Intel® Rack Scale Design Generic Assets Management Interface

API Specification
Software Version 2.1.3

April 2017

Revision 002

Document Number: 335451-002

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at *www.intel.com*.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and noninfringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Copies of documents that have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting *http://www.intel.com/design/literature.htm*.

Intel and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

* Other names and brands may be claimed as the property of others.

Copyright © 2017 Intel Corporation. All rights reserved.

Contents

1 Introduction 13
1.1 Scope 13
1.2 Intended audience 13
1.3 Terminology 13
1.4 References 14
1.5 Notes and Symbol Convention 14
2 Generic Asset Management Interface 15
2.1 GAMI API architecture and design principles 15
2.2 GAMI API synchronous and asynchronous operations 16
3 GAMI API Error Handling 20
3.1 API error response 20
3.1.1 Example error JSON object 20
3.2 API error codes 20
3.2.1 General error codes 20
4 GAMI Data Model 22
4.1 Computer System management module 22
4.2 Storage Services management module 22
4.3 Network management model 23
4.4 Chassis management model 23
4.5 PCIe Fabric management model 23
4.6 Subcomponent collections 24
4.7 Mandatory and Optional properties 24
5 GAMI API Definition 26
5.1 GAM Module Registration 26
5.1.1 Request 27
5.1.2 Response 28
5.2 Heartbeat 28
5.2.1 Request 28
5.2.2 Response 29
5.3 Get Managers Collection 29
5.3.1 Request 29
5.3.2 Response 30
5.4 Get Manager Info 30
5.4.1 Request 30
5.4.2 Response 31
5.5 Get Tasks Collection 35
5.5.1 Request 35
5.5.2 Response 36
5.6 Get Task Info 36
5.6.1 Request 36
5.6.2 Response 37
5.7 Delete Task 38
5.7.1 Request 38
5.7.2 Response 39
5.8 Get Task Result 39
5.8.1 Request 39

5.8.2 Response 40
5.9 Get Collection 40
5.9.1 Request 41
5.9.2 Response 41
5.10 Component Notification 42
5.10.1 Notification 42
5.11 Set Component Attributes 43
5.11.1 Request 43
5.11.2 Response 44
5.12 Get Computer System Info 44
5.12.1 Request 44
5.12.2 Response 45
5.13 Set Computer System Attributes 49
5.14 Get Processor Info 49
5.14.1 Request 49
5.14.2 Response 50
5.15 Get Memory Info 52
5.15.1 Request 52
5.15.2 Response 53
5.16 Get Storage Subsystem Info 57
5.16.1 Request 57
5.16.2 Response 57
5.17 Set Storage Subsystem Attributes 58
5.18 Get Storage Controller Info 59
5.18.1 Request 59
5.18.2 Response 59
5.19 Get Drive Info 62
5.19.1 Request 62
5.19.2 Response 62
5.20 Set Drive Attributes 66
5.21 Get Network Interface Info 66
5.21.1 Request 66
5.21.2 Response 67
5.22 Set Network Interface Attributes 70
5.23 Get Network Device Info 70
5.23.1 Request 70
5.23.2 Response 71
5.24 Get Network Device Function Info 72
5.24.1 Request 72
5.24.2 Response 72
5.25 Add Network Device Function 75
5.25.1 Request 75
5.25.2 Response 77
5.26 Set Network Device Function Attributes 77
5.27 Delete Network Device Function 79
5.27.1 Request 79
5.27.2 Response 79
5.28 Get Fabric Info 80
5.28.1 Request 80
5.28.2 Response 80
5.29 Get Switch Info 82

5.29.1 Request 82
5.29.2 Response 82
5.30 Set Switch Attributes 84
5.31 Get Port Info 85
5.31.1 Request 85
5.31.2 Response 85
5.32 Set Port Attributes 87
5.33 Get Endpoint Info 87
5.33.1 Request 87
5.33.2 Response 88
5.34 Get PCIe Device Info 90
5.34.1 Request 90
5.34.2 Response 91
5.35 Set PCIe Device Attributes 93
5.36 Get PCIe Function Info 93
5.36.1 Request 93
5.36.2 Response 93
5.37 Get Zone Info 95
5.37.1 Request 95
5.37.2 Response 96
5.38 Add Zone 97
5.38.1 Request 97
5.38.2 Response 98
5.39 Delete Zone 98
5.39.1 Request 98
5.39.2 Response 99
5.40 Add Zone Endpoints 99
5.40.1 Request 99
5.40.2 Response 100
5.41 Delete Zone Endpoints 100
5.41.1 Request 100
5.41.2 Response 101
5.42 Get Ethernet Switch Info 101
5.42.1 Request 102
5.42.2 Response 102
5.43 Get Ethernet Switch Port Info 104
5.43.1 Request 105
5.43.2 Response 105
5.44 Set Ethernet Switch Port Attributes 109
5.45 Add Ethernet Switch Port 109
5.45.1 Request 109
5.45.2 Response 110
5.46 Delete Ethernet Switch Port 111
5.46.1 Request 111
5.46.2 Response 111
5.47 Add Ethernet Switch Port Members 112
5.47.1 Request 112
5.47.2 Response 113
5.48 Delete Ethernet Switch Port Members 113
5.48.1 Request 113
5.48.2 Response 114

5.49 Get Remote Ethernet Switch Info 114
5.49.1 Request 114
5.49.2 Response 115
5.50 Get VLAN Info 116
5.50.1 Request 116
5.50.2 Response 117
5.51 Set VLAN Attributes 118
5.52 Add VLAN 118
5.52.1 Request 118
5.52.2 Response 119
5.53 Delete VLAN 119
5.53.1 Request 119
5.53.2 Response 120
5.54 Get Port VLAN Info 120
5.54.1 Request 120
5.54.2 Response 121
5.55 Set Port VLAN Attributes 122
5.56 Add Port VLAN 122
5.56.1 Request 122
5.56.2 Response 123
5.57 Delete Port VLAN 123
5.57.1 Request 124
5.57.2 Response 124
5.58 Get Port Static MAC Info 125
5.58.1 Request 125
5.58.2 Response 125
5.59 Set Port Static MAC Attributes 126
5.60 Add Port Static MAC 126
5.60.1 Request 126
5.60.2 Response 127
5.61 Delete Port Static MAC 128
5.61.1 Request 128
5.61.2 Response 128
5.62 Get ACL Info 129
5.62.1 Request 129
5.62.2 Response 129
5.63 Add ACL 131
5.63.1 Request 131
5.63.2 Response 131
5.64 Delete ACL 132
5.64.1 Request 132
5.64.2 Response 133
5.65 Add ACL Port 133
5.65.1 Request 133
5.65.2 Response 134
5.66 Delete ACL Port 134
5.66.1 Request 134
5.66.2 Response 135
5.67 Get ACL Rule Info 135
5.67.1 Request 136
5.67.2 Response 136

5.68 Add ACL Rule 139
5.68.1 Request 139
5.68.2 Response 141
5.69 Delete ACL Rule 142
5.69.1 Request 142
5.69.2 Response 143
5.70 Set ACL Rule Attributes 143
5.71 Get Chassis Info 144
5.71.1 Request 144
5.71.2 Response 144
5.72 Set Chassis Attributes 147
5.73 Get Power Zone Info 147
5.73.1 Request 147
5.73.2 Response 148
5.74 Set Power Zone Attributes 149
5.75 Get PSU Info 149
5.75.1 Request 149
5.75.2 Response 150
5.76 Set PSU Attributes 151
5.77 Get Thermal Zone Info 151
5.77.1 Request 152
5.77.2 Response 152
5.78 Get Fan Info 153
5.78.1 Request 153
5.78.2 Response 154
5.79 Set Fan Attributes 155
5.80 Get Authorization Certificate 155
5.80.1 Request 156
5.80.2 Response 156
5.81 Get Storage Services Info 157
5.81.1 Request 157
5.81.2 Response 157
5.82 Get Physical Drive Info 159
5.82.1 Request 159
5.82.2 Response 159
5.83 Get Logical Drive Info 161
5.83.1 Request 161
5.83.2 Response 161
5.84 Add Logical Drive 164
5.84.1 Request 164
5.84.2 Response 165
5.85 Delete Logical Drive 166
5.85.1 Request 166
5.85.2 Response 166
5.86 Set Logical Drive Attributes 167
5.87 Get iSCSI Target Info 167
5.87.1 Request 167
5.87.2 Response 167
5.88 Set iSCSI Target Attributes 169
5.89 Add iSCSI Target 169
5.89.1 Request 169

5.89.2 Response 170
5.90 Delete iSCSI Target 171
5.90.1 Request 171
5.90.2 Response 171
6 GAMI Commands Support Requirements 173

Figures

Figure 1 Generic Assets Management Interface in the PSME architecture 15
Figure 2 Sample asynchronous communication 17
Figure 3 Computer System management data model 22
Figure 4 Storage Services management data model 22
Figure 5 Network management data model 23
Figure 6 Chassis management data model 23
Figure 7 PCIe Fabric management data model 23
Figure 8 PSME Generic Asset Management Interface API communication overview 26

Tables

Table 1 Terminology 13
Table 2 General error codes 20
Table 3 GAMI subcomponent collection definition 24
Table 4 GAM Module Registration request 27
Table 5 GAM Module Registration response 28
Table 6 Heartbeat command request 28
Table 7 Heartbeat command response 29
Table 8 Get Managers Collection response 30
Table 9 Get Manager Info request 30
Table 10 Get Manager Info response 31
Table 11 Get Tasks Collection response 36
Table 12 Get Task Info request 36
Table 13 Get Task Info response 37
Table 14 Delete Task request 38
Table 15 Delete Task response 39
Table 16 Get Task Result request 39
Table 17 Get Task Result response 40
Table 18 Get Collection request 41
Table 19 Get Collection response 41
Table 20 Component Notification 42
Table 21 Set Component Attributes request 43
Table 22 Set Component Attributes response 44
Table 23 Get Computer System Info request 44
Table 24 Get Computer System Info response 45
Table 25 Configurable Computer System Attributes 49
Table 26 Get Processor Info request 49
Table 27 Get Processor Info response 50
Table 28 Get Memory Info request 52
Table 29 Get Memory Info response 53
Table 30 Get Storage Subsystem Info request 57

Table 31 Get Storage Subsystem Info response 57
Table 32 Configurable Storage Subsystem Attributes 59
Table 33 Get Storage Controller Info request 59
Table 34 Get Storage Controller Info response 59
Table 35 Get Drive Info request 62
Table 36 Get Drive Info response 62
Table 37 Set Drive Attributes 66
Table 38 Get Network Interface Info request 66
Table 39 Get Network Interface Info response 67
Table 40 Configurable Network Interface Attributes 70
Table 41 Get Network Interface Info request 70
Table 42 Get Network Device Info response 71
Table 43 Get Network Device Function Info request 72
Table 44 Get Network Device Info response 72
Table 45 Add Network Device Function request 75
Table 46 Add Network Device Function response 77
Table 47 Configurable Network Device Function Attributes 77
Table 48 Delete Network Device Function request 79
Table 49 Delete Network Device Function response 79
Table 50 Get Fabric Info request 80
Table 51 Get Fabric Info response 80
Table 52 Get Switch Info request 82
Table 53 Get Switch Info response 82
Table 54 Configurable Switch Attributes 84
Table 55 Get Port Info request 85
Table 56 Get Port Info response 85
Table 57 Configurable Port Attributes 87
Table 58 Get Endpoint Info request 87
Table 59 Get Endpoint Info response 88
Table 60 Get PCIe Device Info request 90
Table 61 Get PCIe Device Info response 91
Table 62 Set PCIe Device Attributes 93
Table 63 Get PCIe Function Info request 93
Table 64 Get PCIe Function Info response 93
Table 65 Get Zone Info request 95
Table 66 Get Zone Info response 96
Table 67 Add Zone request 97
Table 68 Add Zone response 98
Table 69 Delete Zone request 98
Table 70 Delete Zone response 99
Table 71 Add Zone Endpoints request 99
Table 72 Add Zone Endpoints response 100
Table 73 Delete Zone Endpoints request 100
Table 74 Delete Zone Endpoints response 101
Table 75 Get Ethernet Switch Info request 102
Table 76 Get Ethernet Switch Info response 102
Table 77 Get Ethernet Switch Port Info request 105
Table 78 Get Ethernet Switch Port Info response 105
Table 79 Configurable Ethernet Switch Port Attributes 109
Table 80 Add Ethernet Switch Port request 109

