manufacturingDate": <string>,
                "location": <number>,
                "chassis": <string>,
                "qosApplicationProtocol": [
                        {
                                "protocol": <{"TCP", "UDP", "L2"}>,
                                "port": <number>,
                                "priority": <number>
                        },
                        ...
                ],
                "qosPriorityToPriorityGroupMapping": [
                        {
                                "priority": <number>,
                                "priorityGroup": <number>
                        },
                        ...
                ],
                "qosBandwidthAllocation": [
                        {
                                "bandwidthPercent": <number>,
                                "priorityGroup": <number>
```

```
                        },
                        ...
                ],
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"EthernetSwitchPorts", "Acls",
"NeighborSwitches"}>
                        }
                        ...
                ],
                "services": {
                        "ntp": {
                                "enabled": <Boolean>,
                                "pollInterval": <Number>,
                                "timeServers": [
                                                        {
                                                                "name": <String>,
                                                                "address": <String>",
                                                                "port": <Number>,
                                                                "preferred": <Boolean>
                                                        },
                                                        ...
                                ]
                        },
                        "lldp": {
                                "enabled": <Boolean>,
                                "holdTimeSeconds": <Number>,
                                "intervalSeconds": <Number>
                        },
                        "stp": {
                                "enabled": <Boolean>
                        }
                },
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "switchIdentifier": "Switch 3"
                "technology": "Ethernet",
                "macAddress": "AA:BB:CC:DD:EE:FF",
                "firmwareName": "FW file name",
                "firmwareVersion": "1.1.0.2341",
                "role": "TOR",
                "fruInfo": {
                        "serialNumber": "123fed3029c-b23394-12",
                        "manufacturer": "Intel Corporation",
                        "modelNumber": "E323",
                        "partNumber": "29ee2220939"
                },
                "manufacturingDate": "02/21/2015 00:00:00",
```


```
                "location": 1,
                "chassis": "123e4567-e89b-12d3-a456-426655440000",
                "qosApplicationProtocol": [ ],
                "qosPriorityToPriorityGroupMapping": [ ],
                "qosBandwidthAllocation": [ ],
                "collections": [
                        {
                                "name": "EthernetSwitchPorts",
                                "type": "EthernetSwitchPorts"
                        },
                        {
                                "name": "Neighbors",
                                "type": "NeighborSwitches"
                        },
                        {
                                "name": "Acls",
                                "type": "Acls"
                        }
                ],
                "services": {
                        "ntp": {
                                "enabled": true,
                                "pollInterval": 10,
                                "timeServers": [
                                                {
                                                        "name": "Intel NTP",
                                                        "address":
"ntp.intel.com",
                                                        "port": 123,
                                                        "preferred": true
                                                },
                                                {
                                                        "name": "PODM NTP",
                                                        "address": "10.2.21.1",
                                                        "port": 123,
                                                        "preferred": false
                                                }
                                ]
                        },
                        "lldp": {
                                "enabled": true,
                                "holdTimeSeconds": 120,
                                "intervalSeconds": 30
                        },
                        "stp": {
                                "enabled": false
                        }
                },
                "oem": {}
        },
        "id": 987
}
```


5.50 Set Ethernet Switch Attributes

The `"setComponentAttributes"` described in Section 5.5, setComponentAttributes, allows configuration of the Ethernet Switch Attributes listed in the following table.

Table 92. Configurable Ethernet Switch Attributes

Attribute	Type	Description
services	Object	Refer to Section 5.49, getEthernetSwitchInfo
lldpEnabled	Boolean	Indicates whether Link Layer Discovery Protocol (LLDP) IEEE 802.1AB is enabled on this switch
pfcEnabled	Boolean	Indicates whether the Priority Flow Control (PFC) defined in IEEE 802.1Qbb is enabled on this switch.
etsEnabled	Boolean	Indicates whether the Enhanced Transmission Selection (ETS) defined in IEEE 802.1Qaz is enabled on this switch.
dcbxEnabled	Boolean	Indicates whether the Data Center Bridging Extensions (DCBX) is enabled on this switch.
qosApplicationProtocol	Array: Object	Properties which specify a configuration of the application protocol of the Data Center Bridging (DCB) node. For more details refer to Section 5.49, getEthernetSwitchInfo.
qosPriorityToPriorityGroupMapping	Array: Object	Provides configuration of priority to priority group mapping. For more details refer to Section 5.49, getEthernetSwitchInfo.
qosBandwidthAllocation	Array: Object	Provides configuration of bandwidth allocation on converged links in end stations and Switches in a DCB environment. For more details refer to Section 5.49, getEthernetSwitchInfo.
oem	Object	OEM-specific data

5.51 getEthernetSwitchPortInfo

The "getEthernetSwitchPortInfo" command retrieves detailed information about a Single Switch Port.

5.51.1 Request

Table 93. getEthernetSwitchPortInfo Request

Parameters	Type	Mandatory	Description
port	String	Yes	Managed switch port UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getEthernetSwitchPortInfo",
        "params": {
                "port": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getEthernetSwitchPortInfo",
        "params": {
                "port": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.51.2 Response

Table 94. getEthernetSwitchPortInfo Response

Result	Type	Mandatory	Description
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline "</td></tr><tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
name	String	No	Optional name of the component
description	String	No	Optional description of the component
portIdentifier	String	Yes	Port Identifier (port index, name, etc.)
portClass	String	Yes	Port class: "Physical", "Logical", "Reserved"
portType	String	No	Port type: "Upstream", "Downstream", "MeshPort", "Unknown"
portMode	String	No	Port working mode. The value shall correspond to the port class (especially to the logical port definition). General: "Unknown" Logical Link Aggregation port: "LinkAggregationStatic", "LinkAggregationDynamic"
linkTechnology	String	No	This is link technology, such as Ethernet, for this NIC: "Ethernet", "PCIe" "Unknown"
linkSpeedMbps	Number	Yes	Current port speed
maxSpeedMbps	Number	No	Max port speed
operationalState	String	Yes	Operational (runtime) port state: "Up", "Down", "Unknown"
administrativeState	String	Yes	Port administrative state set by the operator: "Up", "Down"
portWidth	Number	No	Port width, for PCIe port this is number of PCIe lanes
frameSize	Number	No	MAC Frame size in bytes
autoSense	Boolean	No	Indicates if the speed and duplex is automatically configured by the port
fullDuplex	Boolean	No	Indicates if a port is the full-duplex switch port
isManagementPort	Boolean	No	Indicates if a port may be used for switch management
lastErrorCode	Number	No	Code of last error detected
errorCleared	Boolean	No	Indicates whether an error has been cleared
lastStateChangeTime	String	No	Time of the last port state change

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>macAddress</td><td>String</td><td>No</td><td>Switch port MAC address: "AA:BB:CC:DD:EE:FF"</td></tr>
<tr><td>ipv4Address</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>address</td><td>String</td><td>Yes</td><td>IPv4 address</td></tr>
<tr><td>subnetMask</td><td>String</td><td>No</td><td>IPv4 subnet mask</td></tr>
<tr><td>addressOrigin</td><td>String</td><td>No</td><td>IPv4 address origin "DHCP", "Static"</td></tr>
<tr><td>gateway</td><td>String</td><td>No</td><td>IPv4 gateway for this address</td></tr>
</table>
</td></tr>
<tr><td>ipv6Address</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>address</td><td>String</td><td>Yes</td><td>IPv6 address</td></tr>
<tr><td>prefixLength</td><td>Number</td><td>No</td><td>IPv6 Address Prefix Length</td></tr>
<tr><td>addressOrigin</td><td>String</td><td>No</td><td>IPv4 address origin "DHCP", "Static", "SLAAC"</td></tr>
<tr><td>addressState</td><td>String</td><td>No</td><td>IPv6 address state: "Preferred", "Deprecated", "Tentative", "Failed"</td></tr>
</table>
</td></tr>
<tr><td>neighborInfo</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>switchIdentifier</td><td>String</td><td>No</td><td>Remote switch identifier</td></tr>
<tr><td>portIdentifier</td><td>String</td><td>No</td><td>Port identifier on the remote switch which is connected to a given local switch port</td></tr>
<tr><td>cableId</td><td>String</td><td>No</td><td>Cable identifier</td></tr>
</table>
</td></tr>
<tr><td>neighborMac</td><td>String</td><td>No</td><td>MAC address of the remote port/interface</td></tr>
<tr><td>vlanEnable</td><td>Boolean</td><td>Yes</td><td>Indicates if VLANs are enabled on the switch port</td></tr>
<tr><td>defaultVlan</td><td>String</td><td>No</td><td>Default VLAN UUID</td></tr>
<tr><td>collections</td><td>Array: Object</td><td>Yes</td><td>Subcomponents collections
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>name</td><td>String</td><td>Yes</td><td>Collection name</td></tr>
<tr><td>type</td><td>String</td><td>Yes</td><td>Collection type: "EthernetSwitchPortVlans", "EthernetSwitchPortMembers", "Acls", "StaticMacs"</td></tr>
</table>
</td></tr>
</table>

Result	Type	Mandatory	Description
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "portIdentifier": <string>,
                "portClass": <{"Physical", "Logical", "Reserved"}>
                "portType": <{"Upstream", "Downstream", "MeshPort", "Unknown"}>,
                "linkTechnology": <{"Ethernet", "pCIe", "Unknown"}>,
                "linkSpeedMbps": <number>,
                "maxSpeedMbps": <number>,
                "operationalState": <{"Up", "Down", "Unknown"}>,
                "administrativeState": <{"Up", "Down"}>,
                "portWidth": <number>,
                "frameSize": <number>,
                "autosense": <boolean>,
                "fullDuplex": <boolean>,
                "isManagementPort": <boolean>,
                "lastErrorCode": <number>,
                "errorCleared": <boolean>,
                "lastStateChangeTime": <string>,
                "macAddress": <string>,
                "ipv4Address": {
                        "address": <string>,
                        "subnetMask": <string>,
                        "addressOrigin": <{"DHCP","Static"}>
                        "gateway": <string>
                },
                "ipv6Address": {
                        "address": <string>,
                        "prefixLength": <number>,
                        "addressOrigin": <{"DHCP","Static","SLAAC"}>
                        "addressState":
<{"Preferred","Depracated","Tentative","Failed"}>
                },
                "neighborInfo": {
                        "switchIdentifier": <string>,
                        "portIdentifier": <string>,
                        "cableId": <string>
                },
                "neighborMac": <string>,
                "vlanEnable": <boolean>,
                "pfcEnabledPriorities": [
                        <number>,
                        ...
                ],
                "defaultVlan": <string>,
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"EthernetSwitchPortVlans",
"EthernetSwitchPortMembers", "Acls", "StaticMacs"}>
                        }
```

```
                        ...
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "portIdentifier": "1",
                "portClass": "Physical",
                "portType": "Downstream",
                "linkTechnology": "Ethernet",
                "linkSpeedMbps": 1000,
                "maxSpeedMbps": 10000,
                "operationalState": "Up",
                "administrativeState": "Up",
                "portWidth": 1,
                "frameSize": 1520,
                "autosense": true,
                "fullDuplex": true,
                "isManagementPort": false,
                "lastErrorCode": 0,
                "errorCleared": false,
                "lastStateChangeTime": "2015-02-23T14:44:00+00:00",
                "macAddress": "AA:BB:CC:DD:EE:FF",
                "ipv4Address": {
                        "address": "10.0.2.10",
                        "subnetMask": "255.255.255.0",
                        "addressOrigin": "DHCP",
                        "gateway": "10.0.2.1"
                },
                "ipv6Address": {
                        "address": "fe80::1ec1:deff:fe6f:1c37",
                        "prefixLength": 16,
                        "addressOrigin": "DHCP",
                        "addressState": "Preferred"
                },
                "neighborInfo": {
                        "switchIdentifier": "123e4567-e89b-12d3-a456-426655440000",
                        "portIdentifier": "19",
                        "cableId": "TOR port 19"
                },
                "neighborMac": "12:34:56:78:90:AB",
                "vlanEnable": true,
                "pfcEnabledPriorities": [ ],
                "defaultVlan": "cd3e4527-af7b-32d3-a489-987656a3d588",
                "collections": [
                        {
                                "name": "EthernetSwtichPortVlans",
                                "type": "EthernetSwtichPortVlans"
                        },
                        {
                                "name": "Acls",
```

```
                                    "type": "Acls"
                            },
                            {
                                    "name": "StaticMacs",
                                    "type": "StaticMacs"
                            }
                    ],
                    "oem": {}
            },
            "id": 987
}
```

5.52 Set Ethernet Switch Port Attributes

The "setComponentAttributes" described in Section 5.5, setComponentAttributes, allows configuration of the Ethernet Switch Port Attributes listed in the following table.

Table 95. Configurable Ethernet Switch Port Attributes

Attribute	Type	Description
linkSpeedMbps	Number	The switch port speed in Mbps
administrativeState	String	Port administrative state: "Up", "Down"
frameSize	Number	MAC frame size in bytes
autoSense	Boolean	Indicates if the port automatically configures the speed and duplex
mode	String	Port working mode. The value shall correspond to the port class (especially to the logical port definition). Logical Link Aggregation port: "LinkAggregationStatic", "LinkAggregationDynamic"
vlanEnable	Boolean	Indicates if VLANs are enabled on the switch port
defaultVlan	String	Default VLAN UUID
lldpEnabled	Boolean	Indicates whether LLDP IEEE 802.1AB is enabled on this switch port
pfcEnabled	Boolean	Indicates whether PFC defined in IEEE 802.1Qbb is enabled on this switch port
pfcEnabledPriorities	Array: Number	List of priorities that should be treated by the switch as lossless. Available values: 0, 1, 2, 3, 4, 5, 6, 7
dcbxState	String	Indicates whether DCBX is enabled on this switch port: "Disabled", "EnabledIEEE", "EnabledCEE"
oem	Object	OEM-specific data

5.53 addEthernetSwitchPort

The "addEthernetSwitchPort" command creates a new logical port on the switch.