Table 81 Add Ethernet Switch Port response 110
Table 82 Delete Ethernet Switch Port request 111
Table 83 Delete Ethernet Switch Port response 111
Table 84 Add Ethernet Switch Port Members request 112
Table 85 Add Ethernet Switch Port Members response 113
Table 86 Delete Ethernet Switch Port Members request 113
Table 87 Delete Ethernet Switch Port Members response 114
Table 88 Get Remote Ethernet Switch Info request 114
Table 89 Get Remote Ethernet Switch Info response 115
Table 90 Get VLAN Info request 116
Table 91 Get VLAN Info request 117
Table 92 Configurable VLAN Attributes 118
Table 93 Add VLAN request 118
Table 94 Add VLAN response 119
Table 95 Delete VLAN request 119
Table 96 Delete VLAN response 120
Table 97 Get Port VLAN Info request 120
Table 98 Get Port VLAN Info response 121
Table 99 Configurable Port VLAN Attributes 122
Table 100 Add Port VLAN request 122
Table 101 Add Port VLAN response 123
Table 102 Delete Port VLAN request 124
Table 103 Delete Port VLAN response 124
Table 104 Get Port Static MAC Info request 125
Table 105 Get Port Static MAC Info response 125
Table 106 Configurable Port Static MAC Attributes 126
Table 107 Add Port Static MAC request 126
Table 108 Add Port Static MAC response 127
Table 109 Delete Port Static MAC request 128
Table 110 Delete Port Static MAC response 128
Table 111 Get ACL Info request 129
Table 112 Get ACL Info response 129
Table 113 Add ACL request 131
Table 114 Add ACL response 131
Table 115 Delete ACL request 132
Table 116 Delete ACL response 133
Table 117 Add ACL Port request 133
Table 118 Add ACL Port response 134
Table 119 Delete ACL Port request 134
Table 120 Delete ACL Port response 135
Table 121 Get ACL Rule Info request 136
Table 122 Get ACL Rule Info response 136
Table 123 Add ACL Rule request 139
Table 124 Add ACL Rule response 141
Table 125 Delete ACL Rule request 142
Table 126 Delete ACL Rule response 143
Table 127 Configurable ACL Rule Attributes 143
Table 128 Get Chassis Info request 144
Table 129 Get Chassis Info response 144
Table 130 Set Chassis Attributes 147

Table 131 Get Power Zone Info request 147
Table 132 Get Power Zone Info response 148
Table 133 Configurable Power Zone Attributes 149
Table 134 Get PSU Info request 149
Table 135 Get PSU Info response 150
Table 136 Configurable PSU Attributes 151
Table 137 Get Thermal Zone Info request 152
Table 138 Get Thermal Zone Info response 152
Table 139 Get Fan Info request 153
Table 140 Get Fan Info response 154
Table 141 Configurable Fan Attributes 155
Table 142 Get Authorization Certificate request 156
Table 143 Get Authorization Certificate response 156
Table 144 Get Storage Services Info request 157
Table 145 Get Storage Services Info response 157
Table 146 Get Physical Drive Info request 159
Table 147 Get Physical Drive Info response 159
Table 148 Get Logical Drive Info request 161
Table 149 Get Logical Drive Info response 161
Table 150 Add Logical Drive request 164
Table 151 Add Logical Drive response 165
Table 152 Delete Logical Drive request 166
Table 153 Delete Logical Drive response 166
Table 154 Configurable Logical Drive Attributes 167
Table 155 Get iSCSI Target Info request 167
Table 156 Get iSCSI Target Info response 167
Table 157 Configurable iSCSI Target Attributes 169
Table 158 Add iSCSI Target request 169
Table 159 Add iSCSI Target response 170
Table 160 Delete iSCSI Target request 171
Table 161 Delete iSCSI Target response 171
Table 162 GAMI Commands Support Requirements 173

Revision History

Revision	Description	Date
002	Intermediate release.	April 20, 2017
001	Initial release.	February 9, 2017

1 Introduction

1.1 Scope

This document contains information about the Generic Asset Management Interface API designed and implemented for Intel® Rack Scale Design Software 2.0 Release for the Bulldog Creek SDV.

1.2 Intended audience

The intended audiences for this document include designers and engineers working with the Software 2.0 Release, porting this software to HW platforms other than the Bulldog Creek SDV.

1.3 Terminology

Table 1 Terminology

Term	Definition
AMC	Asset Management Core
BMC	Baseboard Management Controller
GAM	Generic Asset Management
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*.
Module	Physical component housing a blade or switch.
OData	Open Data Protocol
PCIe	Peripheral Component Interconnect Express
PNC	Pooled NVMe Controller
POD	A physical collection of multiple racks
PODM	POD Manager
PSME	Pooled System Management Engine
REST	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

1.4 References

Doc ID	Title	Location
335451	Intel® Rack Scale Design Generic Assets Management Interface API Specification	*Intel.com/intelrsd_resources*
335452	Intel® Rack Scale Design BIOS & BMC Technical Guide	*Intel.com/intelrsd_resources*
335501	Intel® Rack Scale Design Architecture Specification	*Intel.com/intelrsd_resources*
335454	Intel® Rack Scale Design Software Reference Kit Getting Started Guide	*Intel.com/intelrsd_resources*
335455	Intel® Rack Scale Design Pod Manager API Specification	*Intel.com/intelrsd_resources*
335456	Intel® Rack Scale Design Pod Manager Release Notes	*Intel.com/intelrsd_resources*
335457	Intel® Rack Scale Design Pod Manager User Guide	*Intel.com/intelrsd_resources*
335458	Intel® Rack Scale Design PSME REST API Specification	*Intel.com/intelrsd_resources*
335459	Intel® Rack Scale Design PSME Release Notes	*Intel.com/intelrsd_resources*
335460	Intel® Rack Scale Design PSME User Guide	*Intel.com/intelrsd_resources*
335461	Intel® Rack Scale Design Storage Services API Specification	*Intel.com/intelrsd_resources*
335462	Intel® Rack Scale Design Rack Management Module (RMM) API Specification	*Intel.com/intelrsd_resources*
335463	Intel® Rack Scale Design RMM Release Notes	*Intel.com/intelrsd_resources*
335464	Intel® Rack Scale Design Software RMM User Guide	*Intel.com/intelrsd_resources*
DSP0266	Redfish Scalable Platform Management API Specification	*http://dmtf.org/standards/redfish*

1.5 Notes and Symbol Convention

- Symbol and note convention are similar to typographical conventions used in CIMI specification.
- 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 that the lack of an ellipses does not mean no extensibility point exists, rather it is just not explicitly called out.

§

2 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 2.0 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*.
- *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 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 GAM module receives a command, it decides to finish the action synchronously or asynchronously using the Task mechanism. If action is completed asynchronously, GAM module will return Task UUID. Command **Get Task Info** is for checking the Task state and its additional properties. If Task's state is completed, **Get Task Result** command will return UUID and type of component affected by task.

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

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

Figure 2 Sample asynchronous communication

Example of asynchronous request and response exchange with Add Logical Drive command.

Request:

```
{
        "jsonrpc": "2.0",
        "method": "addLogicalDrive",
        "params": {
                "type": "LVM",
                "capacityGB": 100,
                "mode": "LV",
                "master": "863e4567-e87b-64d3-a489-987656540000",
                "snapshot": false,
                "image": "OpenStack Juno Ubuntu 15.04",
                "protected": false,
                "drives": [
                        "cd3e4527-af7b-32d3-a489-987656a3d588"
                ],
                "oem": {}
        },
        "id": 1
}
```

Response:

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

```
        "id": 2
}
```

Sample Get Task Info request:

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

Response:

```
{
        "jsonrpc": "2.0",
        "result": {
                "name": "CloneLogicalDrive",
                "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
}
```

Get Task Result request:

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

Response:

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

Get Logical Drive Info request:

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

```
        "method": "getLogicalDriveInfo",
        "params": {
                "drive": "987e4567-n46j-34dx-lk78-jg8555440473"
        },
        "id": 7
}
```

Response:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "type": "LVM",
                "capacityGB": 100,
                "mode": "LV",
                "master": "863e4567-e87b-64d3-a489-987656540000",
                "snapshot": false,
                "image": "OpenStack Juno Ubuntu 15.04",
                "bootable": true,
                "protected": false,
                "collections": [
                        {
                                "name": "LogicalDrives",
                                "type": "LogicalDrives"
                        }
                ]
                "oem": {}
        },
        "id": 8
}
```

§

3 GAMI API Error Handling

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

3.1 API error response

The GAMI JSON-RPC API complies with *JSON-RPC 2.0* specification. Each request may return an error response according to the JSON-RPC 2.0 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

3.2.1 General error codes

The error codes described by JSON-RPC 2.0 specification are listed in Table 2. Any command of GAMI protocol described in section 5 may return any of these codes.

Table 2 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.
-1	Unknown Error	Unknown error occurred.
10000	Not found	Requested component could not be found.
11000	Not implemented	The method is not implemented.

Error Code	Message	Description
14000	Invalid value	The JSON contains invalid value for some parameter. This is a generic value related error code used when none of 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 supported values list.
14400	Invalid value format	The value has incompatible format.
14500	Invalid UUID	Component with requested UUID could not be found.
15000	Invalid field	The JSON contains invalid filed name. This is a generic property related error code used when none of 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 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.
30000	Network Error	Set of errors concerning Network module.
40000	Storage Error	Set of errors concerning Storage module.
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.

§

4 GAMI Data Model

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 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 that has at least one collection of subcomponents has the special property object that defines these collections. This property is a table with entries defining the single collection. The collection definition consists of three fields as presented in Table 3.

Table 3 GAMI subcomponent collection definition

Attribute	Type	Mandatory	Description
name	String	Yes	Collection name; unique collection name used to identified 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.
slotMask	String	No	Occupied slots mask; optional parameters that provide info about the max number of slots and location of occupied slots by subcomponents in the particular collection. This attribute makes sense for some of collection (for example, chassis, processors, or fans) that occupied a physical slot or socket, but it does not make any sense for others (like VLANs) and hence should not be provided as a part of the collection definition.

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 *Get Processor Info* command to retrieve detailed information about a given processor identified by 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 *Get Collection* command is described in section 5.5.

4.7 Mandatory and Optional properties

The JSON structures exchanged through the GAMI interface contains the fields defined as "Mandatory" or "Optional". This information is necessary for the receiver of a JSON structure to properly 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 interface specification.
- If a field is specified as "Optional" it means the JSON stricture without such a field is still compliant with GAMI interface specification.

There are a few way how the "Optional" field may be specified in the request JSON structure:

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

g. Set to “null”

Please note that 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 fully avoided, or specified in the same way as in the request JSON structure.

§

5 GAMI API Definition

5.1 GAM Module Registration

Each Generic Asset Management (GAM) Module must register to the Asset Management Core (AMC) application service in order to establish 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 Registration command sent by the GAM module to the AMC application registers the GAM module to the AMC application.

Table 4 GAM Module 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”, “StorageServices”, “Chassis”, “PCIeFabric”

Serialization:

```
{
        “jsonrpc”: “2.0”,
        “method”: “attach”,
        “params”: {
                “version”: <string>,
                “vendor”: <string>,
                “ipv4address”: <string>,
                “port”: <number>,
                “gamiId”: <string>,
                “capabilities”: [
                <{“Compute”, “Network”, “StorageServices”, “Chassis”,
           “PCIeFabric”}>,
                        ...
                ]
        },
        “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

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

Table 5 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 Heart beat command to the AMC application periodically to detect its status. If the GAM module does not receive a response for the Heart beat command, it means that AMC application is not running. If a timestamp returned by the AMC application in a response is lower than a timestamp returned in the previous command, it means that the AMC application has been restarted, since the heart beat timer is restarted along with AMC application. In both cases, the GAM module will reset its state and start new registration procedure.