5.53.1 Request

Table 96. addEthernetSwitchPort Request

Parameters	Type	Mandatory	Description
switch	String	Yes	Managed switch UUID
portIdentifier	String	Yes	Port identifier
mode	string	Yes	Port working mode. The value shall correspond to the port class (especially to the logical port definition). Logical Link Aggregation port: "LinkAggregationStatic", "LinkAggregationDynamic"

Parameters	Type	Mandatory	Description
members	Array: String	Yes	The UUIDs of the switch ports (physical or logical) that should be used to create the new logical switch port
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "addEthernetSwitchPort",
        "params": {
                "switch": <string>,
                "portIdentifier": <string>,
                "mode": <string>,
                "members": [
                        <string>,
                        ...
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addEthernetSwitchPort",
        "params": {
                "switch": "123e4567-e89b-12d3-a456-426655440000",
                "portIdentifier": "UplikAggregation1",
                "mode": "LinkAggregationStatic",
                "members": [
                        "cd3e4527-af7b-32d3-a489-987656a3d588",
                        "863e4567-e87b-64d3-a489-987656540000"
                ],
                "oem": {}
        },
        "id": 987
}
```

5.53.2 Response

Table 97. addEthernetSwitchPort Response

Result	Type	Mandatory	Description
port	String	Yes	Created port UUID
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "port": <string>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "port": "123e4567-e89b-12d3-a456-426655440000",
                "oem": {}
        },
        "id": 987
}
```

5.54 deleteEthernetSwitchPort

The "deleteEthernetSwitchPort" command destroys an existing logical port on the switch.

5.54.1 Request

Table 98. deleteEthernetSwitchPort Request

Parameters	Type	Mandatory	Description
port	String	Yes	Managed port UUID
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "deleteEthernetSwitchPort",
        "params": {
                "port": <string>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "deleteEthernetSwitchPort",
        "params": {
                "port": "123e4567-e89b-12d3-a456-426655440000",
                "oem": {}
        },
        "id": 987
}
```

5.54.2 Response

Table 99. deleteEthernetSwitchPort Response

Result	Type	Mandatory	Description
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
        },
```

```
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 987
}
```

5.55 addEthernetSwitchPortMembers

The "addEthernetSwitchPortMembers" command adds a new member(s) to the existing logical port on the switch.

5.55.1 Request

Table 100. addEthernetSwitchPortMembers Request

Parameters	Type	Mandatory	Description
port	String	Yes	Managed switch port UUID
members	Array: String	Yes	The UUIDs of the switch ports (physical or logical) that should be added to the existing logical switch port
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "addEthernetSwitchPortMembers",
        "params": {
                "port": <string>,
                "members": [
                        <string>,
                        ...
                ]
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addEthernetSwitchPortMembers",
        "params": {
                "port": "123e4567-e89b-12d3-a456-426655440000",
                "members": [
                        "863e4567-e87b-64d3-a489-987656540000"
                ],
                "oem": {}
        },
        "id": 987
}
```

5.55.2 Response

Table 101 addEthernetSwitchPortMembers Response

Result	Type	Mandatory	Description
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 987
}
```

5.56 deleteEthernetSwitchPortMembers

The "deleteEthernetSwitchPortMembers" command removes member(s) from the existing Logical Port on the Switch.

5.56.1 Request

Table 102. deleteEthernetSwitchPortMembers Request

Parameters	Type	Mandatory	Description
port	String	Yes	Managed switch port UUID
members	Array: String	Yes	The UUIDs of the switch ports being members of the existing logical port that should be removed
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "deleteEthernetSwitchPortMembers",
        "params": {
                "port": <string>,
                "members": [
                        <string>,
                        ...
                ]
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "deleteEthernetSwitchPortMembers",
        "params": {
                "port": "123e4567-e89b-12d3-a456-426655440000",
                "members": [
                        "863e4567-e87b-64d3-a489-987656540000"
                ],
                "oem": {}
        },
        "id": 987
}
```

5.56.2 Response

Table 103. deleteEthernetSwitchPortMembers Response

Result	Type	Mandatory	Description
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 987
}
```

5.57 getEthernetSwitchVxlanInfo

The "getEthernetSwitchVxlanInfo" command retrieves detailed information about the VxLAN.

5.57.1 Request

Table 104. getEthernetSwitchVxlanInfo Request

Parameters	Type	Mandatory	Description
vxlan	String	Yes	Switch VxLAN UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getEthernetSwitchVxlanInfo",
        "params": {
                "vxlan": <string>
        },
```

```
        "id": <id>
}
```

Example:

```
{
"jsonrpc": "2.0",
"method": "getEthernetSwitchVxlanInfo",
"params": {
"vxlan": "123e4567-e89b-12d3-a456-890319000000"
},
"id": 987
}
```

5.57.2 Response

Table 105. getEthernetSwitchVxlanInfo Response

Result	Type	Mandatory	Description
name	String	No	Optional name of the component
description	String	No	Optional description of the component
vxLanId	Number	Yes	Identification of the VxLAN
vlanIds	Array of Numbers	Yes	VLAN numbers used
macs	Array: String	Yes	The array of MAC addresses
vlanPorts	Array: String	Yes	The array of switch port UUIDs.
discardInnerVLanTag	Boolean	Yes	Boolean variable if discard inner VLAN tag
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getEthernetSwitchVxlanInfo",
        "result": {
                "vxLanId": <Number>,
                "vlanIds": <Array of Numbers>,
                "macs": <Array of Strings>,
                "vlanPorts": <Array of Stings>,
                "discardInnerVlanTag: <Boolean>,
                "oem": <Object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addEthernetSwitchVxlan",
        "result": {
                "switch": "988be992-8685-4aa4-ac3a-7d97dc75c65f",
                "vxLanId": 777,
                "vlanIds": [
                        100,
                        200
                ],
                "macs": [
                        "00:11:22:33:44:55",
```

```
                        "AB:CD:EF:DE:AD:00"
                ],
                "vlanPorts": [
                        "fb4aa5be-6fc1-40d1-8ef8-6abc246775f8",
                        "86eda6d7-6fa4-49d5-be21-59dca1e5cadf"
                ],
                "discardInnerVlanTag: true,
                "oem": {}
        },
        "id": 1
}
```

5.58 addEthernetSwitchVxlan

The "addEthernetSwitchVxlan" enables VxLAN creation on the Ethernet Switch.

5.58.1 Request

Table 106. addEthernetSwitchVxlan Request

Parameters	Type	Mandatory	Description
switch	String	Yes	Switch UUID
vxLanId	Number	Yes	Identification of VxLAN to create
vlanIds	Array: Number	Yes	Array of VLANs
macs	Array: String	Yes	The array of MAC addresses
vlanPorts	Array: String	Yes	Identifiers of the VLAN ports
discardInnerVlanTag	Boolean	Yes	Boolean variable whether to discard inner VLAN tag or not
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "addEthernetSwitchVxlan",
        "params": {
                "switch": <String>,
                "vxLanId": <Number>,
                "vlanIds": <Array of Numbers>,
                "macs": <Array of Strings>,
                "vlanPorts": <Array of Stings>,
                "discardInnerVlanTag: <Boolean>,
                "oem": <Object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addEthernetSwitchVxlan",
        "params": {
                "switch": "988be992-8685-4aa4-ac3a-7d97dc75c65f",
                "vxLanId": 777,
                "vlanIds": [
                        100,
```

```
                        200
                ],
                "macs": [
                        "00:11:22:33:44:55",
                        "AB:CD:EF:DE:AD:00"
                ],
                "vlanPorts": [
                        "fb4aa5be-6fc1-40d1-8ef8-6abc246775f8",
                        "86eda6d7-6fa4-49d5-be21-59dca1e5cadf"
                ],
                "discardInnerVlanTag: true,
                "oem": {}
        },
        "id": 1
}
```

5.58.2 Response

Table 107. addEthernetSwitchVxlan Response

Result	Type	Mandatory	Description
vxlan	String	Yes	Switch VxLAN UUID
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "vxlan": <string>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "vxlan": "13821bfd-5431-4dc9-b9d5-a6555a8ae3c3",
                "oem": {}
        },
        "id": 1
}
```

5.59 deleteEthernetSwitchVxlan

The "deleteEthernetSwitchVxlan" command enables VxLAN deletion from the Ethernet Switch.

5.59.1 Request

Table 108. deleteEthernetSwitchVxlan Request

Parameters	Type	Mandatory	Description
switch	String	Yes	Switch UUID
vxlan	String	Yes	Switch VxLAN UUID
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "deleteEthernetSwitchVxlan",
        "params": {
                "switch": <String>,
                "vxlan": <String>,
                "oem": <Object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "deleteEthernetSwitchVxlan",
        "params": {
                "switch": "853887b6-a174-4b83-a848-b748f289e3a2",
                "vxlan": "d1ec3a02-d7f0-4b3d-9fa3-bbecce3fd981",
                "oem": {}
        },
        "id": 2
}
```

5.59.2 Response

Table 109. deleteEthernetSwitchVxlan Response

Result	Type	Mandatory	Description
oem	String	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 1
}
```

5.60 getRemoteEthernetSwitchInfo

The "getRemoteEthernetSwitchInfo" command retrieves detailed information about a Single Switch in the distributed network topology.

5.60.1 Request

Table 110. getRemoteEthernetSwitchInfo Request

Parameters	Type	Mandatory	Description
switch	String	Yes	Remote switch UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getRemoteEthernetSwitchInfo",
        "params": {
                 "switch": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getRemoteEthernetSwitchInfo",
        "params": {
                 "switch": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.60.2 Response

Table 111. getRemoteEthernetSwitchInfo Response

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>status</td><td>Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component:
"Enabled",
"Disabled",
"StandbyOffline",
"StandbySpare,"<br>"InTest",
"Starting",
"Absent",
"UnavailableOffline"</td></tr>
<tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource:
"OK", "Warning",
"Critical"</td></tr>
</table>
</td></tr>
<tr><td>name</td><td>String</td><td>No</td><td>Optional name of the component</td></tr>
<tr><td>description</td><td>String</td><td>No</td><td>Optional description of the component</td></tr>
<tr><td>switchIdentifier</td><td>String</td><td>Yes</td><td>Switch identifier</td></tr>
<tr><td>macAddress</td><td>String</td><td>Yes</td><td>Switch MAC Address</td></tr>
<tr><td>nextHop</td><td>Array: Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
</table>
</td></tr>
</table>

Result	Type	Mandatory	Description
			<table><tr><td>metric</td><td>Number</td><td>Yes</td><td>The cost associated with a given path to the remote switch</td></tr><tr><td>portIdentifier</td><td>String</td><td>Yes</td><td>Identifier of a port through which a given remote switch is accessible</td></tr><tr><td>ipv4Address</td><td>String</td><td>No</td><td>IPv4 address through which a given remote switch is accessible</td></tr><tr><td>ipv6Address</td><td>String</td><td>No</td><td>IPv6 address through which a given remote switch is accessible</td></tr></table>
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "switchIdentifier": <string>
                "macAddress": <string>,
                "nextHop": [{
                        "metric": <number>,
                        "portIdentifier": <string>,
                        "ipv4Address": <string>,
                        "ipv6Address": <string>
                        },
                ...
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "switchIdentifier": "Switch 3"
                "macAddress": "AA:BB:CC:DD:EE:FF",
                "nextHop": [{
                        "metric": 128,
                        "portIdentifier": "16",
                        "ipv4Address": "10.0.2.10"
                }],
                "oem": {}
```

```
        },
        "id": 987
}
```

5.61 getVlanInfo

The "getVlanInfo" command retrieves detailed information about a single VLAN configured within the Switch.

Note: This command is optional and may not be implemented if the VLANs are not explicitly created on the Switch–they are created indirectly during the creation of the port VLAN.

5.61.1 Request

Table 112. getVlanInfo Request

Parameters	Type	Mandatory	Description
vlan	String	Yes	Managed VLAN UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getVlanInfo",
        "params": {
                "vlan": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getVlanInfo",
        "params": {
                "vlan": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.61.2 Response

Table 113. getVlanInfo Request

Result	Type	Mandatory	Description
status	Object	Yes	See below

Attribute	Type	Mandatory	Description
state	String	Yes	Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"

Result	Type	Mandatory	Description
			`health` | String | Yes | Overall health state from the view of this resource: `"OK", "Warning", "Critical"`
`name`	String	No	Optional name of the component
`description`	String	No	Optional description of the component
`vlanId`	Number	Yes	VLAN identifier
`vlanName`	String	No	VLAN name
`vlanEnable`	Boolean	Yes	Indicates if VLAN is enabled
`oem`	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "vlanId": <number>,
                "vlanName": <string>,
                "vlanEnable": <boolean>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "vlanId": 100,
                "vlanName": "Management VLAN",
                "vlanEnable": true,
                "oem": {}
        },
        "id": 987
}
```

5.62 Set VLAN Attributes

The `"setComponentAttributes"` described in Section 5.5, setComponentAttributes, allows configuration of the VLAN Attributes listed in the following table.

Table 114. Configurable VLAN Attributes

Attribute	Type	Description
`vlanName`	String	VLAN name

vlanEnable	Boolean	VLAN state (true=enabled, false=disabled)
oem	Object	OEM specific data

5.63 addVlan

The "addVlan" command creates a new VLAN on the Switch.

Note: This command is optional and may not be implemented if VLANs are not explicitly created on the Switch–they are created indirectly during the creation of the port VLAN.

5.63.1 Request

Table 115. addVlan Request

Parameters	Type	Mandatory	Description
switch	String	Yes	Managed switch UUID
vlanId	Number	Yes	New VALN Identifier
vlanName	String	Yes	New VLAN name

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "addVlan",
        "params": {
                "switch": <string>,
                "vlanId": <number>,
                "vlanName": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addVlan",
        "params": {
                "switch": "123e4567-e89b-12d3-a456-426655440000",
                "vlanId": 102,
                "vlanName": "Management VLAN"
        },
        "id": 987
}
```

5.63.2 Response

Table 116. addVlan Response

Result	Type	Mandatory	Description
vlan	String	Yes	Created VLAN UUID
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
```

```
                "vlan": <string>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "vlan": "123e4567-e89b-12d3-a456-426655440000",
                "oem": {}
        },
        "id": 987
}
```

5.64 deleteVlan

The `"deleteVlan"` command destroys an existing VLAN on the Switch.

Note: This command is optional and may not be implemented if VLANs are not explicitly created on the Switch–they are created indirectly during the creation of the port VLAN.