5.2.1 Request

Table 6 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 7 Heartbeat command response

Parameters	Type	Mandatory	Description
timeStamp	Number	Yes	Time stamp 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 Get Managers Collection

The AMC retrieves collection of main managers supported by the particular GAM Module by sending 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 8 Get Managers Collection response

Parameters	Type	Mandatory	Description
managers	Array: Object	Yes	Member: manager; Type: String; Mandatory: Yes; Description: 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 Get Manager Info

Get Manager Info command retrieves detailed information about a single manager.

5.4.1 Request

Table 9 Get Manager 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 10 Get Manager Info response

Result	Type	Mandatory	Description
status	Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** state / String / Yes / Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline" health / String / No / Overall health state from the view of this resource: "OK", "Warning", "Critical"
type	String	Yes	Manager type: "ManagementController", "EnclosureManager", "BMC", "RackManager", "AuxiliaryController"
model	String	No	Manager model
location	String	No	UUID of the chassis being a manager's physical location
guid	String	No	GUID of the management controller.
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 RFC 3339.
dateTimeLocalOffset	String	No	The time offset from UTC that the DateTime property is set to in format: +06:00.
firmwareVersion	String	Yes	Manager firmware version
ipv4Address	String	Yes	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" port / Number / No / Service port number

			enabled | Boolean | Yes | Is service enabled
graphicalConsole	Object	No	**Attribute** | **Type** | **Mandatory** | **Description** enabled | Boolean | Yes | Console availability maxSessions | Number | No | Number of session that can be established at the same time typesSupported | Array: String | No | Supported types of connection
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 flowControl | String | No | Flow control: “None”, “Software”, “Hardware” pinOut | String | No | Pin out: “Cisco”, “Cyclades”, “Digi” enabled | Boolean | Yes | Console availability maxSessions | Number | No | Number of session that can be established at the same time typesSupported | Array: String | No | Supported types of connection
commandShell	Object	No	**Attribute** | **Type** | **Mandatory** | **Description** enabled | Boolean | Yes | Console availability maxSessions | Number | No | Number of session that can be established at the same time typesSupported | Array: String | No | Supported types of connection
collections	Array: Object	Yes	**Attribute** | **Type** | **Mandatory** | **Description** name | String | Yes | Collection name type | String | Yes | Collection type: “Chassis”, ”Systems”, “EthernetSwitches”, “StorageServices”, “Managers”, “Fabrics”, “PCIeDevices” slotMask | String | No | Occupied slots mask (see section 4.5 for details)
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": <{"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"}>,
                          "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"}>,
                                "slotMask": <string>
                        }
                ...
                ],
                "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",
                                "slotMask": "10101010"
                        },
                        {
                                "name": "Drives",
                                "type": "Drives"
                        },
                                "name": "Chassis",
                                "type": "Chassis",
                        }
                ],
                "oem": {}
        },
        "id": 123
}
```

5.5 Get Tasks Collection

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

5.5.1 Request

Serialization:

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

Example

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

5.5.2 Response

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

Table 11 Get Tasks Collection response

Parameters	Type	Mandatory	Description			
tasks	Array: Object	Yes	**Member**	**Type**	**Mandatory**	**Description**
			task	String	Yes	UUID of task.

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
}
```

5.6 Get Task Info

Get Task Info command retrieves detailed information about a single task.

5.6.1 Request

Table 12 Get Task Info request

Parameters	Type	Mandatory	Description
task	String	Yes	UUID of 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
}
```

5.6.2 Response

Table 13 Get Task Info response

Result	Type	Mandatory	Description
name	String	Yes	Task name that identify asynchronous operation.
startTime	String	Yes	ISO 8601 timestamp string indicating task start time.
endTime	String	Yes	ISO 8601 timestamp string indicating task end time.
state	String	Yes	Task state indication: “New”, “Starting”, “Running”, “Suspended”, “Interrupted”, “Pending”, “Stopping”, “Completed”, “Killed”, “Exception”, “Service”.
status	Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** state / String / Yes / Known state of the component: "Enabled", “Disabled”, “StandbyOffline”, “StandbySpare,”,“InTest”, “Starting”, “Absent, “UnavailableOffline” health / String / No / Overall health state from the view of this resource: "OK", “Warning”, “Critical”
messages	Array:Object	No	**Attribute** / **Type** / **Mandatory** / **Description** id / String / Yes / Internal message ID. fields / Array:String / Yes / Field names related to this message.
oem	Object	No	OEM specific data

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"}>
                },
```

```
                "messages": [
                        {
                                "id": <string>,
                                "fields": [
                                                <string>,
                                                …
                                ]
                        },
                        …
                ],
                "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"
                },
                "messages": [
                        {
                                "id": "123e4567",
                                "fields": ["sku", "oem"]
                        }
                ],
                "oem": {}
        },
        "id": 123
}
```

5.7 Delete Task

Delete Task command deletes existing task.

5.7.1 Request

Table 14 Delete Task 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
}
```

5.7.2 Response

Table 15 Delete Task 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
}
```

5.8 Get Task Result

Retrieves UUID and type of component affected by completed task.

5.8.1 Request

Table 16 Get Task Result request

Parameters	Type	Mandatory	Description
task	String	Yes	Existing task UUID.

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getTaskResult",
        "params": {
                "task": <string>
```

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

Example:

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

5.8.2 Response

Table 17 Get Task Result response

Result	Type	Mandatory	Description
uuid	String	Yes	UUID of component affected by task.
type	String	Yes	Type of component affected by 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
}
```

5.9 Get Collection

The AMC retrieves a collection of different kind of subcomponents of the particular parent object by sending getCollection command to the appropriate GAM module.

5.9.1 Request

Table 18 Get Collection request

Parameters	Type	Mandatory	Description
component	String	Yes	Component UUID
name	String	No	Subcomponent collection name. The collection name is provide 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
}
```

5.9.2 Response

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

Table 19 Get Collection 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
}
```

5.10 Component Notification

The GAM module sends Component Notification command to 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 notification command; no response is expected to be sent by the AMC.

5.10.1 Notification

Table 20 Component Notification

Parameters	Type	Mandatory	Description
gamiId	String	Yes	GAM module unique identifier
component	String	Yes	Component UUID
notification	String	Yes	Notification type: "Add", "Remove", "Update"
parent	String	Yes	Parent component UUID, (may be empty(null) if there is no parent component in the hierarchy e.g. top chassis object)
type	String	Yes	Component type: "Chassis", "ThermalZone", "PowerZone", "Fan", "PSU", "Drive", "PhysicalDrive", "PCIeDevice".
timeStamp	Number	No	Notification time stamp

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "componentNotification",
        "params": {
                "gamiId": <string>,
                "component": <string>,
                "notification": <"Add", "Remove", "Update">
                "parent": <string>
                "type": <"Chassis" , "ThermalZone", "PowerZone", "Fan", "PSU",
                "Drive", "PhysicalDrive">
                timeStamp: <number>,
        }
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "componentNotification",
        "params": {
                "gamiId": "40bf1aa4-c440-11e5-9202-00a0c923456f",
                "component": "123e4567-e89b-12d3-a456-426655440000",
                "notification": "Remove",
                "parent": "223e4567-e89b-12d3-a456-426655441234"
                "type": "PSU"
                timeStamp: 5443323,
        }
}
```

5.11 Set Component Attributes

The Set Component Attributes sets new value for one or more attributes of the single component. The command is universal and may be used for various components. If some of the attributes are not set, response will contain error message about the command failure.

5.11.1 Request

Table 21 Set Component Attributes 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
}
```

5.11.2 Response

Table 22 Set Component Attributes response

Parameters	Type	Mandatory	Description
Array of Objects	Array: Object	No	

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",
        "result": [
                        {
                                "attribute": <string>,
                                "code": <error code>,
                                "message": <string>
                        },
                        ...
        ],
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": [
                        {
                                "attribute": "initiatorIQN",
                                "code": 14000,
                                "message": "Binding already exists!"
                        }
        ],

        "id": 987
}
```

5.12 Get Computer System Info

Get Computer System Info command retrieves detailed information about a single computer system.

5.12.1 Request

Table 23 Get Computer System Info 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
}
```

5.12.2 Response

Table 24 Get Computer System Info 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".
biosVersion	String	No	BIOS version
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"
bootSourceOverrideMode	String	Yes	The BIOS Boot Mode (either Legacy or UEFI) to be used when BootSourceOverrideTarget boot source is booted from: "Legacy", "UEFI"
bootOverrideSupported	Array: String	Yes	Boot source override supported targets: "None", "Pxe", "Floppy", "Cd", "Usb", "Hdd", "BiosSetup", "Utilities", "Diags", "UefiTarget"
uefiTarget	String	No	Uefi Device Path of the device to boot from when BootSourceOverrideTarget is UefiTarget.
powerState	String	Yes	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><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>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>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”, “StorageSubsystems”, “NetworkInterfaces”, “NetworkDevices”</td></tr>
<tr><td>slotMask</td><td>String</td><td>No</td><td>Occupied slots mask (see section 4.5 for details)</td></tr>
</table>
</td></tr>
<tr><td>chassis</td><td>String</td><td>Yes</td><td>UUID of the chassis being 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”}>,
                “bootOverrideSupported”: [
                <{“Pxe”, “Floppy”, “Cd”, “Usb”, “Hdd”, “BiosSetup”,
            “Utilities”, “Diags”, “UefiTarget”}>,
                ...
                ],
                “uefiTarget”: <string>,
```

```
                "powerState": <{"On", "Off"}>,
                "pciDevices": [
                        {
                                "vendorId" : <string>,
                                "deviceId" : <string>
                        },
                ...
                ],
                "usbDevices": [
                        {
                                "vendorId" : <string>,
                                "deviceId" : <string>
                        },
                ...
                ],
                "fruInfo": {
                        "serialNumber": <string>,
                        "manufacturer": <string>,
                        "modelNumber": <string>,
                        "partNumber": <string>
                },
                "sku": <string>,
                "assetTag": <string>,
                "indicatorLED": <{"Lit", "Blinking", "Off"}>,
                "collections": [
                        {
                                "name": <string>,
                        "type": <{"Processors", "Memory", "StorageSubsystems",
                        "NetworkInterfaces"}>,
                                "slotMask": <string>
                        }
                ...
                ],
        "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"
                        }
                ],
                "usbDevices": [
                        {
                                "vendorId" : "0x8086",
                                "deviceId" : "0x5678"
                        }
                ],
                "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",
                                "slotMask": "11"
                        },
                        {
                                "name": "Memory",
                                "type": "Memory"
                        },
                        {
                                "name": "StorageSubsystems",
                                "type": "StorageSubsystems"
                        },
                        {
                                "name": "Network",
                                "type": "NetworkInterfaces"
                        }
                ],
                "chassis": "123e4567-e89b-12d3-a456-426655440000",
                "cableIds": ["189237", "hg7234"],
                "oem": {}
        },
        "id": 234
}
```

5.13 Set Computer System Attributes

The *Set Component Attributes* described in the section5.11 allows configuration of the computer system attributes listed in the following table.

Table 25 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"
bootOverrideMode	String	Boot source override mode: “Legacy”, “UEFI”.
powerState	String	Power state of the system: “On”, “ForceOff”, “GracefulShutdown”, “ForceRestart”, “Nmi”, “ForceOn”, “PushPowerButton”
assetTag	String	Tag assigned to the asset.
oem	Object	OEM specific data

5.14 Get Processor Info

Get Processor Info command retrieves detailed information about a single processor.

5.14.1 Request

Table 26 Get Processor Info 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
}
```