5.64.1 Request

Table 117. deleteVlan Request

Parameters	Type	Mandatory	Description
`vlan`	String	Yes	Managed VLAN UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "deleteVlan",
        "params": {
                "vlan": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "deleteVlan",
        "params": {
                "vlan": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.64.2 Response

Table 118. deleteVlan Response

Result	Type	Mandatory	Description
`oem`	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 987
}
```

5.65 getPortVlanInfo

The "getPortVlanInfo" Info command retrieves detailed information about a single VLAN configured on the switch port.

5.65.1 Request

Table 119. getPortVlanInfo Request

Parameters	Type	Mandatory	Description
portVlan	String	Yes	Managed port VLAN UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getPortVlanInfo",
        "params": {
                "portVlan": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getPortVlanInfo",
        "params": {
                "portVlan": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.65.2 Response

Table 120. getPortVlanInfo Response

Result	Type	Mandatory	Description
status	Object	Yes	

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component:
"Enabled",
"Disabled",
"StandbyOffline",
"StandbySpare,"<br>"InTest",
"Starting",
"Absent",
"UnavailableOfflin e"</td></tr>
<tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource:
"OK", "Warning", "Critical"</td></tr>
</table>
</td></tr>
<tr><td>name</td><td>String</td><td>No</td><td>Optional name of the component</td></tr>
<tr><td>description</td><td>String</td><td>No</td><td>Optional description of the component</td></tr>
<tr><td>vlanId</td><td>Number</td><td>Yes</td><td>VLAN identifier for this port</td></tr>
<tr><td>vlanName</td><td>String</td><td>No</td><td>VLAN name</td></tr>
<tr><td>vlanEnable</td><td>Boolean</td><td>Yes</td><td>Indicates if VLAN is enabled</td></tr>
<tr><td>tagged</td><td>Boolean</td><td>Yes</td><td>Indicates if it is tagged VLAN (if not it is port-based VLAN)</td></tr>
<tr><td>oem</td><td>Object</td><td>No</td><td>OEM specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "vlanId": <number>,
                "vlanName": <string>,
                "vlanEnable": <boolean>,
                "tagged": <boolean>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "vlanId": 100,
                "vlanName": "Management VLAN",
```

```
                "vlanEnable": true,
                "tagged": true,
                "oem": {}
        },
        "id": 987
}
```

5.66 Set Port VLAN Attributes

The "setComponentAttributes" described in Section 5.5, setComponentAttributes, allows configuration of the Port VLAN Attributes listed in the following table.

Table 121. Configurable Port VLAN Attributes

Attribute	Type	Description
vlanName	String	VLAN name
vlanEnable	Boolean	VLAN state (true=enabled, false=disabled)
vlanId	Number	VLAN id
oem	Object	OEM specific data

5.67 addPortVlan

The "addPortVlan" command creates a new VLAN on the Switch Port.

5.67.1 Request

Table 122. addPortVlan Request

Parameters	Type	Mandatory	Description
port	String	Yes	Managed switch port UUID
vlanId	Number	Yes	New VALN Identifier
vlanName	String	No	New VLAN name
tagged	Boolean	Yes	Create tagged VLAN
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "addPortVlan",
        "params": {
                "port": <string>,
                "vlanId": <number>,
                "vlanName": <string>,
                "tagged": <boolean>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addPortVlan",
        "params": {
                "port": "123e4567-e89b-12d3-a456-426655440000",
```

```
                "vlanId": 102,
                "vlanName": null,
                "tagged": true,
                "oem": {}
        },
        "id": 987
}
```

5.67.2 Response

Table 123. addPortVlan Response

Result	Type	Mandatory	Description
portVlan	String	Yes	Created port VLAN UUID
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "portVlan": <string>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "portVlan": "123e4567-e89b-12d3-a456-426655440000",
                "oem": {}
        },
        "id": 987
}
```

5.68 deletePortVlan

The "deletePortVlan" command destroys an existing VLAN on the Switch Port.

5.68.1 Request

Table 124. deletePortVlan Request

Parameters	Type	Mandatory	Description
portVlan	String	Yes	Managed port VLAN UUID
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "deletePortVlan",
        "params": {
                "portVlan": <string>,
                "oem": <object>
        },
```

```
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "deletePortVlan",
        "params": {
                "portVlan": "123e4567-e89b-12d3-a456-426655440000",
                "oem": {}
        },
        "id": 987
}
```

5.68.2 Response

Table 125. deletePortVlan Response

Result	Type	Mandatory	Description
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 987
}
```

5.69 getPortStaticMacInfo

The "getPortStaticMacInfo" command retrieves detailed information about a Single Static MAC configured on the Switch Port.

5.69.1 Request

Table 126. getPortStaticMacInfo Request

Parameters	Type	Mandatory	Description
staticMac	String	Yes	Managed port Static MAC UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getPortStaticMacInfo",
        "params": {
```

```
                "staticMac": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getPortStaticMacInfo",
        "params": {
                "staticMac": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.69.2 Response

Table 127. getPortStaticMacInfo Response

Result	Type	Mandatory	Description
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOfflin e"</td></tr><tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
name	String	No	Optional name of the component
description	String	No	Optional description of the component
macAddress	String	Yes	Static MAC address for this port
vlanId	Number	No	VLAN identifier for this static MAC
type	String	No	Type of MAC, i.e., "static."
age	Number	No	Age in seconds
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "macAddress": <String>,
                "vlanId": <Number>,
```

```
                "type": <String>,
                "age": <Number>
                "oem": <Object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "macAddress": "00:11:22:33:44:55",
                "vlanId": 112,
                "type": "static",
                "age": null,
                "oem": {}
        },
        "id": 987
}
```

5.70 Set Port Static MAC Attributes

The "setComponentAttributes" described in Section 5.5, setComponentAttributes, allows configuration of the Port Static MAC Attributes listed in the following table.

Table 128. Configurable Port Static MAC Attributes

Attribute	Type	Description
vlanId	Number	VLAN identifier
macAddress	String	MAC address
oem	Object	OEM specific data

5.71 addPortStaticMac

The "addPortStaticMac" command creates a new Static MAC address on the Switch Port.

5.71.1 Request

Table 129. addPortStaticMac Request

Parameters	Type	Mandatory	Description
port	String	Yes	Managed switch port UUID
macAddress	String	Yes	New Static MAC address
vlanId	Number	No	VLAN Identifier
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "addPortStaticMac",
        "params": {
```

```
                "port": <string>,
                "macAddress": <string>,
                "vlanId": <number>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addPortStaticMac",
        "params": {
                "port": "123e4567-e89b-12d3-a456-426655440000",
                "macAddress": "00:11:22:33:44:55",
                "vlanId": 102,
                "oem": {}
        },
        "id": 987
}
```

5.71.2 Response

Table 130. addPortStaticMac Response

Result	Type	Mandatory	Description
staticMac	String	Yes	Created port Static MAC UUID
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "staticMac": <string>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "staticMac": "123e4567-e89b-12d3-a456-426655440000",
                "oem": {}
        },
        "id": 987
}
```

5.72 deletePortStaticMac

The "deletePortStaticMac" command destroys an existing Static MAC on the Switch Port.

5.72.1 Request

Table 131. deletePortStaticMac Request

Parameters	Type	Mandatory	Description
staticMac	String	Yes	Managed port Static MAC UUID
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "deletePortStaticMac",
        "params": {
                "staticMac": <string>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "deletePortStaticMac",
        "params": {
                "staticMac": "123e4567-e89b-12d3-a456-426655440000",
                "oem": {}
        },
        "id": 987
}
```

5.72.2 Response

Table 132. deletePortStaticMac Response

Result	Type	Mandatory	Description
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 987
}
```

5.73 getAclInfo

The "getAclInfo" command retrieves detailed information about a single ACL configured on the Switch.

5.73.1 Request

Table 133. getAclInfo Request

Parameters	Type	Mandatory	Description
acl	String	Yes	Managed ACL UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getAclInfo",
        "params": {
                "acl": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getAclInfo",
        "params": {
                "acl": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 988
}
```

5.73.2 Response

Table 134. getAclInfo Response

Parameters	Type	Mandatory	Description
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr><tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
name	String	No	Optional name of the component
description	String	No	Optional description of the component

<table>
<tr><th>Parameters</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>collections</td><td>Array:
Object</td><td>Yes</td><td>Subcomponents collections
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>name</td><td>String</td><td>Yes</td><td>Collection name</td></tr>
<tr><td>type</td><td>String</td><td>Yes</td><td>Collection type:
"Rules",
"EthernetSwitchPorts"</td></tr>
</table></td></tr>
<tr><td>oem</td><td>Object</td><td>No</td><td>OEM specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"Rules", "EthernetSwitchPorts"}>
                        }
                        ...
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "collections": [
                        {
                                "name": "Rules",
                                "type": "Rules"
                        },
                        {
                                "name": "EthernetSwitchPorts",
                                "type": "EthernetSwitchPorts"
                        }
                ],
                "oem": {}
        },
        "id": 987
}
```

5.74 addAcl

The "addAcl" command creates a new ACL on the Switch Port.

5.74.1 Request

Table 135. addAcl Request

Parameters	Type	Mandatory	Description
switch	String	Yes	Managed switch UUID
ports	Array: String	No	UUIDs of managed Switch Ports to be bind with new ACL
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "addAcl",
        "params": {
                "switch": <string>,
                "ports": [
                        <string>,
                        ...
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addAcl",
        "params": {
                "switch": "123e4567-e89b-12d3-a456-426655440000",
                "ports": [
                        "498e4567-e89b-12d3-1111-426655449999",
                        "34563454-e89b-12d3-1111-435623434566"
                ],
                "oem": {}
        },
        "id": 987
}
```

5.74.2 Response

Table 136. addAcl Response

Result	Type	Mandatory	Description
acl	String	Yes	Created ACL UUID
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "acl": <string>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "acl": "123e4567-e89b-12d3-a456-426655440000",
                "oem": {}
        },
        "id": 987
}
```

5.75 deleteAcl

The "deleteAcl" command removes the existing ACL.

5.75.1 Request

Table 137. deleteAcl Request

Parameters	Type	Mandatory	Description
acl	String	Yes	Managed ACL UUID
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "deleteAcl",
        "params": {
                "acl": <string>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "deleteAcl",
        "params": {
                "acl": "123e4567-e89b-12d3-a456-426655440000",
                "oem": {}
        },
        "id": 987
}
```

5.75.2 Response

Table 138. deleteAcl Response

Result	Type	Mandatory	Description
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
```

```
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 987
}
```

5.76 addAclPort

The "addAclPort" command binds the Switch Port(s) to the existing ACL.

5.76.1 Request

Table 139. addAclPort Request

Result	Type	Mandatory	Description
acl	String	Yes	Managed ACL UUID
ports	Array: String	Yes	UUID of switch ports to be bound to the ACL
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "addAclPort",
        "params": {
                "acl": <string>,
                "ports": [
                        <string>,
                        ...
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addAclPort",
        "params": {
                "acl": "123e4567-e89b-12d3-a456-426655440000",
                "ports": [
                        "cd3e4527-af7b-32d3-a489-987656a3d588"
                ],
                "oem": {}
        },
        "id": 987
}
```

5.76.2 Response

Table 140. addAclPort Response

Parameters	Type	Mandatory	Description
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 987
}
```

5.77 deleteAclPort

The "deleteAclPort" command unbinds the Switch Port(s) from the ACL.

5.77.1 Request

Table 141. deleteAclPort Request

Parameters	Type	Mandatory	Description
acl	String	Yes	Managed ACL UUID.
ports	Array: String	Yes	UUID of switch ports to be unbound from the ACL
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "deleteAclPort",
        "params": {
                "acl": <string>,
                "ports": [
                        <string>,
                        ...
                ],
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "deleteAclPort",
```

```
        "params": {
                "acl": "123e4567-e89b-12d3-a456-426655440000"
                "ports": [
                        "cd3e4527-af7b-32d3-a489-987656a3d588"
                ],
        },
        "id": 987
}
```

5.77.2 Response

Table 142. deleteAclPort Response

Result	Type	Mandatory	Description
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 987
}
```

5.78 getAclRuleInfo

The "getAclRuleInfo" command retrieves detailed information about a single rule configured within the ACL.

5.78.1 Request

Table 143. getAclRuleInfo Request

Parameters	Type	Mandatory	Description
rule	String	Yes	Managed ACL Rule UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getAclRuleInfo",
        "params": {
                "rule": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getAclRuleInfo",
        "params": {
                "rule": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 988
}
```

5.78.2 Response

Table 144. getAclRuleInfo Response

<table>
<tr><th>Parameters</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>status</td><td>Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component:
"Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr>
<tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource:
"OK", "Warning", "Critical"</td></tr>
</table>
</td></tr>
<tr><td>name</td><td>String</td><td>No</td><td>Optional name of the component</td></tr>
<tr><td>description</td><td>String</td><td>No</td><td>Optional description of the component</td></tr>
<tr><td>ruleId</td><td>Number</td><td>Yes</td><td>Rule identifier on the switch.</td></tr>
<tr><td>action</td><td>String</td><td>Yes</td><td>Type of action taken if traffic match the rule:
"Permit", "Deny", "Forward", "Mirror"</td></tr>
<tr><td>forwardMirrorPort</td><td>String</td><td>No</td><td>A UUID of the Managed Switch Port, which traffic should be mirrored/forwarded to0.</td></tr>
<tr><td>mirroredPorts</td><td>Array: String</td><td>No</td><td>UUIDs of mirrored switch ports</td></tr>
<tr><td>mirrorType</td><td>String</td><td>No</td><td>Mirror type:
"Egress", "Ingress", "Bidirectional", "Redirect"</td></tr>
<tr><td>vlanId</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>id</td><td>Number</td><td>Yes</td><td>VLAN identifier</td></tr>
<tr><td>mask</td><td>Number</td><td>No</td><td>VLAN mask number</td></tr>
</table>
</td></tr>
<tr><td>sourceIp</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>address</td><td>String</td><td>Yes</td><td>IPv4 address</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Parameters</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>mask</td><td>String</td><td>No</td><td>IP subnet mask</td></tr>
</table>
</td></tr>
<tr><td>destinationIp</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>address</td><td>String</td><td>Yes</td><td>IPv4 address</td></tr>
<tr><td>mask</td><td>String</td><td>No</td><td>IP subnet mask</td></tr>
</table>
</td></tr>
<tr><td>sourceMac</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>address</td><td>String</td><td>Yes</td><td>MAC address</td></tr>
<tr><td>mask</td><td>String</td><td>No</td><td>MAC mask</td></tr>
</table>
</td></tr>
<tr><td>destinationMac</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>address</td><td>String</td><td>Yes</td><td>MAC address</td></tr>
<tr><td>mask</td><td>String</td><td>No</td><td>MAC mask</td></tr>
</table>
</td></tr>
<tr><td>sourceL4Port</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>port</td><td>Number</td><td>Yes</td><td>L4 port</td></tr>
<tr><td>mask</td><td>Number</td><td>No</td><td>L4 port mask</td></tr>
</table>
</td></tr>
<tr><td>destinationL4Port</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>port</td><td>Number</td><td>Yes</td><td>L4 port</td></tr>
<tr><td>mask</td><td>Number</td><td>No</td><td>L4 port mask</td></tr>
</table>
</td></tr>
<tr><td>protocol</td><td>Number</td><td>No</td><td>Rule L4 protocol identifier compatible with IP specification.</td></tr>
<tr><td>oem</td><td>Object</td><td>No</td><td>OEM specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                  "status": {
                           "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"}>,
                           "health": <{"OK", "Warning", "Critical"}>
                  },
                  "ruleId": <number>,
                  "action": <{"Permit", "Deny", "Forward", "Mirror"}>,
                  "forwardMirrorPort": <string>,
                  "mirroredPorts": [
                           <string>,
                           ...
                  ],
                  "mirrorType": <{"Egress", "Ingress", "Bidirectional", "Redirect"}>,
                  "vlanId": {
                           "id": <number>,
                           "mask": <number>
                  },
                  "sourceIp": {
                           "address": <string>,
                           "mask": <string>
                  },
                  "destinationIp": {
```

```
                        "address": <string>,
                        "mask": <string>
                },
                "sourceMac": {
                        "address": <string>,
                        "mask": <string>
                },
                "destinationMac": {
                        "address": <string>,
                        "mask": <string>
                },
                "sourceL4Port": {
                        "port": <number>,
                        "mask": <number>
                },
                "destinationL4Port": {
                        "port": <number>,
                        "mask": <number>
                },
                "protocol": <number>,
                "oem": {}
        },
        "id": <id>}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "ruleId": 123,
                "action": "Mirror",
                "forwardMirrorPort": "123e4567-e89b-12d3-a456-426655440000",
                "mirroredPorts": [
                        "123e4567-e89b-12d3-a456-426655448673",
                        "59204595-56bd-16d3-s335-448574857000"
                ],
                "mirrorType": "Bidirectional",
                "vlanId": {
                        "id": 1088,
                        "mask": 4095
                },
                "sourceIp": {
                        "address": "192.168.1.0",
                        "mask": "0.0.0.255"
                },
                "sourceMac":{
                        "address": "00:11:22:33:44:55"
                },
                "sourceL4Port": {
                        "port": 22,
                        "mask": 255
                },
                "protocol": 17,
                "oem": {}
        },
        "id": 987
}
```

5.79 addAclRule

The "addAclRule" command creates a new rule on the ACL.

5.79.1 Request

Table 145. addAclRule Request

Parameters	Type	Mandatory	Description
acl	String	Yes	Managed ACL UUID, owner of the rule.
ruleId	Number	No	Rule identifier on the switch.
action	String	Yes	Type of action taken if traffic match the rule: "Permit", "Deny", "Forward", "Mirror"
forwardMirrorPort	String	No	UUID of managed switch port which traffic should be mirrored/forwarded to
mirroredPorts	Array: String	No	UUIDs of mirrored switch ports
mirrorType	String	No	Mirror type: "Egress", "Ingress", "Bidirectional", "Redirect"
vlanId	Object	No	**Attribute** / **Type** / **Mandatory** / **Description** id / Number / Yes / VLAN identifier mask / Number / No / VLAN mask number
sourceIp	Object	No	**Attribute** / **Type** / **Mandatory** / **Description** address / String / Yes / IPv4 address mask / String / No / IP subnet mask
destinationIp	Object	No	**Attribute** / **Type** / **Mandatory** / **Description** address / String / Yes / IPv4 address mask / String / No / IP subnet mask
sourceMac	Object	No	**Attribute** / **Type** / **Mandatory** / **Description** address / String / Yes / MAC address mask / String / No / MAC mask
destinationMac	Object	No	**Attribute** / **Type** / **Mandatory** / **Description** address / String / Yes / MAC address mask / String / Yes / MAC mask
sourceL4Port	Object	No	**Attribute** / **Type** / **Mandatory** / **Description** port / Number / Yes / L4 port mask / Number / No / L4 port mask

Parameters	Type	Mandatory	Description
destinationL4Port	Object	No	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>port</td><td>Number</td><td>Yes</td><td>L4 port</td></tr><tr><td>mask</td><td>Number</td><td>No</td><td>L4 port mask</td></tr></table>
protocol	Number	No	Rule L4 protocol identifier compatible with IP specification.
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "addAclRule",
        "params": {
                "acl": <string>,
                "ruleId": <number>,
                "action": <{"Permit", "Deny", "Forward", "Mirror"}>,
                "forwardMirrorPort": <string>,
                "mirroredPorts": [
                        <string>,
                        ...
                ],
                "mirrorType": <{"Egress", "Ingress", "Bidirectional", "Redirect"}>,
                "vlanId": {
                        "id": <number>,
                        "mask": <number>
                },
                "sourceIp": {
                        "address": <string>,
                        "mask": <string>
                },
                "destinationIp": {
                        "address": <string>,
                        "mask": <string>
                },
                "sourceMac": {
                        "address": <string>,
                        "mask": <string>
                },
                "destinationMac": {
                        "address": <string>,
                        "mask": <string>
                },
                "sourceL4Port": {
                        "port": <number>,
                        "mask": <number>
                },
                "destinationL4Port": {
                        "port": <number>,
                        "mask": <number>
                },
                "protocol": <number>,
                "oem": {}
        },
```

```
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addAclRule",
        "params": {
                "acl": "123e4567-e89b-12d3-a456-426655440000",
                "ruleId": 123,
                "action": "Mirror",
                "forwardMirrorPort": "123e4567-e89b-12d3-a456-426655440000",
                "mirroredPorts": [
                        "123e4567-e89b-12d3-a456-426655448673",
                        "59204595-56bd-16d3-s335-448574857000"
                ],
                "mirrorType": "Bidirectional",
                "vlanId": {
                        "id": 1088,
                        "mask": 4095
                },
                "sourceIp": {
                        "address": "192.168.1.0",
                        "mask": "0.0.0.255"
                },
                "sourceMac":{
                        "address": "00:11:22:33:44:55"
                },
                "sourceL4Port": {
                        "port": 22,
                        "mask": 255
                },
                "protocol": 17,
                "oem": {}
        },
        "id": 987
}
```

5.79.2 Response

Table 146. addAclRule Response

Result	Type	Mandatory	Description
rule	String	Yes	UUID of the newly created rule.
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "rule": <string>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "rule": "123e4567-e89b-12d3-a456-426655440000",
                "oem": {}
        },
        "id": 987
}
```

5.80 deleteAclRule

The "deleteAclRule" command removes a rule from the ACL.

5.80.1 Request

Table 147. deleteAclRule Request

Parameters	Type	Mandatory	Description
rule	String	Yes	UUID of managed ACL Rule
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "deleteAclRule",
        "params": {
                "rule": <string>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "deleteAclRule",
        "params": {
                "rule": "123e4567-e89b-12d3-a456-426655440000",
                "oem": {}
        },
        "id": 987
}
```

5.80.2 Response

Table 148. deleteAclRule Response

Result	Type	Mandatory	Description
oem	object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
```

```
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 987
}
```

5.81 Set ACL Rule Attributes

The "setComponentAttributes" described in Section 5.5, setComponentAttributes, allows configuration of the ACL rule attributes listed in the following table.

Table 149. Configurable ACL Rule Attributes

Attribute	Type	Description
action	String	Type of action taken if traffic match the rule: "Permit", "Deny", "Forward", "Mirror"
ruleId	Number	This is ACL rule ID which determines rule priority
forwardMirrorPort	String	UUID of managed switch port which traffic should be mirrored/forwarded to
mirroredPorts	Array: String	UUIDs of mirrored switch ports
mirrorType	String	Mirror type: "Egress", "Ingress", "Bidirectional", "Redirect"
vlanId	Object	

Attribute	Type	Mandatory	Description
id	Number	Yes	VLAN identifier
mask	Number	No	VLAN mask number

sourceIp	Object		

Attribute	Type	Mandatory	Description
address	String	Yes	IPv4 address
mask	String	No	IP subnet mask

destinationIp	Object		

Attribute	Type	Mandatory	Description
address	String	Yes	IPv4 address
mask	String	No	IP subnet mask

sourceMac	Object		

Attribute	Type	Mandatory	Description
address	String	Yes	MAC address
mask	String	No	MAC mask

destinationMac	Object		

Attribute	Type	Mandatory	Description
address	String	Yes	MAC address
mask	String	No	MAC mask

sourceL4Port	Object		

Attribute	Type	Mandatory	Description

Attribute	Type	Description
		port: Number, Yes, L4 port mask: Number, No, L4 port mask
destinationL4Port	Object	Attribute / Type / Mandatory / Description port: Number, Yes, L4 port mask: Number, No, L4 port mask
protocol	Number	Rule L4 protocol identifier compatible with IP specification.
oem	Object	OEM specific data

5.82 getChassisInfo

The "getChassisInfo" gathers the information about an object of the Chassis type. A Chassis can exist as a Rack, a Drawer, a Module or a Blade.

5.82.1 Request

Table 150. getChassisInfo Request

Parameters	Type	Mandatory	Description
chassis	String	Yes	Managed Chassis UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getChassisInfo",
        "params": {
                "chassis": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getChassisInfo",
        "params": {
                "chassis": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 988
}
```

5.82.2 Response

Table 151. getChassisInfo Response

Result	Type	Mandatory	Description
status	Object	Yes	Attribute / Type / Mandatory / Description state: String, Yes, Known state of the component:

Result	Type	Mandatory	Description
			<table><tr><td></td><td></td><td></td><td>"Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr><tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
name	String	No	Optional name of the component
description	String	No	Optional description of the component
type	String	Yes	Chassis type: "Rack", "Drawer", "Module", "Blade", "Enclosure", "StandAlone", "RackMount", "Card", "Cartridge", "Row", "Pod", "Expansion", "Sidecar", "Zone", "Sled", "Shelf", "Component", "Other".
isManaged	Boolean	Yes	Indicates if the chassis is managed by Parent Manager resource.
locationOffset	Number	Yes	Indicates the location of the resource in Parent Container's Units
parentId	String	Yes	Parent Chassis identifier
powerZone	String	No	Identifier of the Power Zone powering the chassis
thermalZone	String	No	Identifier of the Thermal Zone cooling the chassis
fruInfo	Object	No	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>serialNumber</td><td>String</td><td>No</td><td>Module serial number</td></tr><tr><td>manufacturer</td><td>String</td><td>No</td><td>Manufacturer name</td></tr><tr><td>modelNumber</td><td>String</td><td>No</td><td>Model number</td></tr><tr><td>partNumber</td><td>String</td><td>No</td><td>Part number</td></tr></table>
sku	String	No	Switch SKU
assetTag	String	No	Asset tag
geoTag	String	No	User-defined the geographic location
disaggregatedPowerCoolingSupport	Boolean	No	Indicates if the chassis support disaggregated power and cooling feature
locationId	String	No	User-defined rack unique identified for chassis
allowedActions	Array: String	No	Allowable reset types
indicatorLED	String	No	The state of the indicator LED used to identify the drive: "Lit", "Blinking", "Off".
collections	Array: Object	Yes	Subcomponents collections <table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>name</td><td>String</td><td>Yes</td><td>Collection name</td></tr><tr><td>type</td><td>String</td><td>Yes</td><td>Collection type: "Chassis" "PowerZones", "ThermalZones", "Drives"</td></tr></table>
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"}>, }>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "type": <{"Rack", "Drawer", "Module", "Blade", "Enclosure",
"StandAlone", "RackMount", "Card", "Cartridge", "Row", "Pod", "Expansion", "Sidecar",
"Zone", "Sled", "Shelf", "Component", "Other".
                "size": <number>,
                "locationInfo": <string>,
                "locationInfoFormat": <string>,
                "powerZone": <string>,
                "thermalZone": <string>,
                "fruInfo": {
                        "serialNumber": <string>,
                        "manufacturer": <string>,
                        "modelNumber": <string>,
                        "partNumber": <string>
                },
                "sku": <string>,
                "assetTag": <string>,
                "geoTag": <string>,
                "locationId": <string>,
                "allowedActions": [<string>],
                "indicatorLED": <{"Lit", "Blinking", "Off"}>,
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"Chassis" "PowerZones", "ThermalZones",
"Drives"}>
                        }
                ...
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "type": "Drawer",
                "size": 2,
                "locationInfo": "Id:Drawer1 ParentId:Rack1",
                "locationInfoFormat": "Id:string ParentId:string",
                "powerZone": "123e4567-e89b-12d3-a456-426655440000",
                "thermalZone": "cd3e4527-af7b-32d3-a489-987656a3d588",
                "fruInfo": {
                        "serialNumber": "123fed3029c-b23394-12",
                        "manufacturer": "Intel Corporation",
                        "modelNumber": "E323",
```

```
                        "partNumber": "29ee2220939"
                },
                "sku": "AZT38KX",
                "assetTag": "To be filled by OEM",
                "geoTag": "Poland/Gdansk",
                "locationId": "0-1-0-1",
                "allowedActions": [ "HardReset", "SoftReset" ],
                "indicatorLED": "Off",
                "collections": [
                        {
                                "name": "Power",
                                "type": "PowerZones"
                        },
                        {
                                "name": "Thermal",
                                "type": "ThermalZones"
                        },
                        {
                                "name": "Chassis",
                                "type": "Chassis"
                        },
                        {
                                "name": "Drives",
                                "type": "Drives"
                        }

                ],
                "oem": {}
        },
        "id": 987
}
```


5.83 SetChassis Attributes

The `"setComponentAttributes"` described in Section 5.5, setComponentAttributes, allows configuration of the Chassis in the following table.

Table 152. SetChassis Attributes

Attribute	Type	Description
`assetTag`	String	Tag assigned to the asset.
`geoTag`	String	User defined string for geographic location
`locationId`	String	User defined string for rack unique identifier for chassis
`allowedActions`	Array: String	Allowable reset types
`reset`	String	Reset chassis component. Allowable values: `"On"`, `"ForceOff"`, `"GracefulShutdown"`, `"GracefulRestart"`, `"ForceRestart"`, `"Nmi"`, `"ForceOn"`, `"PushPowerButton"`.
`oem`	Object	OEM specific data

5.84 getPowerZoneInfo

The AMC retrieves full information about specific Power Zones by sending the `"getPowerZoneInfo"` command to appropriate GAM Module.

5.84.1 Request

Table 153. getPowerZoneInfo Request

Parameters	Type	Mandatory	Description
zone	String	Yes	Managed power zone UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getPowerZoneInfo",
        "params": {
                "zone": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getPowerZoneInfo",
        "params": {
                "zone": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.84.2 Response

The GAM Module replies with the following response in successful case:

Table 154. getPowerZoneInfo Response

Parameters	Type	Mandatory	Description
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr><tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
name	String	No	Optional name of the component
description	String	No	Optional description of the component
powerConsumedWatts	Number	No	Power consumed by the chassis
powerRequestedWatts	Number	No	Power budget requested for the chassis

Parameters	Type	Mandatory	Description
powerCapacityWatts	Number	No	Maximum rated power capacity of the zone
powerAllocatedWatts	Number	No	Power allocated for the chassis
powerAvailableWatts	Number	No	Available power for the chassis
collections	Array: Object	Yes	Subcomponents collections Attribute / Type / Mandatory / Description name / String / Yes / Collection name type / String / Yes / Collection type: "PSUs", "ChassisSensors"
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,"
                                                                "InTest",
"Starting", "Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },

                "powerConsumedWatts": <Number>,
                "powerRequestedWatts": <Number>,
                "powerCapacityWatts": <Number>,
                "powerAllocatedWatts": <Number>,
                "powerAvailableWatts": <Number>,
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"PSUs", "ChassisSensors"}>
                        }
                …
                ]
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "powerConsumedWatts": 1250,
                "powerRequestedWatts": 2000,
                "powerCapacityWatts": 8000,
                "powerAllocatedWatts": 4000,
                "powerAvailableWatts": 8000,
```

```
                "collections": [
                        {
                                "name": "PSUs",
                                "type": "PSUs"
                        }
                ],
                "oem": {}
        },
        "id": 987
}
```

5.85 getPsuInfo

The AMC retrieves full information about the specific Power Supply Unit by sending the "getPsuInfo" command to the appropriate GAM Module.