5.14.2 Response

Table 27 Get Processor Info 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>
socket	String	No	Socket identifier where processor is installed, e.g. “CPU 1”
processorType	String	Yes	Processor type: "CPU", "GPU", "FPGA", "DSP", "Accelerator", "OEM"
processorArchitecture	String	Yes	Processor architecture: "x86", "IA-64", "ARM", "MIPS", “OEM”
instructionSet	String	Yes	Supported CPU instruction sets such as: "x86", "x86-64", “IA-64”, "ARM-A32”, “ARM-A64”, "MIPS32", “MIPS64”, “OEM”
capabilities	Array: String	No	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	Yes	Processor manufacturer e.g. "Intel(R) Corporation"
model	String	Yes	Processor model (based on CPUID): Xeon family: “E3”, “E5”, “E7” SoC family: “X3” (Avoton), “X5” (Broadwell-DE), “X7” Core family: “I3”, “I5”, “I7” Unknown model: “Unknown”
modelName	String	Yes	Processor model, e.g. "Multi-Core Intel(R) Xeon(R) processor 7xxx Series"
cpuid	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>No</td><td>Numeric vendor id: “GenuineIntel”</td></tr><tr><td>numericId</td><td>String</td><td>No</td><td>cpuid.1.eax[31:0]</td></tr><tr><td>family</td><td>String</td><td>No</td><td>CPU family</td></tr><tr><td>model</td><td>String</td><td>No</td><td>CPU model</td></tr><tr><td>step</td><td>String</td><td>No</td><td>CPU stepping</td></tr><tr><td>microcodelnfo</td><td>String</td><td>No</td><td>Microcode version</td></tr></table>
maxSpeedMHz	Number	No	Maximum supported frequency of CPU
totalCores	Number	No	Number of available cores
enabledCores	Number	No	Number of enabled cores
totalThreads	Number	No	Number of available threads
enabledThreads	Number	No	Number of enabled threads
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"}>
                },
                "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"}>,
                "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>,
                "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,
                "oem": {}
        },
        "id": 987
}
```

5.15 Get Memory Info

Get Memory Info command retrieves detailed information about a single memory module.

5.15.1 Request

Table 28 Get Memory Info 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
}
```

5.15.2 Response

Table 29 Get Memory Info 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>
memoryType	String	No	Memory type: “DRAM”,“NVDIMM_N”, “NVDIMM_F”, “NVDIMM_P”
deviceType	String	No	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”
moduleType	String	No	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”
media	Array: String	No	Memory module media types: “DRAM”, “NAND”, “Proprietary”
memoryModes	Array: String	No	Memory modes supported by the memory module: “Volatile”, “Persistent”, “Block”
capacityMB	Number	Yes	Memory module size in MB
dataWidthBits	Number	No	Data Width in bits
busWidthBits	Number	No	Bus Width in bits
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>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>
firmwareRevision	String	No	Memory module firmware revision
firmwareApiVersion	String	No	Memory module firmware API version
functionClasses	Array: String	No	Function classes supported by the DIMM: “Volatile”, “Block”, “Persistent”
vendorID	String	No	Vendor ID
deviceID	String	No	Device ID
operatingSpeedMHz	Number	No	Memory module operating speed in MHz
allowedSpeedsMHz	Array: Number	No	Speed bins supported by this memory module
voltageVolt	Number	No	Typical memory module voltage
minimumVoltageVolt	Number	No	Minimum memory module voltage
maximumVoltageVolt	Number	No	Maximum memory module voltage
deviceLocator	String	No	Location of the DIMM in the platform, typically marked in the silk screen

<table>
<tr><td>location</td><td>Object</td><td>No</td><td>Property describing DIMM location with respect to 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>regions</td><td>Array: Object</td><td>No</td><td>Memory regions configured within the memory module
<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>regionId</td><td>String</td><td>No</td><td>Unique region ID representing a specific region within the memory module</td></tr>
<tr><td>memoryType</td><td>String</td><td>No</td><td>Region memory type: “Volatile”, “Block”, “Persistent”</td></tr>
<tr><td>offsetMB</td><td>Number</td><td>No</td><td>Offset with in the memory module corresponding to the region begin</td></tr>
<tr><td>sizeMB</td><td>Number</td><td>No</td><td>Region size in MB</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”}>,
                “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”}>,
                       ...
                ],
                “memoryModes”: [
                       <{“Volatile”, “Persistent”, “Block” }>,
                       ...
                ],
                “capacityMB”: <number>,
                “dataWidthBits”: <number>,
                “busWidthBits”: <number>,
```

```
                "fruInfo": {
                        "serialNumber": <string>,
                        "manufacturer": <string>,
                        "modelNumber": <string>,
                        "partNumber": <string>
                },
                "firmwareRevision": <string>,
                "firmwareApiVersion": <string>,
                "functionClasses": [
                        <{"Volatile", "Block", "Persistent"}>,
                        ...
                ],
                "vendorID": <string>,
                "deviceID": <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"}>,
                        "offsetMB": <number>,
                        "sizeMB": <number>
                        }
                ...
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "memoryType": "DRAM",
                "deviceType": "DDR4",
```

```
                "moduleType": "RDIMM",
                "media": [
                      "DRAM"
                ],
                "memoryModes": [
                      "Volatile"
                ],
                "capacityMB": 16384,
                "dataWidthBits": 64,
                "busWidthBits": 72,
                "fruInfo": {
                      "serialNumber": "123fed3029c-b23394-12",
                      "manufacturer": "Intel Corporation",
                      "modelNumber": "E323",
                      "partNumber": "29ee2220939"
                },
                "firmwareRevision": "RevAbc",
                "firmwareApiVersion": "ApiAbc",
                "functionClasses": [
                      "Volatile"
                ],
                "vendorID": "0x8086",
                "deviceID": "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",
                      "offsetMB": 0,
                      "sizeMB": 16384
                      }
                ],
                "oem": {}
        },
        "id": 987
}
```

5.16 Get Storage Subsystem Info

Get Storage Subsystem Info command retrieves detailed information about a single storage subsystem.

5.16.1 Request

Table 30 Get Storage Subsystem Info 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
}
```

5.16.2 Response

Table 31 Get Storage Subsystem Info response

Result	Type	Mandatory	Description
status	Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** state / String / Yes / Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline" health / String / Yes / Overall health state from the view of this resource: "OK", "Warning", "Critical"
collections	Array: Object	Yes	Subcomponents collections **Attribute** / **Type** / **Mandatory** / **Description** name / String / Yes / Collection name type / String / Yes / Collection type: "Drives", "StorageControllers" slotMask / String / No / Occupied slots mask (see section 4.5 for details)
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"}>,
                               "slotMask": <string>
                       }
                ...
                ],
                "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
}
```

5.17 Set Storage Subsystem Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the storage controller attributes listed in the following table.

Table 32 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

5.18 Get Storage Controller Info

Get Storage Controller Info command retrieves detailed information about a single storage controller.

5.18.1 Request

Table 33 Get Storage Controller Info 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
}
```

5.18.2 Response

Table 34 Get Storage Controller Info response

Result	Type	Mandatory	Description
status	Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** state / String / Yes / Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline" health / String / Yes / Overall health state from the view of this resource: "OK", "Warning", "Critical"
fruInfo	Object	No	**Attribute** / **Type** / **Mandatory** / **Description** serialNumber / String / No / Module serial number

<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>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 & 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 world wide, 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”.</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"}>
                        },
                        …
                ],
                "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"
                },
                "interface": "SAS"
                "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
}
```

5.19 Get Drive Info

Get Drive Info command retrieves detailed information about a single drive.

5.19.1 Request

Table 35 Get Drive Info 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
}
```

5.19.2 Response

Table 36 Get Drive Info response

Result	Type	Mandatory	Description
status	Object	Yes	

Attribute	Type	Mandatory	Description
state	String	Yes	Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline"
health	String	Yes	Overall health state from the view of this resource: "OK", "Warning", "Critical"

interface	String	Yes	Drive interface: "PCIe", “AHCI”, “UHCI”, “SAS”, “SATA”, “USB”, “NVMe”, “FC”, “iSCSI”, “FCoE”, “NVMeOverFabrics”, “SMB”, “NFSv3”, “NFSv4”, “HTTP”, ”HTTPS”, “FTP”, “SFTP”.
type	String	No	Drive type: “HDD”, “SSD”, “SMR”
rpm	Number	No	Rotation per minute (HDD only)
capacityGB	Number	Yes	Drive capacity in GB
fruInfo	Object	No	**Attribute** / **Type** / **Mandatory** / **Description** serialNumber / String / No / Module serial number manufacturer / String / No / Manufacturer name modelNumber / String / No / Model number partNumber / String / No / Part number
firmwareVersion	String	No	Drive firmware version
physicalId	String	No	Drive physical location (JBOD)
indicatorLED	String	No	The state of the indicator LED, used to identify the drive: “Lit”, “Blinking”, “Off”.
assetTag	String	No	The value of this property shall be an identifying string used to track the drive for inventory purposes.
capableSpeedGbps	Number	No	The speed which this drive can communicate to a storage controller in ideal conditions in Gigabits per second.
negotiatedSpeedGbps	Number	No	The speed which this drive is currently communicating to the storage controller in Gigabits per second.
location	Array:Object	No	**Attribute** / **Type** / **Mandatory** / **Description** info / String / Yes / This indicates the location of the resource. infoFormat / String / Yes / This represents the format of the Info property.
statusIndicator	String	No	The state of the status indicator, used to communicate status information about this drive: “Ok”, “Fail”, “Rebuild”, “PredictiveFailureAnalysis”, “Hotspare”, “InACriticalArray”, “InAFailedArray”
revision	String	No	The revision of this Drive.
failurePredicted	Boolean	No	Is this drive currently predicting a failure in the near future.
sku	String	No	This is the SKU for this drive.
identifiers	Array:Object	No	This property shall contain a list of all known durable names for the associated drive: **Attribute** / **Type** / **Mandatory** / **Description** durableName / String / Yes / This indicates the world wide, persistent name of the resource. durableNameFormat / String / Yes / Represents the format of the durableName property: “NAA”, “iQN”, “FC_WWN”, “UUID”, “EUI”.
hotspareType	String	No	The type of hotspare this drive is currently serving as: “None”, “Global”, “Chassis”, “Dedicated”
encryptionAbiity	String	No	The encryption abilities of this drive: “None”, “SelfEncryptingDrive”, “Other”.
encryptionStatus	String	No	The status of the encryption of this drive: “Unencrypted”,”Unlocked”,”Locked”,”Foreign”

blockSizeBytes	Number	No	The size of the smallest unit (Block) of this drive in bytes.
predictedMediaLifeLeftPercent	Number	No	The percentage of reads and writes that are predicted to still be available for the media.
erased	Boolean	No	True if data on drive has been erased.
collections	Array: Object	No	Subcomponents collections **Attribute** / **Type** / **Mandatory** / **Description** name / String / Yes / Collection name type / String / Yes / Collection type: "PCIeFunctions" slotMask / String / No / Occupied slots mask (see section 4.5 for details)
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"}>
                },
                "interface": <string>
                "type": <string>,
                "rpm": <number>,
                "capacityGB": <number>,
                "fruInfo": {
                       "serialNumber": <string>,
                       "manufacturer": <string>,
                       "modelNumber": <string>,
                       "partNumber": <string>
                },
                "firmwareVersion": <string>,
                "physicalId" : <string>,
                "pcieEndpoint": <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"}>
                        },
                        …
                ],
                "hotspareType": <{"None", "Global", "Chassis", "Dedicated"}>,
                "encryptionAbility": <{"None", "SelfEncryptingDrive", "Other
}>,
                "encryptionStatus":
<{"Unencrypted","Unlocked","Locked","Foreign"}>,
                "blockSizeBytes": <number>,
                "predictedMediaLifeLeftPercent": <number>,
                "erased": <boolean>,
                "collections": [
                        {
                                "name": "PCIeFunctions",
                                "type": "PCIeFunctions",
                        }
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "interface": "SATA"
                "type": "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,
                "erased": true,
                "collections": [
                        {
                                "name": "PCIeFunctions",
                                "type": "PCIeFunctions",
                        }
                ],
                "oem": {}
        },
        "id": 987
}
```


5.20 Set Drive Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the drive in the following table.

Table 37 Set Drive Attributes

Attribute	Type	Description
assetTag	String	Tag assigned to the asset.
securelyErase	Boolean	Triggers secure drive erase.
erased	Boolean	Indicates whether the data has been securely erased.
eraseOnDetach	Boolean	Indicates whether the data will be securely erased on drive detach.
oem	Object	OEM specific data

5.21 Get Network Interface Info

Get Network Interface Info commands retrieves detailed information about a network interface.

5.21.1 Request

Table 38 Get Network Interface Info 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
}
```