5.85.1 Request

Table 155. getPsuInfo Request

Parameters	Type	Mandatory	Description
psu	String	Yes	Managed PSU UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getPsuInfo",
        "params": {
                "psu": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getPsuInfo",
        "params": {
                "psu": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.85.2 Response

The GAM Module replies with the following response in successful case:

Table 156. getPsuInfo Response

<table>
<tr><th>Parameters</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>status</td><td>Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr>
<tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr>
</table>
</td></tr>
<tr><td>name</td><td>String</td><td>No</td><td>Optional name of the component</td></tr>
<tr><td>description</td><td>String</td><td>No</td><td>Optional description of the component</td></tr>
<tr><td>powerSupplyType</td><td>String</td><td>No</td><td>Power supply type. Valid values: "AC", "DC", "ACorDC"</td></tr>
<tr><td>lineInputVoltageType</td><td>String</td><td>No</td><td>The line voltage supported as the input signal. Valid values: "Unknown", "ACLowLine", "ACMidLine", "ACHighLine", "DCNeg48V", "DC380V", "AC120V", "AC240V", "AC277V", "ACandDCWideRange", "ACWideRange", "DC240V"</td></tr>
<tr><td>lineInputVoltageVolts</td><td>Number</td><td>No</td><td>The input voltage at which PSU is operating</td></tr>
<tr><td>firmwareVersion</td><td>String</td><td>No</td><td>Firmware version</td></tr>
<tr><td>powerCapacityWatts</td><td>Number</td><td>No</td><td>Power capacity in watts</td></tr>
<tr><td>lastPowerOutputWatts</td><td>Number</td><td>No</td><td>Last known power output in watts</td></tr>
<tr><td>indicatorLED</td><td>String</td><td>No</td><td>LED operation reporting asset state. Available values: "Lit", "Blinking", "Off"</td></tr>
<tr><td>fruInfo</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>serialNumber</td><td>String</td><td>No</td><td>Module serial number</td></tr>
<tr><td>manufacturer</td><td>String</td><td>No</td><td>Manufacturer name</td></tr>
<tr><td>modelNumber</td><td>String</td><td>No</td><td>Model number</td></tr>
<tr><td>partNumber</td><td>String</td><td>No</td><td>Part number</td></tr>
</table>
</td></tr>
<tr><td>oem</td><td>Object</td><td>No</td><td>OEM specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result" : {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,"
```

```
                                                                    "InTest",
"Starting", "Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "powerSupplyType": <{"AC", "DC", "ACorDC"}>,
                "lineInputVoltageType": <{"Unknown", "ACLowLine", "ACMidLine",
"ACHighLine", "DCNeg48V", "DC380V", "AC120V", "AC240V", "AC277V", "ACandDCWideRange",
"ACWideRange", "DC240V"}>,
                "lineInputVoltageVolts": <number>,
                "firmwareVersion": <string>,
                "powerCapacityWatts": <number>,
                "lastPowerOutputWatts": <number>,
                "indicatorLED": <{"Lit", "Blinking", "Off"}>,
                "fruInfo": {
                        "serialNumber": <string>,
                        "manufacturer": <string>,
                        "modelNumber": <string>,
                        "partNumber": <string>
                },
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },

                "powerSupplyType": "AC",
                "lineInputVoltageType": "AC240V"
                "lineInputVoltageVolts": 230,
                "firmwareVersion": "2.03A",
                "powerCapacityWatts": 2500,
                "lastPowerOutputWatts": 2500,
                "indicatorLED": "Off",
                "fruInfo": {
                        "serialNumber": "123fed3029c-b23394-12",
                        "manufacturer": "Intel Corporation",
                        "modelNumber": "E323",
                        "partNumber": "29ee2220939"
                },
                "oem": {}
        },
        "id": 345
}
```

5.86 Set PSU Attributes

The `"setComponentAttributes"` described in Section 5.5, setComponentAttributes, allows configuration of the PSU Attributes listed in the following table.

Table 157. Configurable PSU Attributes

Attribute	Type	Description
`state`	String	Triggers `RequestStateChange` on particular PSU. Allowed values: `"Enabled"`, `"Disabled"`
`oem`	Object	OEM specific data

5.87 getThermalZoneInfo

The AMC retrieves full information about a specific Thermal Zone by sending the `"getThermalZoneInfo"` command to the appropriate GAM Module.

5.87.1 Request

Table 158. getThermalZoneInfo Request

Parameters	Type	Mandatory	Description
`zone`	String	Yes	Managed thermal zone UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getThermalZoneInfo",
        "params": {
                 "zone": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getThermalZone",
        "params": {
                 "zone": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.87.2 Response

The GAM Module replies with the following response if the request is successful:

Table 159. getThermalZoneInfo Response

Parameters	Type	Mandatory	Description
`status`	Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** `state` / String / Yes / Known state of the component:

<table>
<tr><th>Parameters</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td></td><td></td><td></td><td>"Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr>
<tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr>
</table>
</td></tr>
<tr><td>name</td><td>String</td><td>No</td><td>Optional name of the component</td></tr>
<tr><td>description</td><td>String</td><td>No</td><td>Optional description of the component</td></tr>
<tr><td>volumetricAirflowCfm</td><td>Number</td><td>No</td><td>Rack level Power Thermal-Aware Solution (PTAS) telemetry, volumetric airflow in the zone expressed in cubic feet per minute (cfm)</td></tr>
<tr><td>desiredSpeedPwm</td><td>Number</td><td>No</td><td>Desired fan speed in the zone Pulse-width Modulation (PWM)</td></tr>
<tr><td>collections</td><td>Array: Object</td><td>Yes</td><td>Subcomponents collections
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>name</td><td>String</td><td>Yes</td><td>Collection name</td></tr>
<tr><td>type</td><td>String</td><td>Yes</td><td>Collection type: "Fans", "ChassisSensors"</td></tr>
</table>
</td></tr>
<tr><td>oem</td><td>Object</td><td>No</td><td>OEM specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,"
                                                                "InTest",
"Starting", "Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "volumetricAirflowCfm": <number>,
                "desiredSpeedPwm": <number>,
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"Fans", "ChassisSensors"}>
                        }
                ...
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "volumetricAirflowCfm": 300,
                "desiredSpeedPwm": 20,
                "collections": [
                        {
                                "name": "Fans",
                                "type": "Fans"
                        }
                ],
                "oem": {}
        },
        "id": 345
}
```

5.88 Set Thermal Zone Attributes

The `"setComponentAttributes"` described in Section 5.5, setComponentAttributes, allows configuration of the Thermal Zone attributes listed in the following table.

Table 160. Configurable PSU Attributes

Attribute	Type	Description
`desiredSpeedPwm`	Number	The speed of the fan in Thermal Zone (PWM)
`oem`	Object	OEM specific data

5.89 getFanInfo

The RMM asset manager can query information about a specific fan by sending the `"getFanInfo"` command to the appropriate GAM Module.

5.89.1 Request

Table 161. getFanInfo Request

Parameters	Type	Mandatory	Description
`fan`	String	Yes	Managed fan UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getFanInfo",
        "params": {
                "fan": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
```

```
        "method": "getFanInfo",
        "params": {
                "fan": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.89.2 Response

The GAM Module replies with the following response if the request is successful:

Table 162. getFanInfo Response

Parameters	Type	Mandatory	Description
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr><tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
name	String	No	Optional name of the component
description	String	No	Optional description of the component
physicalContext	String	No	Describes the area or device associated with this fan: "Room", "Intake", "Exhaust", "Front", "Back", "Upper", "Lower", "CPU", "GPU", "Backplane", "SystemBoard", "PowerSupply", "VoltageRegulator", "StorageDevice", "NetworkingDevice", "ComputeBay", "StorageBay", "NetworkBay", "ExpansionBay", "PowerSupplyBay"
currentSpeed	Number	Yes	Fan current speed
currentSpeedUnits	String	No	Speed reading units. Allowed values: "Percent", "RPM"
fruInfo	Object	No	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>serialNumber</td><td>String</td><td>No</td><td>Module serial number</td></tr><tr><td>manufacturer</td><td>String</td><td>No</td><td>Manufacturer name</td></tr><tr><td>modelNumber</td><td>String</td><td>No</td><td>Model number</td></tr><tr><td>partNumber</td><td>String</td><td>No</td><td>Part number</td></tr></table>
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
```

```
        "result" : {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,"
                                                                "InTest",
"Starting", "Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "physicalContext": <{"Room", "Intake", "Exhaust", "Front", "Back",
"Upper", "Lower", "CPU", "GPU", "Backplane", "SystemBoard", "PowerSupply",
"VoltageRegulator", "StorageDevice", "NetworkingDevice", "ComputeBay", "StorageBay",
"NetworkBay", "ExpansionBay", "PowerSupplyBay"}>,
                "currentSpeed": <number>,
                "currentSpeedUnits": <{"Percent", "RPM"}>,
                "fruInfo": {
                        "serialNumber": <string>,
                        "manufacturer": <string>,
                        "modelNumber": <string>,
                        "partNumber": <string>
                },
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "physicalContext": "Backplane",
                "currentSpeed": 5000,

                "currentSpeedUnits": "RPM",
                "fruInfo": {
                        "serialNumber": "123fed3029c-b23394-12",
                        "manufacturer": "Intel Corporation",
                        "modelNumber": "E323",
                        "partNumber": "29ee2220939"
                },
                "oem": {}
        },
        "id": 987
}
```

5.90 getChassisSensorInfo

The RMM Asset Manager can query specific sensor information by sending the `"getChassisSensorInfo"` command to the appropriate GAM Module.

5.90.1 Request

Table 163. getChassisSensorInfo Request

Parameters	Type	Mandatory	Description
sensor	String	Yes	Managed sensor UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getChassisSensorInfo",
        "params": {
                 "sensor": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getChassisSensorInfo",
        "params": {
                 "sensor": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.90.2 Response

The GAM Module replies with the following response if the request is successful:

Table 164. getChassisSensorInfo Response

Parameters	Type	Mandatory	Description
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr><tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
name	String	No	Optional name of the component
description	String	No	Optional description of the component
reading	Number	Yes	Sensor reading value

Parameters	Type	Mandatory	Description
readingUnit	String	Yes	Reading unit. Valid values: "Volt", "Celsius"
sensorNumber	Number	Yes	Numerical identifier representing sensor
physicalContext	String	No	Describes the area or device associated with this sensor: "Room", "Intake", "Exhaust", "Front", "Back", "Upper", "Lower", "CPU", "GPU", "Backplane", "SystemBoard", "PowerSupply", "VoltageRegulator", "StorageDevice", "NetworkingDevice", "ComputeBay", "StorageBay", "NetworkBay", "ExpansionBay", "PowerSupplyBay"
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result" : {
                    "status": {
                           "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare,"
                                                                    "InTest",
"Starting", "Absent", "UnavailableOffline"}>,
                           "health": <{"OK", "Warning", "Critical"}>
                    },
                    "reading": <Number>,
                    "physicalContext": <{"Room", "Intake", "Exhaust", "Front", "Back",
"Upper", "Lower", "CPU", "GPU", "Backplane", "SystemBoard", "PowerSupply",
"VoltageRegulator", "StorageDevice", "NetworkingDevice", "ComputeBay", "StorageBay",
"NetworkBay", "ExpansionBay", "PowerSupplyBay"}>,
                    "readingUnit": <{"Volt", "Celsius"}>,
                    "sensorNumber": <Number>,
                    "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                    "status": {
                           "state": "Enabled",
                           "health": "OK"
                    },
                    "physicalContext": "Backplane",
                    "readingUnit": "Celsius",
                    "reading": 36.6,
                    "sensorNumber": 13
                    "oem": {}
        },
        "id": 987
}
```

5.91 Set Fan Attributes

The "setComponentAttributes" described in Section 5.5, setComponentAttributes, allows configuration of the Fan Attributes listed in the following table.

Table 165. Configurable Fan Attributes

Attribute	Type	Description
oem	Object	OEM specific data

5.92 getAuthorizationCertificate

The AMC can use the Authorization Certificates obtained from underlying GAM Modules.

5.92.1 Request

Table 166. getAuthorizationCertificate Request

Parameters	Type	Mandatory	Description
certificateType	String	Yes	Type of the certificate: "PODM", "RMM"

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getAuthorizationCertificate",
        "params": {
                "certificateType": <string>,
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getAuthorizationCertificate",
        "params": {
                "certificateType": "PODM"
        },
        "id": 987
}
```

5.92.2 Response

The GAM Module replies with the following response if the request is successful:

Table 167. getAuthorizationCertificate Response

Parameters	Type	Mandatory	Description
name	String	No	Optional name of the component
description	String	No	Optional description of the component
certificate	String	Yes	Encoded certificate
encodingMethod	String	Yes	Certificate encoding method: "BASE64"
certificateHash	String	Yes	Encoded certificate hash
hashMethod	String	Yes	Hashing method: "MD5"
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result" : {
                "certificate": <string>,
                "encodingMethod": <{"BASE64"}>,
                "certificateHash": <string>,
                "hashMethod": <{"MD5"}>,
                "oem": <Object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "certificate": "ABCDEFGHIJKLMNOPQRSTUVWXYZ",
                "encodingMethod": "BASE64",
                "certificateHash": "be3cd5cb003392654570dc1e54641764",
                "hashMethod": "MD5",
                "oem": {}
        },
        "id": 987
}
```

Note: If the queried certificate is not yet available, the appropriate error message should be returned (`"Object not found"`). The AMC should be aware of a possible time delay required for the GAM Module to obtain a certificate and expose it through the GAMI.

5.93 getStorageServiceInfo

The `"getStorageServiceInfo"` command retrieves information about the Storage Service.

5.93.1 Request

Table 168. getStorageServiceInfo Request

Parameters	Type	Mandatory	Description
`service`	String	Yes	Storage Service instance UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getStorageServiceInfo",
        "params": {
                "service": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getStorageServiceInfo",
        "params": {
                "service": "123e4567-e89b-12d3-a456-426655440000"
```

```
        },
        "id": 987
}
```

5.93.2 Response

Table 169. getStorageServiceInfo Response

Result	Type	Mandatory	Description
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr><tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
name	String	No	Optional name of the component
description	String	No	Optional description of the component
collections	Array: Object	Yes	Subcomponents collections <table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>name</td><td>String</td><td>Yes</td><td>Collection name</td></tr><tr><td>type</td><td>String</td><td>Yes</td><td>Collection type: "Drives", "Volumes", "StoragePools"</td></tr></table>
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
"StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"Drives", "Volumes", "StoragePools"}>
                        }
                        ...
                ]
                "oem": <object>
        },
```

```
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "collections": [
                        {
                                "name": "Drives",
                                "type": "Drives"
                        },
                        {
                                "name": "Volumes",
                                "type": "Volumes"
                        },
                        {
                                "name": "StoragePools",
                                "type": "StoragePools"
                        }
                ],
                "oem": {}
        },
        "id": 987
}
```

5.94 getStoragePoolInfo

Container data storage is capable of providing capacity conforming to one of its supported classes of service.

Note: The storage pool does not support Input/Output (I/O) to its data storage.

The `"getStoragePoolInfo"` command retrieves detailed information about a single Storage Pool. The Storage Pool may be defined on top of physical drives or other Storage Pools.

5.94.1 Request

Table 170. getStoragePoolInfo Request

Parameters	Type	Mandatory	Description
`storagePool`	String	Yes	Managed storage pool UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getStoragePoolInfo",
        "params": {
                "storagePool": <String>
        },
        "id": <id>
}
```

Example:

```
{
```

```
        "jsonrpc": "2.0",
        "method": "getStoragePool",
        "params": {
                "storagePool": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.94.2 Response

Table 171. getStoragePoolInfo Response

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>status</td><td>Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component:
"Enabled",
"Disabled",
"StandbyOffline",
"StandbySpare,"<br>"InTest", "Starting",
"Absent",
"UnavailableOffline"</td></tr>
<tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource:
"OK", "Warning",
"Critical"</td></tr>
</table>
</td></tr>
<tr><td>name</td><td>String</td><td>No</td><td>Optional name of the component</td></tr>
<tr><td>description</td><td>String</td><td>No</td><td>Optional description of the component</td></tr>
<tr><td>blockSizeBytes</td><td>Number</td><td>Yes</td><td>Block size for the Storage Pool</td></tr>
<tr><td>capacity</td><td>Object</td><td>Yes</td><td>Storage Pool capacity details.
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>consumedBytes</td><td>Number</td><td>No</td><td>The number of bytes consumed in this data store for this data type.</td></tr>
<tr><td>allocatedBytes</td><td>Number</td><td>No</td><td>The number of bytes currently allocated by the storage system in this data store for this data type.</td></tr>
<tr><td>guaranteedBytes</td><td>Number</td><td>No</td><td>The number of bytes the storage system guarantees can be allocated in this data store for this data type.</td></tr>
<tr><td>provisionedBytes</td><td>Number</td><td>No</td><td>The maximum number of bytes that can be allocated in this data store for this data type.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td>capacitySources</td><td>Array: Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>consumedBytes</td><td>Number</td><td>No</td><td>The number of bytes consumed in this data store for this data type.</td></tr>
<tr><td>allocatedBytes</td><td>Number</td><td>No</td><td>The number of bytes currently allocated by the storage system in this data store for this data type.</td></tr>
<tr><td>guaranteedBytes</td><td>Number</td><td>No</td><td>The number of bytes the storage system guarantees can be allocated in this data store for this data type.</td></tr>
<tr><td>provisionedBytes</td><td>Number</td><td>No</td><td>The maximum number of bytes that can be allocated in this data store for this data type.</td></tr>
<tr><td>providingDrives</td><td>Array: String</td><td>No</td><td>The drive or drives UUID that provide this space.</td></tr>
<tr><td>providingVolumes</td><td>Array: String</td><td>No</td><td>The volume or volumes UUID that provide this space.</td></tr>
<tr><td>providingPools</td><td>Array: String</td><td>No</td><td>The pool or pools that provide this space.</td></tr>
</table>
</td></tr>
<tr><td>identifiers</td><td>Array: Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>durableName</td><td>String</td><td>Yes</td><td>This indicates the worldwide, the persistent name of the resource.</td></tr>
<tr><td>durableNameFormat</td><td>String</td><td>Yes</td><td>This represents the format of the DurableName property. Values: "NAA", "iQN", "FC_WWN", "UUID", "EUI", "NQN", "NSID"</td></tr>
</table>
</td></tr>
<tr><td>collections</td><td>Array: Object</td><td>Yes</td><td>Contains a list of allocated volumes and storage pools. Related resources collections.
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
</table>
</td></tr>
</table>

Result	Type	Mandatory	Description			
			name	String	Yes	Collection name
			type	String	Yes	Collection type: "StoragePools", "Volumes".
oem	Object	No	OEM specific data			

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",
        "StandbySpare,","InTest", "Starting", "Absent",
        "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "capacity": {
                        "consumedBytes": <Number>,
                        "allocatedBytes": <Number>,
                        "guaranteedBytes": <Number>,
                        "provisionedBytes": <Number>
                }
                "capacitySources": [
                        {
                                consumedBytes: <Number>,
                                allocatedBytes: <Number>,
                                guaranteedBytes: <Number>,
                                providingDrives: [
                                                <String>, …
                                ],
                                providingVolumes: [
                                                <String>, …
                                ],
                                providingPools: [
                                                <String>, …
                                ]
                        }
                ],
                "blockSizeBytes": <Number>,
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"Volumes", "StoragePools"}>
                        }
                        ...
                ]
        "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
```

```
                },
                "capacity": {
                        "consumedBytes": 234534527,
                        "allocatedBytes": 1099511627776,
                        "guaranteedBytes": 1099511627776,
                        "provisionedBytes": 1099511627776
                }
                "capacitySources": [
                        {
                                consumedBytes: 100849846,
                                allocatedBytes: 549755813888,
                                guaranteedBytes: 549755813888,
                                providingPools: [
                                                "b2fe60a7-33a8-63bb-da16-
3e7b1a835ee5"
                                ]
                        },
                        {
                                consumedBytes: 133684681,
                                allocatedBytes: 549755813888,
                                guaranteedBytes: 549755813888,
                                providingPools: [
                                                "cca5b252-112a-78e3-23a1-
5bec7252311f"
                                ]
                        }
                ],

                "blockSizeBytes": 512,

                "collections": [
                        {
                                "name": "Volumes",
                                "type": "Volumes"
                        }
                ]
                "oem": {}
        },
        "id": 987
}
```


5.95 addStoragePool

The `"addStoragePool"` command is used to create logical storage capacity for volume creation. The Storage Pool may be created on top of Drives or other Storage Pools.

5.95.1 Request

Table 172. addStoragePool Request

Parameters	Type	Mandatory	Description
`blockSizeBytes`	Number	No	Maximum Block size in bytes.

<table>
<tr><th>Parameters</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>capacitySources</td><td>Array: Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>providingDrives</td><td>Array: String</td><td>No</td><td>The drive or drives UUID that provide this space.</td></tr>
<tr><td>providingVolumes</td><td>Array: String</td><td>No</td><td>The volume or volumes UUID that provide this space.</td></tr>
<tr><td>providingPools</td><td>Array: String</td><td>No</td><td>The pool or pools that provide this space.</td></tr>
</table>
</td></tr>
<tr><td>oem</td><td>Object</td><td>No</td><td>OEM specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "addStoragePool",
        "params": {
                "blockSizeBytes": <Number>,
                "capacitySources": [
                        {
                                "providingDrives": <String>,
                                "providingVolumes": <String>,
                                "providingPools": <String>
                        }
                ]
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addStoragePool",
        "params": {
                "blockSizeBytes": <Number>,
                "capacitySources": [
                        {
                                "providingDrives": "f293a746-00ae-31b3-c2f4-
a95e0d4f2a73"
                        }
                ]
        },
        "id": <id>
}
```

5.95.2 Response

Table 173. addStoragePool Response

Result	Type	Mandatory	Description
storagePool	String	Yes	Created logical drive UUID
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "storagePool": <string>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "storagePool": "6cde4560-e89b-82d3-a456-4266c544aab1",
                "oem": {}
        },
        "id": 987
}
```

5.96 deleteStoragePool

The "deleteStoragePool" command gives the user ability to delete a Storage Pool previously created with the "addStoragePool" command or if a Storage Pool is in absent state. Released storage capacity should be returned to the pool on successful "deleteStoragePool" command execution.

5.96.1 Request

Table 174. deleteStoragePool Request

Result	Type	Mandatory	Description
storagePool	String	Yes	Managed storage pool UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "deleteStoragePool",
        "params": {
                "storagePool": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "storagePool",
        "params": {
                "storagePool": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.96.2 Response

Table 175. deleteStoragePool Response

Parameters	Type	Mandatory	Description
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 987
}
```

5.97 getVolumeInfo

The "getVolumeInfo" command retrieves detailed information about a single logical volume. The Logical Volume may be any object in the storage infrastructure hierarchy definition representing the logical storage object including block device, RAID array or RAID logical drive, logical volume group or logical volume. The volume may be defined on top of storage pools, drives, and volumes.

5.97.1 Request

Table 176. getVolumeInfo Request

Parameters	Type	Mandatory	Description
volume	String	Yes	Managed volume UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getVolumeInfo",
        "params": {
                "volume": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getVolumeInfo",
        "params": {
                "volume": "123e4567-e89b-12d3-a456-426655440000"
        },
```