5.21.2 Response

Table 39 Get Network Interface Info response

Result	Type	Mandatory	Description
status	Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** state / String / Yes / Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline" health / String / Yes / Overall health state from the view of this resource: "OK", "Warning", "Critical"
frameSize	Number	No	MAC Frame size in bytes
speedMbps	Number	No	The current speed in Mbps of this network interface
autoSense	Boolean	No	Indicates if the speed and duplex is automatically configured by the NIC
fullDuplex	Boolean	No	Indicates if the NIC is in Full Duplex mode or not
macAddress	String	No	This is the currently assigned MAC address for this NIC
factoryMacAddress	String	No	Factory assigned MAN address
ipv4Addresses	Array: Object	No	**Attribute** / **Type** / **Mandatory** / **Description** address / String / Yes / Ipv4 address subnetMask / String / No / Ipv4 subnet mask addressOrigin / String / No / Ipv4 address origin "DHCP", "Static" gateway / String / No / Ipv4 gateway for this address
ipv6Addresses	Array: Object	No	**Attribute** / **Type** / **Mandatory** / **Description** address / String / Yes / Ipv6 address prefixLength / Number / No / Ipv6 Address Prefix Length addressOrigin / String / No / Ipv4 address origin "DHCP", "Static", "SLAAC"

			addressState | String | No | Ipv6 address state: “Preferred”, “Deprecated”, “Tentative”, “Failed”
ipv6DefaultGateway	String	No	Ipv6 default gateway address
maxIPv6StaticAddresses	Number	No	Indicates the maximum number of Static Ipv6 addresses that can be configured on this interface
vlanEnable	Boolean	Yes	Indicates if VLANs are enabled on the switch port
defaultVlan	Number	No	Default VLAN identifier
collections	Array: Object	Yes	Subcomponents collections
oem	Object	No	OEM specific data

Attribute	Type	Mandatory	Description
name	String	Yes	Collection name
type	String	Yes	Collection type “EthernetSwitchPortVlans” , “NetworkDeviceFunctions”
slotMask	String	No	Occupied slots mask (see section 4.5 for details)

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”: <string>,
                                “addressState”: <string>
                        }
                ...
```

```
                ],
                "ipv6DefaultGateway": <string>,
                "maxIPv6StaticAddresses": <number>,
                "vlanEnable": <boolean>,
                "defaultVlan": <number>,
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"EthernetSwitchPortVlans"}>,
                                "slotMask": <string>
                        }
                ...
                ],
                "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,
                "vlanEnable": true,
                "defaultVlan": 4090,
                "collections": [
                        {
                                "name": "EthernetSwitchPortVlans",
```

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

5.22 Set Network Interface Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the network interface attributes listed in the following table.

Table 40 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 speed and duplex is automatically configured by the interface
oem	Object	OEM specific data

5.23 Get Network Device Info

Get Network Device Info command retrieves detailed information about network device.

5.23.1 Request

Table 41 Get Network Interface Info request

Parameters	Type	Mandatory	Description
device	String	Yes	Managed network device UUID

Serialization:

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

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getNetworkDeviceInfo",
        "params": {
                "device": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 2113
}
```

5.23.2 Response

Table 42 Get Network Device Info 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>
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 “NetworkDeviceFunctions”</td></tr><tr><td>slotMask</td><td>String</td><td>No</td><td>Occupied slots mask (see section 4.5 for details)</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”: <{”NetworkDeviceFunctions”}>,
                                “slotMask”: <string>
                        }
                ...
                ],
                “oem”: <object>
        },
        “id”: <id>
}
```

Example:

```
{
        “jsonrpc”: “2.0”,
        “result”: {
                “status”: {
                        “state”: “Enabled”,
```

```
                        "health": "OK"
                },
                "collections": [
                        {
                                "name": "NetworkDeviceFunctions",
                                "type": "NetworkDeviceFunctions"
                        }
                ],
                "oem": {}
        },
        "id": 987
}
```

5.24 Get Network Device Function Info

Get Network Device Function Info command retrieves detailed information about network device function.

5.24.1 Request

Table 43 Get Network Device Function Info request

Parameters	Type	Mandatory	Description
function	String	Yes	Network device function UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getNetworkDeviceFunctionInfo",
        "params": {
                "function": <String>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getNetworkDeviceFunctionInfo",
        "params": {
                "function": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 2113
}
```

5.24.2 Response

Table 44 Get Network Device Info response

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

			health | String | Yes | Overall health state from the view of this resource: "OK", "Warning", "Critical"
deviceEnabled	Boolean	Yes	Whether the network device function is enabled.
ethernet	Object	Yes	**Attribute** | **Type** | **Mandatory** | **Description** macAddress | String | No | MAC address of the NIC
iSCSIBoot	Object	No	**Attribute** | **Type** | **Mandatory** | **Description** ipAddressType | String | Yes | Type of IP address. Supported values:"IPv4", "IPv6" initiatorIpAddress | String | Yes | IPv4 or IPv6 address of iSCSI initiator initiatorName | String | Yes | iSCSI initiator name initiatorDefaultGateway | String | No | IPv4 or IPv6 iSCSI boot default gateway initiatorNetmask | String | Yes | IPv4 or IPv6 netmask of iSCSI boot initiator targetInfoViaDhcp | Boolean | Yes | Whether the iSCSI boot target name, LUN , IP address and netmask should be obtained from DHCP primaryTargetName | String | No | Name of iSCSI primary boot target primaryTargetIpAddress | String | No | IPv4 or IPv6 address of primary iSCSI boot target primaryTargetTcpPort | Number | No | TCP port number of primary iSCSI boot target primaryLun | Number | No | Logical Unit Number for the primary iSCSI boot target primaryVlanEnable | Boolean | Yes | Indicator whether primary VLAN is enabled primaryVlanId | Number | No | Primary VLAN id primaryDns | String | No | Primary DNS IPv4 or IPv6 address secondaryTargetName | String | No | Name of secondary iSCSI boot target name secondaryTargetIpAddress | String | No | IPv4 of IPv6 iSCSI boot target secondaryTargetTcpPort | Number | No | TCP port number of secondary iSCSI boot target secondaryLun | Number | No | LUN for secondary iSCSI boot target secondaryVlanEnable | Boolean | No | Indicator if secondary VLAN is enabled secondaryVlanId | Number | No | Secondary LAN Id secondaryDns | String | No | Secondary DNS

			ipMaskDnsViaDhcp	Boolean	Yes	Indicator if iSCSI boot initiator uses DHCP to obtain initiator name, IP address, and netmask
			routerAdvertisementEnabled	Boolean	Yes	Indicator if router advertisement is enabled for iSCSI boot target
			authenticationMethod	String	Yes	Supported values:"None", "CHAP", "MutualCHAP"
			chapUsername	String	No	Username for CHAP authentication
			chapSecret	String	No	Shared secret for CHAP authentication
			mutualChapUsername	String	No	Username for 2-way CHAP authentication
			mutualChapSecret	String	No	Secret for 2-way CHAP authentication
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"}>
                },
                "deviceEnabled": <Boolean>,
                "ethernet": <Object>,
                "iSCSIBoot": <Object>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                       "state": "Enabled",
                       "health": "OK"
                },
                "deviceEnabled": True,
                "ethernet": {
                       "macAddress": "AA:BB:CC:DD:EE:FF"
                },
                "iSCSIBoot": {
                       "ipAddressType": "IPv4",
                       "initiatorIpAddress": "10.1.3.122",
                       "initiatorName": "com.intel.rsd.node0001initiator",
                       "initiatorDefaultGateway": "10.1.3.1",
                       "initiatorNetmask": "255.255.0.0",
```

```
                        "targetInfoViaDhcp": False,
                        "primaryTargetName": "com.intel.rsd.ubuntu1604-20GB",
                        "primaryTargetIpAddress": "10.1.3.10",
                        "primaryTargetTcpPort": 3260,
                        "primaryLun": 1,
                        "primaryVlanEnable": False,
                        "ipMaskDnsViaDhcp": False,
                        "routerAdvertisementEnabled": False,
                        "authenticationMethod": "None"
                },
                "oem": {}
        },
        "id": 987
}
```

5.25 Add Network Device Function

Adds Network Device Function.

5.25.1 Request

Table 45 Add Network Device Function request

Parameters	Type	Mandatory	Description
device	String	Yes	Network device UUID
deviceEnabled	Boolean	Yes	Whether the network device function is enabled.
ethernet	Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** macAddress / String / No / MAC address of the NIC
iSCSIBoot	Object	No	**Attribute** / **Type** / **Mandatory** / **Description** ipAddressType / String / Yes / *Supported values:"IPv4", "IPv6" initiatorIpAddress / String / Yes / * initiatorName / String / Yes / * initiatorDefaultGateway / String / No / * inititaorNetmask / String / Yes / * targetInfoViaDhcp / Boolean / Yes / * primaryTargetName / String / Yes / * primaryTargetIpAddress / String / Yes / * primaryTargetTcpPort / Number / Yes / * primaryLun / Number / Yes / * primaryVlanEnable / Boolean / Yes / * primaryVlanId / Number / No / * primaryDns / String / No / * secondaryTargetName / String / No / * secondaryTargetIpAddres s / String / No / * secondaryTargetTcpPort / Number / No / * secondaryLun / Number / No / * secondaryVlanEnable / Boolean / No / * secondaryVlanId / Number / No / * secondaryDns / String / No / * ipMaskDnsViaDhcp / Boolean / Yes / *

Parameters	Type	Mandatory	Description
			routerAdvertisementEnabled: Boolean, Yes, * authenticationMethod: String, Yes, *Supported values:"None", "CHAP", "MutualCHAP" chapUsername: String, No, * chapSecret: String, No, * mutualChapUsername: String, No, * mutualChapSecret: String, No, *
oem	Object	No	OEM specific data

*See *Get Network Device Function Info* for detailed property description.

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getNetworkDeviceFunctionInfo",
        "params": {
                "function": <string>,
                "deviceEnabled": <Boolean>,
                "ethernet": <Object>,
                "iSCSIBoot": <Object>,
                "bootMode": <String>,
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "getNetworkDeviceFunctionInfo",
        "params": {
                "function": "123e4567-e89b-12d3-a456-426655440000",
                "deviceEnabled": True,
                "ethernet": {
                        "macAddress": "AA:BB:CC:DD:EE:FF"
                },
                "iSCSIBoot": {
                        "ipAddressType": "IPv4",
                        "initiatorIpAddress": "10.1.3.122",
                        "initiatorName": "com.intel.rsd.node0001initiator",
                        "initiatorDefaultGateway": "10.1.3.1",
                        "initiatorNetmask": "255.255.0.0",
                        "targetInfoViaDhcp": False,
                        "primaryTargetName": "com.intel.rsd.ubuntu1604-20GB",
                        "primaryTargetIpAddress": "10.1.3.10",
                        "primaryTargetTcpPort": 3260,
                        "primaryLun": 1,
                        "primaryVlanEnable": False,
                        "ipMaskDnsViaDhcp": False,
```

```
                        "routerAdvertisementEnabled": False,
                        "authenticationMethod": "None"
                },
                "bootMode": "iSCSI",
        },
        "id": 2113
}
```

5.25.2 Response

Table 46 Add Network Device Function response

Result	Type	Mandatory	Description
function	String	Yes	Created Network Device Function UUID
oem	Object	No	OEM specific data

Serialization:

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

Example:

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

5.26 Set Network Device Function Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the network device function attributes listed in the following table.