```
        "id": 987
}
```

5.97.2 Response

Table 177. getVolumeInfo Response

Result	Type	Mandatory	Description
status	Object	Yes	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>state</td><td>String</td><td>Yes</td><td>Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare," "InTest", "Starting", "Absent", "UnavailableOffline"</td></tr><tr><td>health</td><td>String</td><td>Yes</td><td>Overall health state from the view of this resource: "OK", "Warning", "Critical"</td></tr></table>
name	String	No	Optional name of the component
description	String	No	Optional description of the component
capacity	Object	Yes	Volume capacity details.<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>consumedBytes</td><td>Number</td><td>No</td><td>The number of bytes consumed in this data store for this data type.</td></tr><tr><td>allocatedBytes</td><td>Number</td><td>No</td><td>The number of bytes currently allocated by the storage system in this data store for this data type.</td></tr><tr><td>guaranteedBytes</td><td>Number</td><td>No</td><td>The number of bytes the storage system guarantees can be allocated in this data store for this data type.</td></tr></table>

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>provisionedBytes</td><td>Number</td><td>No</td><td>The maximum number of bytes that can be allocated in this data store for this data type.</td></tr>
</table>
</td></tr>
<tr><td>capacitySources</td><td>Array: Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>consumedBytes</td><td>Number</td><td>No</td><td>The number of bytes consumed in this data store for this data type.</td></tr>
<tr><td>allocatedBytes</td><td>Number</td><td>No</td><td>The number of bytes currently allocated by the storage system in this data store for this data type.</td></tr>
<tr><td>guaranteedBytes</td><td>Number</td><td>No</td><td>The number of bytes the storage system guarantees can be allocated in this data store for this data type.</td></tr>
<tr><td>provisionedBytes</td><td>Number</td><td>No</td><td>The maximum number of bytes that can be allocated in this data store for this data type.</td></tr>
<tr><td>providingDrives</td><td>Array: String</td><td>No</td><td>The drive or drives UUID that provide this space.</td></tr>
<tr><td>providingVolumes</td><td>Array: String</td><td>No</td><td>The volume or volumes UUID that provide this space.</td></tr>
<tr><td>providingPools</td><td>Array: String</td><td>No</td><td>The pool or pools that provide this space.</td></tr>
</table>
</td></tr>
<tr><td>volumeType</td><td>String</td><td>No</td><td>Volume type: "RawDevice", "NonRedundant", "Mirrored", "StripedWithParity", "SpannedMirrors", "SpannedStripesWithParity"</td></tr>
<tr><td>encrypted</td><td>Boolean</td><td>No</td><td>Is this volume encrypted</td></tr>
<tr><td>encryptionTypes</td><td>Array: String</td><td>No</td><td>Encryption type: "NativeDriveEncryption", "ControllerAssisted", SoftwareAssisted".</td></tr>
</table>

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>identifiers</td><td>Array: Object</td><td>Yes</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>durableName</td><td>String</td><td>Yes</td><td>This indicates the worldwide, the persistent name of the resource.</td></tr>
<tr><td>durableNameFormat</td><td>String</td><td>Yes</td><td>This represents the format of the DurableName property. Values: "NAA", "iQN", "FC_WWN", "UUID", "EUI", "NQN",</td></tr>
</table>
</td></tr>
<tr><td>blockSizeBytes</td><td>Number</td><td>No</td><td>The size of the smallest addressible unit (Block) of this volume in bytes</td></tr>
<tr><td>optimumIoSizeBytes</td><td>Number</td><td>No</td><td>The size in bytes of this Volume's optimum IO size.</td></tr>
<tr><td>accessCapabilities</td><td>Array: String</td><td>Yes</td><td>List of supported access capabilities. Allowed values: "Read", "Write", "WriteOnce", "Append", "Streaming".</td></tr>
<tr><td>bootable</td><td>Boolean</td><td>No</td><td>Indicates if the logical drive is populated with a bootable image.</td></tr>
<tr><td>isThinProvisioned</td><td>Boolean</td><td>No</td><td>Indicates whether the volume is thin provisioned.</td></tr>
<tr><td>replicaInfos</td><td>Array: Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>replicaReadOnlyAccess</td><td>String</td><td>Yes</td><td>This property specifies whether the source, the target, or both elements are read-only to the host. Values: "SourceElement", "ReplicaElement", "Both".</td></tr>
<tr><td>replicaType</td><td>String</td><td>Yes</td><td>ReplicaType describes the intended outcome of the replication. Values: "Mirror", "Snapshot", "Clone", "TokenizedClone"</td></tr>
<tr><td>replicaRole</td><td>String</td><td>Yes</td><td>The source or target role of this replica. Values: "Source", "Target"</td></tr>
<tr><td>replica</td><td>String</td><td>Yes</td><td>The resource UUID that is the source of this replica.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>collections</td><td>Array: Object</td><td>Yes</td><td>Related resources collections.
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>name</td><td>String</td><td>Yes</td><td>Collection name</td></tr>
<tr><td>type</td><td>String</td><td>Yes</td><td>Collection type: "Endpoints". For associated Volumes, StoragePools and Disks, refer to capacitySources.</td></tr>
</table></td></tr>
<tr><td>oem</td><td>object</td><td>No</td><td>OEM specific data</td></tr>
</table>

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": <{"Enabled", "Disabled", "StandbyOffline",

        "StandbySpare,","InTest", "Starting", "Absent", "UnavailableOffline"}>,
                        "health": <{"OK", "Warning", "Critical"}>
                },
                "capacity": {
                        "consumedBytes": <Number>,
                        "allocatedBytes": <Number>,
                        "guaranteedBytes": <Number>,
                        "provisionedBytes": <Number>
                }
                "capacitySources": [
                        {
                                consumedBytes: <Number>,
                                allocatedBytes: <Number>,
                                guaranteedBytes: <Number>,
                                providingDrives: [
                                                <String>, …
                                ],
                                providingVolumes: [
                                                <String>, …
                                ],
                                providingPools: [
                                                <String>, …
                                ]
                        }
                ]
                "volumeType": <{"RawDevice", "NonRedundant", "Mirrored",
"StrippedWithParity", "SpannedMirrors", "SpannedStripesWithParity"}>,
                "encrypted": <Boolean>,
                "encryptionTypes":
[<{"NativeDriveEncryption","ControllerAssisted","SoftwareAssisted"}>],
                "identifiers": [
                        {
                                "durableName": <String>,
                                "durableNameFormat": <{"NAA", "iQN", "FC_WWN",
"UUID", "EUI", "NQN",

"NSID"}>
```

```
                        },
                        …
                ],
                "blockSizeBytes": <Number>,
                "optimumIoSizeBytes": <Number>,
                "accessCapabilities": [<{"Read", "Write", "WriteOnce", "Append",
"Streaming"}>],
                "bootable": <Boolean>,
                "replicaInfos": [
                        {
                                "replicaReadOnlyAccess": <{"SourceElement",
"ReplicaElement", "Both"}>,
                                "replicaType": <{"Mirror", "Snapshot", "Clone",
"TokenizedClone"}>,
                                "replicaRole": <{"Source", "Target"}>,
                                "replica": <String>,
                        },
                        …
                ],
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"Drives", "Volumes", "StoragePools"}>
                        }
                        ...
                ]
        "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "name": "nvme3n0"
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "capacity": {
                        "consumedBytes": 234534527,
                        "allocatedBytes": 1099511627776,
                        "guaranteedBytes": 1099511627776,
                        "provisionedBytes": 1099511627776
                }
                "capacitySources": [
                        {
                                consumedBytes: 100849846,
                                allocatedBytes: 549755813888,
                                guaranteedBytes: 549755813888,
                                providingPools: [
                                        "b2fe60a7-33a8-63bb-da16-
3e7b1a835ee5"
                                ]
                        },
                        {
                                consumedBytes: 133684681,
                                allocatedBytes: 549755813888,
                                guaranteedBytes: 549755813888,
```

```
                                providingPools: [
                                                        "cca5b252-112a-78e3-23a1-
5bec7252311f"
                                ]
                        }
                ],
                "volumeType": "NonRedundant",
                "encrypted": false,
                "encryptionTypes": [],
                "identifiers": []
                "blockSizeBytes": 512,
                "optimumIoSizeBytes": 4096,
                "accessCapabilities": [ "Read", "Write" ],
                "bootable": false,
                "replicaInfos": [
                        {
                                "replicaReadOnlyAccess": "SourceElement",
                                "replicaType": "Clone",
                                "replicaRole": "Target",
                                "replica": "4e20b351-6732-ab36-121e-887efa325e4c",
                        }
                ],
                "collections": [
                        {
                                "name": "Endpoints",
                                "type": "Endpoints"
                        }
                ]
                "oem": {}
        },
        "id": 987
}
```

5.98 addVolume

The `"addVolume"` command creates a new volume within the storage infrastructure hierarchy. Optionally it may create a newly populated volume with the image from a existing volume (e.g. cloning operation). User may request a specific `capacityBytes` and let the PSME select most appropriate capacity sources or provide a `capacitySources` array to select existing pools, volumes, or drives.

5.98.1 Request

Table 178. addVolume Request

Result	Type	Mandatory	Description
`volumeType`	String	No	Volume type: `"RawDevice"`, `"NonRedundant"`, `"Mirrored"`, `"StripedWithParity"`, `"SpannedMirrors"`, `"SpannedStripesWithParity"`
`capacityBytes`	Number	Yes	Requested volume capacity

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>capacitySources</td><td>Array: Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>consumedBytes</td><td>Number</td><td>No</td><td>The number of bytes currently allocated by the storage system in this data store for this data type.</td></tr>
<tr><td>allocatedBytes</td><td>Number</td><td>No</td><td>The number of bytes currently allocated by the storage system in this data store for this data type.</td></tr>
<tr><td>guaranteedBytes</td><td>Number</td><td>No</td><td>The number of bytes the storage system guarantees can be allocated in this data store for this data type.</td></tr>
<tr><td>provisionedBytes</td><td>Number</td><td>No</td><td>The maximum number of bytes that can be allocated in this data store for this data type.</td></tr>
<tr><td>providingDrives</td><td>Array: String</td><td>No</td><td>The drive or drives UUID that provide this space.</td></tr>
<tr><td>providingVolumes</td><td>Array: String</td><td>No</td><td>The volume or volumes UUID that provide this space.</td></tr>
<tr><td>providingPools</td><td>Array: String</td><td>No</td><td>The pool or pools that provide this space.</td></tr>
</table>
</td></tr>
<tr><td>accessCapabilities</td><td>Array: String</td><td>Yes</td><td>List of supported access capabilities. Allowed values: "Read", "Write", "WriteOnce", "Append", "Streaming".</td></tr>
<tr><td>bootable</td><td>Boolean</td><td>No</td><td>Boolean whether the volume is bootable or not.</td></tr>
<tr><td>replicaInfos</td><td>Array: Object</td><td>No</td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>replicaReadOnlyAccess</td><td>String</td><td>Yes</td><td>This property specifies whether the source, the target, or both elements are read-only to the host. Values: "SourceElement", "ReplicaElement", "Both".</td></tr>
<tr><td>replicaType</td><td>String</td><td>Yes</td><td>ReplicaType describes the intended outcome of the replication. Values: "Mirror", "Snapshot", "Clone", "TokenizedClone"</td></tr>
<tr><td>replicaRole</td><td>String</td><td>Yes</td><td>The source or target role of this replica. Values: "Source", "Target"</td></tr>
<tr><td>replica</td><td>String</td><td>Yes</td><td>The resource UUID that is the source of this replica.</td></tr>
</table>
</td></tr>
</table>

Result	Type	Mandatory	Description
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "addVolume",
        "params": {
                "volumeType": <{"RawDevice", "NonRedundant", "Mirrored",
"StripedWithParity", "SpannedMirrors", "SpannedStripesWithParity"}>,
                "capacityBytes": <number>,
                "capacitySources": [
                        {
                                allocatedBytes: <Number>,
                                providingDrives: [
                                        <String>, …
                                ],
                                providingVolumes: [
                                        <String>, …
                                ],
                                providingPools: [
                                        <String>, …
                                ]
                        }
                ],
                "accessCapabilities": [ <{"Read", "Write"}> ],
                "bootable": <Boolean>,
                "replicaInfos": [
                        {
                                "replicaReadOnlyAccess": <{"SourceElement",
"ReplicaElement", "Both"}>,
                                "replicaType": <{"Mirror", "Snapshot", "Clone",
"TokenizedClone"}>,
                                "replicaRole": <{"Source", "Target"}>,
                                "replica": <String>,
                        },
                        …
                ],
        "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addVolume",
        "params": {
                "volumeType": "RawDevice",
                "capacityBytes": 1099511627776,
                "capacitySources": [
                        {
                                providingPools: [
                                        "b2fe60a7-33a8-63bb-da16-
3e7b1a835ee5"
                                ]
                        },
                        {
                                providingPools: [
```

```
                                                        "cca5b252-112a-78e3-23a1-
5bec7252311f"
                                ]
                        }
                ],
                "accessCapabilities" : [ "Read", "Write" ],
                "replicaInfos": [
                        {
                                "replicaReadOnlyAccess": "SourceElement",
                                "replicaType": "Clone",
                                "replicaRole": "Target",
                                "replica": "4e20b351-6732-ab36-121e-887efa325e4c",
                        }
                ],
                "oem": {}
        },
        "id": 987
}
```

5.98.2 Response

Table 179 addVolume Response

Parameters	Type	Mandatory	Description
volume	String	Yes	Created volume UUID
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "volume": <string>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "volume": "123e4567-e89b-12d3-a456-426655440000",
                "oem": {}
        },
        "id": 987
}
```

5.99 deleteVolume

The "deleteVolume" command destroys an existing volume created using "addVolume" command.

5.99.1 Request

Table 180. deleteVolume Request

Parameters	Type	Mandatory	Description
volume	String	Yes	Managed volume UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "deleteVolume",
        "params": {
                "volume": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "deleteVolume",
        "params": {
                "volume": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 987
}
```

5.99.2 Response

Table 181. deleteVolume Response

Result	Type	Mandatory	Description
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "oem": {}
        },
        "id": 987
}
```

5.100 Set Volume Attributes

The "setComponentAttributes" described in Section 5.5, setComponentAttributes allows for configuration of the Volume Attributes listed in the following table.

Table 182. Configurable Volume Attributes

Attribute	Type	Description
bootable	Boolean	True if the drive is bootable volume. Otherwise false.
allocatedBytes	Number	The number of bytes that shall be allocated by the storage system in this data store for this data type
oem	Object	OEM specific data

5.101 getMetricDefinitionsCollection

The "getMetricDefinitionsCollection" returns a list of sensor definitions available on the platform.

5.101.1 Request

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getMetricDefinitionsCollection",
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getMetricDefinitionsCollection",
        "id": 987
}
```

5.101.2 Response

The GAM replies with the following response in a successful case:

Table 183. getMetricDefinitionsCollection Response

Parameters	Type	Mandatory	Description
Result	Array: Object	Yes	**Member** / **Type** / **Mandatory** / **Description** metricDefinition / String / Yes / UUID of the sensor.

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": [
                {
                        "metricDefinition": <string>
                },
                ...
        ],
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": [
                {
                        "metricDefinition": "123e4567-e89b-12d3-a456-426655440000",
                },
                {
                        "metricDefinition": "863e4567-e87b-64d3-a489-987656540000",
                }
        ],
        "id": 987
}
```

5.102 getMetricDefinitionInfo

The "getMetricDefinitionInfo" command returns information about a specific sensor on the platform.

5.102.1 Request

Table 184. getMetricDefinitionInfo Definition

Parameters	Type	Mandatory	Description
metricDefinition	String	No	Metric definition UUID.