Table 47 Configurable Network Device Function Attributes

Attribute	Type	Description
ethernet	Object	**Attribute** / **Type** / **Mandatory** / **Description** macAddress / String / No / MAC address of the NIC
iSCSIBoot	Object	**Attribute** / **Type** / **Mandatory** / **Description** ipAddressType / String / No / Type of IP address. Supported values:"IPv4", "IPv6" initiatorIpAddress / String / Nos / IPv4 or IPv6 address of iSCSI initiator initiatorName / String / No / iSCSI initiator name

		initiatorDefaultGateway	String	No	IPv4 or IPv6 iSCSI boot default gateway
		initiatorNetmask	String	No	IPv4 or IPv6 netmask of iSCSI boot initiator
		targetInfoViaDhcp	Boolean	No	Whether the iSCSI boot target name, LUN , IP address and netmask should be obtained from DHCP
		primaryTargetName	String	No	Name of iSCSI primary boot target
		primaryTargetIpAddress	String	No	IPv4 or IPv6 address of primary iSCSI boot target
		primaryTargetTcpPort	Number	No	TCP port number of primary iSCSI boot target
		primaryLun	Number	No	Logical Unit Number for the primary iSCSI boot target
		primaryVlanEnable	Boolean	No	Indicator whether primary VLAN is enabled
		primaryVlanId	Number	No	Primary VLAN id
		primaryDns	String	No	Primary DNS IPv4 or IPv6 address
		secondaryTargetName	String	No	Name of secondary iSCSI boot target name
		secondaryTargetIpAddress	String	No	IPv4 of IPv6 iSCSI boot target
		secondaryTargetTcpPort	Number	No	TCP port number of secondary iSCSI boot target
		secondaryLun	Number	No	LUN for secondary iSCSI boot target
		secondaryVlanEnable	Boolean	No	Indicator if secondary VLAN is enabled
		secondaryVlanId	Number	No	Secondary LAN Id
		secondaryDns	String	No	Secondary DNS
		ipMaskDnsViaDhcp	Boolean	No	Indicator if iSCSI boot initiator uses DHCP to obtain initiator name, IP address, and netmask
		routerAdvertisementEnabled	Boolean	No	Indicator if router advertisement is enabled for iSCSI boot target
		authenticationMethod	String	No	Supported values:"None", "CHAP", "MutualCHAP"
		chapUsername	String	No	Username for CHAP authentication
		chapSecret	String	No	Shared secret for CHAP authentication
		mutualChapUsername	String	No	Username for 2-way CHAP authentication
		mutualChapSecret	String	No	Secret for 2-way CHAP authentication
oem	Object	OEM specific data			

5.27 Delete Network Device Function

Deletes Network Device Functions.

5.27.1 Request

Table 48 Delete Network Device Function request

Parameters	Type	Mandatory	Description
function	String	Yes	Network device function UUID
oem	Object	No	OEM specific data

Serialization:

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

Example:

```
{
        "jsonrpc": "2.0",
        "method": "deleteNetworkDeviceFunction",
        "params": {
                "function": "123e4567-e89b-12d3-a456-426655440000"
        },
        "id": 2113
}
```

5.27.2 Response

Table 49 Delete Network Device Function 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
}
```

5.28 Get Fabric Info

Get Fabric command retrieves detailed information about single fabric.

5.28.1 Request

Table 50 Get Fabric Info 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
}
```

5.28.2 Response

Table 51 Get Fabric Info 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>
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: "Zones", "Switches", "Endpoints"</td></tr></table>
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”}>
                },
                “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
}
```

5.29 Get Switch Info

Get Switch Info command retrieves detailed information about a single switch.

5.29.1 Request

Table 52 Get Switch Info 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
}
```

5.29.2 Response

Table 53 Get Switch Info response

Result	Type	Mandatory	Description
status	Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** state / String / Yes / Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline" health / String / Yes / Overall health state from the view of this resource: "OK", "Warning", "Critical"
switchId	String	Yes	Switch identifier
chassis	String	Yes	UUID of the chassis being the physical container for the PCIe switch
assetTag	String	No	Asset tag

<table>
<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>model</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 switch 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: “Ports”</td></tr>
</table>
</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>powerState</td><td>String</td><td>No</td><td>The value of this property shall contain the power state of the switch: “On”, “Off”, “PoweringOn”, “PoweringOff”.</td></tr>
<tr><td>allowedActions</td><td>Array: String</td><td>Yes</td><td>List of allowed actions: "On", "ForceOff", "GracefulShutdown", "ForceRestart", "Nmi", "GracefulRestart", "ForceOn", "PushPowerButton"</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>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”}>
                },
                “switchId”: <string>,
                “chassis”: <string>,
                “assetTag”: <string>,
                “fruInfo”: {
                       “manufacturer”: <string>,
                       “model”: <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
}
```

5.30 Set Switch Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the switch attributes listed in the following table.

Table 54 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

5.31 Get Port Info

Get Port Info command retrieves detailed information about a single port.

5.31.1 Request

Table 55 Get Port Info 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
}
```

5.31.2 Response

Table 56 Get Port Info response

Result	Type	Mandatory	Description
status	Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** state / String / Yes / Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline" health / String / Yes / Overall health state from the view of this resource: "OK", "Warning", "Critical"
portId	String	Yes	Port Identifier (port index, name, etc.)
portType	String	Yes	Port type: "HostPort", "DevicePort", "InterSwitchPort"
cableId	String	No	Cable Identifier
speedGBps	Number	Yes	Port speed in GBps
width	Number	Yes	Port width (PCIe lanes number)

maxSpeedGbps	Number	Yes	Max port speed in GBps
maxWidth	Number	Yes	Max port width (PCIe lines number)
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”: <{“HostPort”, “DevicePort”, “InterSwitchPort”}>,
                “cableId”: <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”: “HostPort”,
                “cableId”: “Host1:3-Switch1:1”
                “speedGbps ”: 4,
                “width”: 2,
                “maxSpeedGTps”: 8,
                “maxWidth”: 4,
                “allowedActions”: [ “ForceRestart” ],
```

```
                "protocol" "PCIe",
                "oem": {}
        },
        "id": 987
}
```

5.32 Set Port Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the switch port attributes listed in the following table.

Table 57 Configurable Port Attributes

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

5.33 Get Endpoint Info

Get Endpoint Info command retrieves detailed information about a single endpoint.

5.33.1 Request

Table 58 Get Endpoint Info request

Parameters	Type	Mandatory	Description
endpoint	String	Yes	Managed endpoint UUID.

Serialization:

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

Example:

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

5.33.2 Response

Table 59 Get Endpoint Info 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>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 world wide, 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”.</td></tr>
</table>
</td></tr>
<tr><td>entities</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>type</td><td>String</td><td>Yes</td><td>This indicates entity type:
"RootComplex", "StorageInitiator", "NetworkController", "Drive", "StorageExpander", "DisplayController", "Bridge", "Processor"</td></tr>
<tr><td>role</td><td>String</td><td>Yes</td><td>Represents the entity role:
"Initiator", "Target", "Both"</td></tr>
<tr><td>entity</td><td>String</td><td>Yes</td><td>UUID of the entity component.</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>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:
“Ports”</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"}>
                },
                "identifiers": [
                       {
                               "durableName": <string>,
                               "durableNameFormat": <{"NAA", "iQN", "FC_WWN",
"UUID", "EUI"}>
                       },
                       …
                ],
                "entities": [
                       {
                       "type": <{"RootComplex", "StorageInitiator",
                      "NetworkController", "Drive", "StorageExpander",
                      "DisplayController", "Bridge", "Processor"}>,
                               "role": <{"Initiator", "Target", "Both"}>,
                               "entity": <string>
                       },
                       …
                ],
                "protocol": <{"PCIe", "AHCI", "UHCI", "SAS", "SATA", "USB",
                    "NVMe", "FC", "iSCSI", "FCoE", "NVMeOverFabrics", "SMB",
                    "NFSv3", "NFSv4", "HTTP", "HTTPS", "FTP", "FTPS"}>,
                "collections": [
                       {
                               "name": <string>,
                               "type": <{"Ports"}>
                       }
                …
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                       "state": "Enabled",
                       "health": "OK"
                },
                "identifiers": [
                       {
                               "durableName": "123e4567-e89b-12d3-a456-
426655440000",
```

```
                                "durableNameFormat": "UUID"
                        }
                ],
                "entities": [
                        {
                        "type": "Drive",
                                "role": "Target",
                                "entity": "123e4567-e89b-12d3-a456-
426655440000"
                        },
                        …
                ],
                "protocol": "PCIe",
                "collections": [
                        {
                                "name": "Ports",
                                "type": "Ports"
                        }
                ],
                "oem": {}
        },
        "id": 987
}
```

5.34 Get PCIe Device Info

Get PCIe Device Info command retrieves detailed information about a single PCIe device.

5.34.1 Request

Table 60 Get PCIe Device Info 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
}
```

5.34.2 Response

Table 61 Get PCIe Device Info 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>
deviceId	String	Yes	Device Identifier
deviceClass	String	Yes	Device type: “SingleFunction”, “MultiFunction”
speedGbps	Number	Yes	Link speed in GBps
width	Number	Yes	Link width
maxSpeedGbps	Number	No	Max link speed in GBps
maxWidth	Number	No	Max link width
firmwareVersion	String	No	Firmware version
chassis	String	Yes	UUID of the chassis being the physical container for the PCIe device
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>model</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 device 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: “PCIeFunctions”</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”}>
                },
                “deviceId”: <string>,
                “deviceClass”: <{“SingleFunction”, “MultiFunction”}>,
                “speedGbps”: <number>,
                “width”: <number>,
```

```
                "maxSpeedGbps": <number>,
                "maxWidth": <number>,
                "firmwareVersion": <string>,
                "chassis": <string>,
                "assetTag": <string>,
                "fruInfo": {
                        "manufacturer": <string>,
                        "model": <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",
                        "model": "Model",
                        "serialNumber": "123fed3029c-b23394-12",
                        "partNumber": "29ee2220939",
                },
                "sku": "Rev.1",
                "collections": [
                        {
                                "name": "Functions",
                                "type": "PCIeFunctions"
                        }
                ],
                "oem": {}
```

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

5.35 Set PCIe Device Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the PCIe device in the following table.

Table 62 Set PCIe Device Attributes

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

5.36 Get PCIe Function Info

Get PCIe Function Info command retrieves detailed information about a single PCIe function.