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getMetricDefinitionInfo",
        "params": {
                "metricDefinition": <string>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getMetricDefinitionInfo",
        "params": {
                "metricDefinition": "1535fd67-e14e-12d3-a456-42e230140000"
        },
        "id": 1
}
```

5.102.2 Response

Table 185. getMetricDefinitionInfo Response

Result	Type	Mandatory	Description
id	String	Yes	Identification of the sensor
name	String	Yes	Name of the sensor
description	String	No	Description of the sensor
accuracy	Number	No	Estimated error percent of measured vs. actual values
calculable	String	No	Calculability of the metric. Allowed values:

<table>
<tr><th>Result</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>

- "NonCalculable" – arithmetic on corresponding metric values makes no sense
- "Summable" – it is reasonable to sum corresponding metric values over many instances
- "NonSummable" – it does not make sense to sum corresponding metric values over many instances

</td></tr>
<tr><td>calculationAlgorithm</td><td>String</td><td>No</td><td>The calculation which is performed on source metric to obtain the metric being defined. Allowed values:

- "AverageOverInterval" – metric is calculated as the average over a sliding time interval
- "MaximumDuringInterval" – metric is calculated as the maximum value of a source metric during a time interval

</td></tr>
<tr><td>calculationParameters</td><td>Array: String</td><td>No</td><td>Specifies the resource properties (metric) which are characterized by this definition.
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>sourceMetric</td><td>Number</td><td>Yes</td><td>This property reference the metric property used in the calculation</td></tr>
<tr><td>resultMetric</td><td>String</td><td>Yes</td><td>This property shall reference the metric property which the results of the calculation are placed.</td></tr>
</table>
</td></tr>
<tr><td>calculationTimeInterval</td><td>String</td><td>No</td><td>The time interval over which a calculated metric algorithm is performed.</td></tr>
<tr><td>calibration</td><td>Number</td><td>No</td><td>Calibration offset added to the reading to obtain accurate values</td></tr>
<tr><td>dataType</td><td>String</td><td>Yes</td><td>The data type of the related metric nalues. Allowed values are: "Binary", "Boolean", "Byte", "Date", "DateTimeOffset", "Duration", "TimeOfDay", "Decimal", "Double", "Single", "Int16", "Int32", "Int64", "String", "SByte" and other. Please see current metadata.</td></tr>
<tr><td>discreteValues</td><td>Array: String</td><td>No</td><td>The array property specifies possible values of a discrete metric.</td></tr>
<tr><td>sensorType</td><td>String</td><td>Yes</td><td>The value of this property shall describe the type of sensor (e.g., Temperature, Voltage, Current, Fan, etc.)</td></tr>
<tr><td>metricType</td><td>String</td><td>Yes</td><td>Type of sensor. Allowed: "Counter", Gauge", "Numeric", "Discrete".</td></tr>
<tr><td>calculationPrecision</td><td>Number</td><td>No</td><td>This property specifies the precision of a calculated metric (calculated metric shall be aligned to a value specified by This property.</td></tr>
<tr><td>discreteMetricType</td><td>String</td><td>No</td><td>This array property specifies possible values of a discrete metric. Values: "Single", "Multiple".</td></tr>
<tr><td>implementation</td><td>String</td><td>Yes</td><td>Specifies how the sensor is implemented. Possible values:

- "PhysicalSensor" – the metric is implemented as a physical sensor

</td></tr>
</table>

Result	Type	Mandatory	Description
			• "Synthesized" - The metric is implemented by applying a calculation on the readings from one or more physical sensors. • "Calculated" - The metric is implemented by applying a calculation on one more metric properties. The CalculationAlgorithm property specifies the algorithm. • "DigitalMeter" - The metric is implemented as a digital meter.
isLinear	Boolean	No	Indicates linear or non-linear values
maxReadingRange	Number	Yes	The maximum value for reading
minReadingRange	Number	Yes	The minimum value for reading
timestampAccuracy	String	No	The accuracy of the timestamp. The format of the property shall conformant to ISO 8601 duration format
wildcards	Array: String	No	Wildcards used to replace values in MetricProperties and CalculationParameters metric property arrays:

Attribute	Type	Mandatory	Description
name	String	Yes	This property shall contain a name for a wildcard.
values	Array: String	Yes	This property shall contain a list of values the server shall substitute. Each not empty value shall be substituted for the wildcard.

sensingInterval	String	Yes	The time interval between when a metric or sensor reading is updated. The value shall be in the format specified in ISO 8601.
physicalContext	String	Yes	Physical asset the sensor is related to. (e.g., Backplane, SystemBoard)
units	String	Yes*	Units of measure for this metric (consistent with Unified Code for Units of Measure as defined at *http://unitsofmeasure.org/ucum.html*)
precision	Number	Yes*	The precision of the readings for the sensor. Only for numeric sensors.
oem	Object	No	OEM specific data

Note: There are two types of sensors (metricType): Numeric and Discrete. For Numeric sensors the following parameters are valid: Units, minReadingRange, maxReadingRange, precision, and calibration. For Discrete sensors only, discreteValues (among marked) are valid. The response should include only the appropriate fields depending on the metric type.

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                  "id": <String>,
                  "name" <String>,
                  "description": <String>,
                  "datatype": <String>
                  "sensorType": <String>,
                  "metricType": <String>,
```

```
                "implementation": <String>,
                "sensingInterval": <Number>,
                "physicalContext": <String>,
                "discreteValues": <Array of Strings>,
                "units": <String>,
                "minReadingRange": <Number>,
                "maxReadingRange": <Number>,
                "precision": <Number>,
                "calibration": {
                        "offset": <Number>
                },
                "components": [
                        {
                                "component": <String>,
                                "type": <String>
                        },
                        …
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example (Numeric):

```
{
        "jsonrpc": "2.0",
        "result": {
                "id": "TEMP1",
                "name" "Temperature Sensor Definition",
                "description": "This is main CPU temperature sensor definition",
                "datatype": "Double"
                "sensorType": "Temperature",
                "metricType": "Numeric",
                "implementation": "Physical",
                "sensingInterval": 1000,
                "physicalContext": "CPU",
                "units": "°C",
                "minReadingRange": 0,
                "maxReadingRange": 80,
                "precision": 0.1,
                "calibration": {
                        "offset": 1
                },
                "components": [
                        {
                                "component": "834e731a-94c1-012b-22fe-04892b8a7d30",
                                "type": "Processors"
                        },
                        {
                                "component": "4ec72611-abe3-5fe3-03cc-a034b27810ea",
                                "type": "Processors"
                        }
                ],
                "oem": {}
        },
        "id": 1
}
```

Example (Discrete):

```
{
        "jsonrpc": "2.0",
        "result": {
                "id": "CPU1HEALTH",
                "name" "CPU1 IPMI Health Sensor",
                "description": "This is CPU1 Health Sensor Definition (using IPMI)",
                "datatype": "String"
                "sensorType": "Processor",
                "metricType": "Discrete",
                "implementation": "Physical",
                "sensingInterval": 1000,
                "physicalContext": "CPU",
                "discreteValues": [
                        "OK",
                        "Internal Error",
                        "Thermal Trip",
                        "FRB1 BIST Failure",
                        "Terminator Presence Detected",
                        "Processor Throttled",
                        "Machine Check Exception",
                        "Correctable Machine Check Error"
                ],
                "components": [
                        {
                                "component": "834e731a-94c1-012b-22fe-04892b8a7d30",
                                "type": "Processors"
                        },
                        {
                                "component": "4ec72611-abe3-5fe3-03cc-a034b27810ea",
                                "type": "Processors"
                        }
                ],
                "oem": {}
        },
        "id": 2
}
```

5.103 Set Sensor Definition Attributes

The `"setComponentAttributes"` described in Section 5.5, setComponentAttributes, allows configuration of the Sensor Definition Attributes listed in the following table.

Table 186. Configurable Ethernet Switch Port Attributes

Attribute	Type	Description
`sensingInterval`	Object	See Get Sensor Definition
`oem`	Object	OEM specific data

5.104 getMetrics

Read sensor values for a particular sensor its parent or component type sensor belongs to.

5.104.1 Request

Table 187. getMetrics Request

Parameters	Type	Mandatory	Description
metric	String	No	Requested sensor UUID
metricDefinition	String	No	Metric definition UUID
component	String	No	Sensor's parent UUID
name	String	No	Sensor's name

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getMetrics",
        "params": {
                "metric": <String>,
                "metricDefinition": <String>,
                "component": <String>,
                "name": <String>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getMetrics",
        "params": {
                "metric": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 1
}
```

5.104.2 Response

Table 188. getMetrics Response

Result	Type	Mandatory	Description
metrics	Array: Object	Yes	See nested table below
oem	Object	No	OEM specific data

Attribute	Type	Mandatory	Description
component	String	Yes	Metric's component UUID
componentType	String	Yes	Metric's component type
name	String	No	Name of the metric
description	String	No	Optional description of the component
timestamp	String	Yes	Timestamp of last metric read
uuid	String	Yes	Metric UUID
metricDefinition	String	Yes	Sensor definition the metric refers to
value	Object	Yes	Value of the sensor. Value type depends on sensor.

Serialization:

```
{
        "jsonrpc": "2.0",
        "result": {
                 "metrics": [
                          {
            "component": "1cfb03c0-41f9-11e7-8072-54ab3a36b489",
            "componentType": "Processor",
            "name": "consumedPower",
            "timestamp": 1495792384,
            "uuid": "1cfc11d4-41f9-11e7-9b91-54ab3a36b489",
            "metricDefinition": "1cfc11d4-51f9-11e7-9b91-54ab3a36b423",
            "value": 13
         },
         {
            "component": "1cfb03c0-41f9-11e7-8072-54ab3a36b489",
            "componentType": "Processor",
            "name": "cpu health",
            "timestamp": 1495792384,
            "uuid": "1cfc47a8-41f9-11e7-9b5b-54ab3a36b489",
            "metricDefinition": "447811d4-3216-11e7-28be-54ab3a123498",
            "value": "InternalError"
         }
                 ],
                 "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                 "metrics": [
                          {
            "component": "1cfb03c0-41f9-11e7-8072-54ab3a36b489",
            "componentType": "Processor",
            "name": "consumedPower",
            "timestamp": 1495792384,
            "uuid": "1cfc11d4-41f9-11e7-9b91-54ab3a36b489",
            "metricDefinition": "1cfc11d4-51f9-11e7-9b91-54ab3a36b423",
            "value": 13
         },
         {
            "component": "1cfb03c0-41f9-11e7-8072-54ab3a36b489",
            "componentType": "Processor",
            "name": "cpu health",
            "timestamp": 1495792384,
            "uuid": "1cfc47a8-41f9-11e7-9b5b-54ab3a36b489",
            "metricDefinition": "447811d4-3216-11e7-28be-54ab3a123498",
            "value": "InternalError"
         }
                 ],
                 "oem": {}
        },
        "id": 1
}
```

5.104.3 Example sensors by asset type

Example sensors as defined in the Redfish metadata, refer to Table 2.

Table 189. Example Sensors by Asset Type

Asset	Sensor	Example value
CPU	AverageFrequencyMHz	3014
CPU	ThrottlingCelsius	19
CPU	TemperatureCelsius	73
CPU	Health	["RFB1 BIST Failure", "Processor Throttled"]
Memory	TemperatureCelsius	46
Memory	Health	["OK"]
System	ProcessorBandwidthPercent	17
System	ProcessorPowerWatt	153
System	MemoryBandwidthPercent	23
System	MemoryThrottledCyclesPercent	16
System	MemoryPowerWatt	234
System	IOBandwidthGBps	4
System	Health	["OK"]
Switch Port	Health	"OK"

Note: The list of available sensors depends on the hardware platform. Refer to the specific hardware requirements specification for the platform used.

§

6.0 GAMI Commands Support Requirements

This specification defines types of GAM modules:

- `"Compute"` – a module managing the compute assets within the drawer. i.e., system, processors, and memory
- `"Network"` – a module managing the network assets within the drawer. i.e., switches and switch ports
- `"StorageServices"` – a module managing the storage services node assets. i.e., physical drives, logical drives, remote targets
- `"Chassis"` – a module managing the chassis assets within the drawer. i.e., like drawer chassis, local power and thermal
- `"PCIeFabric"` – a module managing the PCIe* assets. i.e., PCIe switch, PCIe ports, PCIe device

The single GAM module can provide management for one or more asset types. Management capability is reported during the registration process.

The following table provides information about the GAM commands that support requirements for different types of GAM modules.

Table 190. GAMI Commands Support Requirements

Command	Compute	Storage Services (iSCSI, NVMe-oF)	Network	Chassis	PCIe* Fabric
GAM Module Registration	Mandatory	Mandatory	Mandatory	Mandatory	Mandatory
Get Managers Collection	Mandatory	Mandatory	Mandatory	Mandatory	Mandatory
Get Manager Info	Mandatory	Mandatory	Mandatory	Mandatory	Mandatory
Get Tasks Collection	Mandatory	Mandatory	Mandatory	Mandatory	Mandatory
Get Task Info	Mandatory	Mandatory	Mandatory	Mandatory	Mandatory
Delete Task	Mandatory	Mandatory	Mandatory	Mandatory	Mandatory
Get Collection	Mandatory	Mandatory	Mandatory	Mandatory	Mandatory
Component Notification	Mandatory	Mandatory	Mandatory	Mandatory	Mandatory
Get Computer System Info	Mandatory				Optional
Set Computer System Attributes	Mandatory				Optional
Get Processor Info	Mandatory				Optional
Get Memory Info	Mandatory				
Get Storage Controller Info	Mandatory				
Get Storage Subsystem Info	Mandatory				Optional
Set Storage Subsystem Attributes	Optional				Optional
Get Drive Info	Optional			Mandatory	Optional
Set Drive Attributes				Optional	
Get Network Interface Info	Mandatory				
Set Network Interface Attributes	Optional				
Get Trusted Module Info	Optional				
Set Trusted Module Attributes	Optional				
Get Switch Info					Mandatory
Get Port Info					Mandatory
Set Port Attributes					Mandatory
Get Endpoint Info		Mandatory			Mandatory
Add Endpoint		Mandatory			Mandatory
Delete Endpoint		Mandatory			Mandatory

Command	Compute	Storage Services (iSCSI, NVMe-oF)	Network	Chassis	PCIe* Fabric
Set Endpoint Attributes		Optional			Optional
Get PCIe Device Info	Optional				Mandatory
Get PCIe Function Info	Optional				Mandatory
Get Zone Info		Mandatory			Mandatory
Add Zone		Mandatory			Mandatory
Delete Zone		Mandatory			Mandatory
Add Zone Endpoints		Mandatory			Mandatory
Delete Zone Endpoints		Mandatory			Mandatory
Get Ethernet Switch Info			Mandatory		
Get Ethernet Switch Port Info			Mandatory		
Set Ethernet Switch Port Attributes			Mandatory		
Add Ethernet Switch Port			Optional		
Delete Ethernet Switch Port			Optional		
Add Ethernet Switch Port Members			Optional		
Delete Ethernet Switch Port Members			Optional		
Get Remote Ethernet Switch Info			Optional		
Get VLAN Info			Optional[1]		
Set VLAN Attributes			Optional		
Add VLAN			Optional[1]		
Delete VLAN			Optional[1]		
Get Port VLAN Info			Mandatory		
Set Port VLAN Attributes			Optional		
Add Port VLAN			Mandatory		
Delete Port VLAN			Mandatory		
Get Port Static MAC Info			Mandatory		
Set Port Static MAC Attributes			Optional		
Add Port Static MAC			Mandatory		
Delete Port Static MAC			Mandatory		
Get ACL Info			Optional		
Add ACL			Optional		
Delete ACL			Optional		
Add ACL Port			Optional		
Delete ACL Port			Optional		
Get ACL Rule Info			Optional		
Add ACL Rule			Optional		
Delete ACL Rule			Optional		
Set ACL Rule Attributes			Optional		
Get Chassis Info				Mandatory	
Get Power Zone Info				Optional	

[1] *This command is optional and may not be implemented if VLANs are not explicitly created on the switch (VLANs are created indirectly during creation of the port VLAN).*

Command	Compute	Storage Services (iSCSI, NVMe-oF)	Network	Chassis	PCIe* Fabric
Set Power Zone Attributes				Optional	
Get PSU Info				Optional[2]	
Set PSU Attributes				Optional	
Get Thermal Zone Info				Optional	
Set Thermal Zone Attributes				Optional	
Get Fan Info				Optional[3]	
Set Fan Attributes				Optional	
Get Authorization Certificate				Optional	
Get Storage Service Info		Mandatory			
Get Volume Info		Mandatory			
Add Volume		Mandatory			
Delete Volume		Mandatory			
Get Storage Pool Info		Mandatory			
Add Storage Pool		Optional			
Delete Storage Pool		Optional			
Get Metric Definitions Collection	Optional	Optional	Optional	Optional	Optional
Get Metric Definition Info	Optional	Optional	Optional	Optional	Optional

§

[2] *This command is mandatory if Get Power Zone Info command is implemented and contains PSU collection(s)*
[3] *This command is mandatory if Get Thermal Zone Info command is implemented and contains Fan collection(s)*