5.36.1 Request

Table 63 Get PCIe Function Info 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
}
```

5.36.2 Response

Table 64 Get PCIe Function Info response

Result	Type	Mandatory	Description
status	Object	Yes	

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

						"Enabled", “Disabled”, “StandbyOffline”, “StandbySpare,”,“InTest”, “Starting”, “Absent, “UnavailableOffline”
			health	String	Yes	Overall health state from the view of this resource: ”OK”, “Warning”, “Critical”
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
}
```

5.37 Get Zone Info

Get Zone Info command retrieves detailed information about a single zone.

5.37.1 Request

Table 65 Get Zone Info request

Parameters	Type	Mandatory	Description
zone	String	Yes	Managed zone UUID

Serialization:

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

Example:

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

5.37.2 Response

Table 66 Get Zone Info response

Result	Type	Mandatory	Description
status	Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** state / String / Yes / Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline" health / String / Yes / Overall health state from the view of this resource: "OK", "Warning", "Critical"
collections	Array: Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** name / String / Yes / Collection name type / String / Yes / Collection type: "Endpoints"
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": <{"Endpoints"}>
                        }
                ...
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

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

5.38 Add Zone

Add Zone command creates new zone.

5.38.1 Request

Table 67 Add Zone 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
}
```

5.38.2 Response

Table 68 Add Zone 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
}
```

5.39 Delete Zone

Delete Zone command deletes existing zone.

5.39.1 Request

Table 69 Delete Zone 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
}
```

5.39.2 Response

Table 70 Delete Zone 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
}
```

5.40 Add Zone Endpoints

Add Zone Endpoints command attaches existing endpoints to zone.

5.40.1 Request

Table 71 Add Zone Endpoints 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": "addPortStaticMac",
        "params": {
                "zone": <string>,
                "endpoints": [
                        <string>,
                        …
                ],
```

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

Example:

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

5.40.2 Response

Table 72 Add Zone Endpoints 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
}
```

5.41 Delete Zone Endpoints

Delete Zone Endpoints command detaches existing endpoints from zone.

5.41.1 Request

Table 73 Delete Zone Endpoints 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": "addPortStaticMac",
        "params": {
                "zone": <string>,
                "endpoints": [
                        <string>,
                        …
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

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

5.41.2 Response

Table 74 Delete Zone Endpoints 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
}
```

5.42 Get Ethernet Switch Info

Get Ethernet Switch Info command retrieves detailed information about a single Ethernet switch.

5.42.1 Request

Table 75 Get Ethernet Switch Info 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
}
```

5.42.2 Response

Table 76 Get Ethernet Switch Info 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>
switchIdentifier	String	Yes	Switch identifier
technology	String	No	Switch technology: "Ethernet", "PCIe"
macAddress	String	Yes	Switch MAC Address
firmwareName	String	No	Name of switch FW
firmwareVersion	String	No	Version of switch FW
role	String	No	Switch role in the network: "TOR", "EOR", "Drawer", "Unknown"

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>
manufacturingDate	String	No	Manufacturing date
location	Number	Yes	Switch location within the chassis (aka. socket)
chassis	String	Yes	UUID of the chassis containing the Ethernet switch
maxAclNumber	Number	No	The maximum number of ACLs that can be handled by the switch
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: ”EthernetSwitchPorts”, “Acls”, “NeighborSwitches”</td></tr><tr><td>slotMask</td><td>String</td><td>No</td><td>Occupied slots mask (see section 4.5 for details)</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>
                “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>,
                “collections”: [
                       {
                                “name”: <string>,
                                “type”: <{“EthernetSwitchPorts”, “Acls”,
“NeighborSwitches”}>,
                                “slotMask”: <string>
                       }
                …
```

```
                ],
                "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",
                "collections": [
                        {
                                "name": "EthernetSwitchPorts",
                                "type": "EthernetSwitchPorts",
                                "slotMask": "11111100011110001111000111101111"
                        },
                        {
                                "name": "Neighbors",
                                "type": "NeighborSwitches"
                        },
                        {
                                "name": "Acls",
                                "type": "Acls"
                        }
                ],
                "oem": {}
        },
        "id": 987
}
```

5.43 Get Ethernet Switch Port Info

Get Ethernet Switch Port Info command retrieves detailed information about a single switch port.

5.43.1 Request

Table 77 Get Ethernet Switch Port Info 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
}
```

5.43.2 Response

Table 78 Get Ethernet Switch Port Info response

Result	Type	Mandatory	Description
status	Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** state / String / Yes / Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline" health / String / Yes / Overall health state from the view of this resource: "OK", "Warning", "Critical"
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 operator: “Up”, “Down”
portWidth	Number	No	Port width, for PCI-e port this is number of PCI-e 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	
lastStateChangeTime	String	No	Time of the last port state change
macAddress	String	No	Switch port MAC address: “AA:BB:CC:DD:EE:FF”
ipv4Address	Object	No	

Attribute	Type	Mandatory	Description
address	String	Yes	IPv4 address
subnetMask	String	No	IPv4 subnet mask
addressOrigin	String	No	IPv4 address origin “DHCP”, “Static”
gateway	String	No	IPv4 gateway for this address

ipv6Address	Object	No	

Attribute	Type	Mandatory	Description
address	String	Yes	IPv6 address
prefixLength	Number	No	IPv6 Address Prefix Length
addressOrigin	String	No	IPv4 address origin “DHCP”, “Static”, “SLAAC”
addressState	String	No	IPv6 address state: “Preferred”, “Deprecated”, “Tentative”, “Failed”

neighborInfo	Object	No	

Attribute	Type	Mandatory	Description
switchIdentifier	String	No	Remote switch identifier
portIdentifier	String	No	Port identifier on the remote switch which is connected to a given local switch port
cableId	String	No	Cable identifier

neighborMac	String	No	MAC address of the remote port/interface
vlanEnable	Boolean	Yes	Indicates if VLANs are enabled on the switch port
defaultVlan	String	No	Default VLAN UUID
collections	Array: Object	Yes	Subcomponents collections

Attribute	Type	Mandatory	Description
name	String	Yes	Collection name
type	String	Yes	Collection type: ”EthernetSwitchPortVlans”,

						“EthernetSwitchPortMembers”, “Acls”, “StaticMacs”
			slotMask	String	No	Occupied slots mask (see section 4.5 for details)
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”}>,
                “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”: <string>,
                      “gateway”: <string>
                },
                “ipv6Address”: {
                      “address”: <string>,
                      “prefixLength”: <number>,
                      “addressOrigin”: <string>,
                      “addressState”: <string>
                },
                “neighborInfo”: {
                      “switchIdentifier”: <string>,
                      “portIdentifier”: <string>,
                      “cableId”: <string>
                },
                “neighborMac”: <string>,
                “vlanEnable”: <boolean>,
                “defaultVlan”: <string>,
```

```
                "collections": [
                        {
                                "name": <string>,
                        "type": <{"EthernetSwitchPortVlans",
                        "EthernetSwitchPortMembers", "Acls", "StaticMacs"}>,
                                "slotMask": <string>
                        }
                ...
                ],
                "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,
                "defaultVlan": "cd3e4527-af7b-32d3-a489-987656a3d588",
                "collections": [
                        {
                                "name": "EthernetSwtichPortVlans",
                                "type": "EthernetSwtichPortVlans"
                        },
                        {
                                "name": "Acls",
                                "type": "Acls"
                        },
                        {
                                "name": "StaticMacs",
                                "type": "StaticMacs"
                        }
                ],
                "oem": {}
        },
        "id": 987
}
```


5.44 Set Ethernet Switch Port Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the switch port attributes listed in the following table.

Table 79 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 speed and duplex is automatically configured by the port
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
oem	Object	OEM specific data

5.45 Add Ethernet Switch Port

Add Ethernet Switch Port command creates new logical port on the switch.

5.45.1 Request

Table 80 Add Ethernet Switch Port 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”
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
}
```

5.45.2 Response

Table 81 Add Ethernet Switch Port 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
}
```

5.46 Delete Ethernet Switch Port

Delete Ethernet Switch Port command destroys existing logical port on the switch.

5.46.1 Request

Table 82 Delete Ethernet Switch Port 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
}
```

5.46.2 Response

Table 83 Delete Ethernet Switch Port 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 Add Ethernet Switch Port Members

Add Ethernet Switch Port command adds a new member(s) to the existing logical port on the switch.

5.47.1 Request

Table 84 Add Ethernet Switch Port Members 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
}
```

5.47.2 Response

Table 85 Add Ethernet Switch Port Members 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 Delete Ethernet Switch Port Members

Delete Ethernet Switch Port command removes member(s) from the existing logical port on the switch.

5.48.1 Request

Table 86 Delete Ethernet Switch Port Members 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
}
```

5.48.2 Response

Table 87 Delete Ethernet Switch Port Members 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 Get Remote Ethernet Switch Info

Get Remote Ethernet Switch Info command retrieves detailed information about a single switch in the distributed network topology.

5.49.1 Request

Table 88 Get Remote Ethernet Switch Info 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
}
```

5.49.2 Response

Table 89 Get Remote Ethernet Switch Info response

Result	Type	Mandatory	Description
status	Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** state / String / Yes / Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline" health / String / Yes / Overall health state from the view of this resource: ""OK", "Warning", "Critical"
switchIdentifier	String	Yes	Switch identifier
macAddress	String	Yes	Switch MAC Address
nextHop	Array: Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** metric / Number / Yes / Cost associated with a given path to the remote switch portIdentifier / String / Yes / Identifier of a port through which a given remote switch is accessible ipv4Address / String / No / IPv4 address through which a given remote switch is accessible ipv6Address / String / No / IPv6 address through which a given remote switch is accessible
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
}
```

5.50 Get VLAN Info

Get VLAN Info command retrieves detailed information about a single VLAN configured within the switch. This command is optional and may not be implemented if VLANs are not explicitly created on the switch (they are created indirectly during creation of the port VLAN).

5.50.1 Request

Table 90 Get VLAN Info 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
}
```

5.50.2 Response

Table 91 Get VLAN Info request

<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>vlanId</td><td>Number</td><td>Yes</td><td>VLAN identifier</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>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>,
                “oem”: <object>
```

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

Example:

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

5.51 Set VLAN Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the VLAN attributes listed in the following table.

Table 92 Configurable VLAN Attributes

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

5.52 Add VLAN

Add VLAN command creates new VLAN on the switch. This command is optional and may not be implemented if VLANs are not explicitly created on the switch (they are created indirectly during creation of the port VLAN).

5.52.1 Request

Table 93 Add VLAN 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
}
```

5.52.2 Response

Table 94 Add VLAN 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
}
```

5.53 Delete VLAN

Delete VLAN command destroys existing VLAN on the switch. This command is optional and may not be implemented if VLANs are not explicitly created on the switch (they are created indirectly during creation of the port VLAN).

5.53.1 Request

Table 95 Delete VLAN 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
}
```

5.53.2 Response

Table 96 Delete VLAN 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
}
```

5.54 Get Port VLAN Info

Get Port VLAN Info command retrieves detailed information about a single VLAN configured on the switch port.

5.54.1 Request

Table 97 Get Port VLAN Info 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
}
```

5.54.2 Response

Table 98 Get Port VLAN Info 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>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
}
```

5.55 Set Port VLAN Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the Port VLAN attributes listed in the following table.

Table 99 Configurable Port VLAN Attributes

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

5.56 Add Port VLAN

Add Port VLAN command creates new VLAN on the switch port.

5.56.1 Request

Table 100 Add Port VLAN request

Parameters	Type	Mandatory	Description
port	String	Yes	Managed switch port UUID
vlanId	Number	Yes	New VALN Identifier
vlanName	String	Yes	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": "Management VLAN",
                "tagged": true,
                "oem": {}
        },
        "id": 987
}
```

5.56.2 Response

Table 101 Add Port VLAN 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
}
```

5.57 Delete Port VLAN

Delete Port VLAN command destroys existing VLAN on the switch port.

5.57.1 Request

Table 102 Delete Port VLAN 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
}
```

5.57.2 Response

Table 103 Delete Port VLAN 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
}
```

5.58 Get Port Static MAC Info

Get Port Static MAC Info command retrieves detailed information about a single Static MAC configured on the switch port.

5.58.1 Request

Table 104 Get Port Static MAC Info 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
}
```

5.58.2 Response

Table 105 Get Port Static MAC Info response

Result	Type	Mandatory	Description
status	Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** state / String / Yes / Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline" health / String / Yes / Overall health state from the view of this resource: "OK", "Warning", "Critical"
macAddress	String	Yes	Static MAC address for this port
vlanId	Number	No	VLAN identifier for this static MAC
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>,
                "oem": <object>
        },
        "id": <id>
}
```

Example:

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

5.59 Set Port Static MAC Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the Port Static MAC attributes listed in the following table.

Table 106 Configurable Port Static MAC Attributes

Attribute	Type	Description
vlanId	Number	VLAN identifier
oem	Object	OEM specific data

5.60 Add Port Static MAC

Add Port Static MAC command creates new Static MAC address on the switch port.

5.60.1 Request

Table 107 Add Port Static MAC 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
}
```

5.60.2 Response

Table 108 Add Port Static MAC 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
}
```

5.61 Delete Port Static MAC

Delete Port Static MAC command destroys existing Static MAC on the switch port.

5.61.1 Request

Table 109 Delete Port Static MAC 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
}
```

5.61.2 Response

Table 110 Delete Port Static MAC 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
}
```

5.62 Get ACL Info

Get ACL command retrieves detailed information about a single ACL configured on the switch.

5.62.1 Request

Table 111 Get ACL Info 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
}
```

5.62.2 Response

Table 112 Get ACL Info 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>
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></table>

			type	String	Yes	Collection type: ”Rules”, “EthernetSwitchPorts”
			slotMask	String	No	Occupied slots mask (see section 4.5 for details)
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”: <{“Rules”, “EthernetSwitchPorts”}>,
                               “slotMask”: <string>
                       }
                       ...
                ],
                “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
}
```

5.63 Add ACL

Add ACL command creates new ACL on the switch port.

5.63.1 Request

Table 113 Add ACL 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
}
```

5.63.2 Response

Table 114 Add ACL 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
}
```

5.64 Delete ACL

Delete ACL command removes the existing ACL.

5.64.1 Request

Table 115 Delete ACL 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
}
```

5.64.2 Response

Table 116 Delete ACL 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 Add ACL Port

Add ACL Port command binds the switch port(s) to the existing ACL.

5.65.1 Request

Table 117 Add ACL Port 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
}
```

5.65.2 Response

Table 118 Add ACL Port 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
}
```

5.66 Delete ACL Port

Delete ACL Port command unbounds the switch ports from the ACL.

5.66.1 Request

Table 119 Delete ACL Port 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
}
```

5.66.2 Response

Table 120 Delete ACL Port 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
}
```

5.67 Get ACL Rule Info

Get ACL Rule Info command retrieves detailed information about a single rule configured within the ACL.

5.67.1 Request

Table 121 Get ACL Rule Info 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
}
```

5.67.2 Response

Table 122 Get ACL Rule Info 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>
ruleId	Number	Yes	Rule identifier on the switch.
action	String	Yes	Type of action taken if a traffic match the rule: "Permit", "Deny", "Forward", "Mirror"
forwardMirrorPort	String	No	UUID of managed switch port which a 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	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr></table>

			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	No	MAC mask
sourceL4Port	Object	No	**Attribute**	**Type**	**Mandatory**	**Description**
			port	Number	Yes	L4 port
			mask	Number	No	L4 port mask
destinationL4Port	Object	No	**Attribute**	**Type**	**Mandatory**	**Description**
			port	Number	Yes	L4 port
			mask	Number	No	L4 port mask
protocol	Number	No	Rule L4 protocol identifier compatible with IP specification.			
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"}>
```

```
                },
                "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": {
                        "adress": <string>,
                        "mask": <string>
                },
                "destinationMac": {
                        "adress": <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
}
```

5.68 Add ACL Rule

Add ACL Rule command creates new rule on the ACL.

5.68.1 Request

Table 123 Add ACL Rule 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 a traffic match the rule: "Permit", "Deny", "Forward", "Mirror"
forwardMirrorPort	String	No	UUID of managed switch port which a 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
destinationL4Port	Object	No	**Attribute** / **Type** / **Mandatory** / **Description** port / Number / Yes / L4 port mask / Number / No / L4 port mask
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": {
                        "adress": <string>,
                        "mask": <string>
                },
                "destinationMac": {
                        "adress": <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
}
```

5.68.2 Response

Table 124 Add ACL Rule 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
}
```

5.69 Delete ACL Rule

Delete ACL Rule command removes a rule from the ACL.

5.69.1 Request

Table 125 Delete ACL Rule 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
}
```

5.69.2 Response

Table 126 Delete ACL Rule 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
}
```

5.70 Set ACL Rule Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the ACL rule attributes listed in the following table.

Table 127 Configurable ACL Rule Attributes

Attribute	Type	Description
action	String	Type of action taken if a traffic match the rule: "Permit", "Deny", "Forward", "Mirror"
forwardMirrorPort	String	UUID of managed switch port which a 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**	
		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				

5.71 Get Chassis Info

Get Chassis Info gathers the information about any object of the Chassis type. A Chassis can exist as a Rack, a Drawer, a module or a blade.

5.71.1 Request

Table 128 Get Chassis Info 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
}
```

5.71.2 Response

Table 129 Get Chassis Info response

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

			"Enabled", “Disabled”, “StandbyOffline”, “StandbySpare,”,“InTest”, “Starting”, “Absent, “UnavailableOffline”
			health: String, Yes — Overall health state from the view of this resource: “OK”, “Warning”, “Critical”
type	String	Yes	Chassis type: “Rack”, “Drawer”, “Module”, “Blade”, “Enclosure”, “StandAlone”, “RackMount”, “Card”, “Cartridge”, “Row”, “Pod”, “Expansion”, “Sidecar”, “Zone”, “Sled”, “Shelf”, “Component”, “Other”.
size	Number	Yes	Size in Parent Container’s Units (example: A chassis whose rack is the parent would likely be expressed in “Rack-U” units.)
locationOffset	Number	Yes	Location Offset 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	
sku	String	No	Switch SKU
assetTag	String	No	Asset tag
indicatorLED	String	No	The state of the indicator LED, used to identify the drive: “Lit”, “Blinking”, “Off”.
collections	Array: Object	Yes	Subcomponents collections
oem	Object	No	OEM specific data

Attribute	Type	Mandatory	Description
serialNumber	String	No	Module serial number
manufacturer	String	No	Manufacturer name
modelNumber	String	No	Model number
partNumber	String	No	Part number

Attribute	Type	Mandatory	Description
name	String	Yes	Collection name
type	String	Yes	Collection type: “Chassis”,”PowerZones”, “ThermalZones”, “Drives”
slotMask	String	No	Occupied slots mask (see section 4.5 for details)

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>,
```

```
                "locationOffset": <number>,
                "parentId": <string>,
                "powerZone": <string>,
                "thermalZone": <string>,
                "fruInfo": {
                        "serialNumber": <string>,
                        "manufacturer": <string>,
                        "modelNumber": <string>,
                        "partNumber": <string>
                },
                "indicatorLED": <{"Lit", "Blinking", "Off"}>,
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"Chassis","PowerZones",
"ThermalZones", "Drives"}>,
                                "slotMask": <string>
                        }
                ...
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "type": "Drawer",
                "size": 2,
                "locationOffset": 4,
                "parentId": "1",
                "powerZone": "123e4567-e89b-12d3-a456-426655440000",
                "thermalZone": "cd3e4527-af7b-32d3-a489-987656a3d588",
                "fruInfo": {
                        "serialNumber": "123fed3029c-b23394-12",
                        "manufacturer": "Intel Corporation",
                        "modelNumber": "E323",
                        "partNumber": "29ee2220939"
                },
                "indicatorLED": "Off",
                "collections": [
                        {
                                "name": "Power",
                                "type": "PowerZones"
                        },
                        {
                                "name": "Thermal",
                                "type": "ThermalZones"
                        },
```

```
                        {
                                "name": "Chassis",
                                "type": "Chassis"
                        },
                        {
                                "name": "Drives",
                                "type": "Drives"
                        }

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

5.72 Set Chassis Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the chassis in the following table.

Table 130 Set Chassis Attributes

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

5.73 Get Power Zone Info

The AMC retrieves full information about specific power zone by sending getPowerZoneInfo command to appropriated GAM Module.

5.73.1 Request

Table 131 Get Power Zone Info 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
```

```
}
```

5.73.2 Response

GAM Module replies with the following response in successful case:

Table 132 Get Power Zone Info response

Parameters	Type	Mandatory	Description
status	Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** state / String / Yes / Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline" health / String / Yes / Overall health state from the view of this resource: "OK", "Warning", "Critical"
powerInput	Number	Yes	Present power input
powerOutput	Number	Yes	Present power output
powerCapacity	Number	Yes	Max rated power capacity of the zone
collections	Array: Object	Yes	Subcomponents collections **Attribute** / **Type** / **Mandatory** / **Description** name / String / Yes / Collection name type / String / Yes / Collection type: "PSUs" slotMask / String / No / Occupied slots mask (see section 4.5 for details)
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"}>
                },
                "powerInput": <number>,
                "powerOutput": <number>,
                "powerCapacity": <number>,
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"PSUs"}>,
                                "slotMask": <string>
                        }
                …
                ]
                "oem": <object>
        },
```

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

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "powerInput": 5000,
                "powerOutput": 5000,
                "powerCapacity": 5000,
                "collections": [
                        {
                                "name": "PSUs",
                                "type": "PSUs",
                                "slotMask": "110110"
                        }
                ],
                "oem": {}
        },
        "id": 987
}
```

5.74 Set Power Zone Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the power zone attributes listed in the following table.

Table 133 Configurable Power Zone Attributes

Attribute	Type	Description
powerInput	Number	Present power input
powerOutput	Number	Present power output
oem	Object	OEM specific data

5.75 Get PSU Info

The AMC retrieves full information about specific power supply unit by sending getPsuInfo command to appropriated GAM Module.

5.75.1 Request

Table 134 Get PSU Info 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
}
```

5.75.2 Response

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

Table 135 Get PSU Info response

Parameters	Type	Mandatory	Description
status	Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** state / String / Yes / Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline" health / String / Yes / Overall health state from the view of this resource: "OK", "Warning", "Critical"
slot	Number	Yes	Zone slot number occupied by a PSU
powerInput	Number	Yes	Present power input
powerOutput	Number	Yes	Present power output
powerCapacity	Number	Yes	Max rated power capacity
fruInfo	Object	No	**Attribute** / **Type** / **Mandatory** / **Description** serialNumber / String / No / Module serial number manufacturer / String / No / Manufacturer name modelNumber / String / No / Model number partNumber / String / No / Part number
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"}>
                },
                "slot": <number>,
                "powerInput":  <number>,
```

```
                "powerOutput": <number>,
                "powerCapacity": <number>,
                "fruInfo": {
                        "serialNumber": <string>,
                        "manufacturer": <string>,
                        "modelNumber": <string>,
                        "partNumber": <string>
                },
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "slot": 1,
                "powerInput":  5000,
                "powerOutput": 5000,
                "powerCapacity": 5000,
                "fruInfo": {
                        "serialNumber": "123fed3029c-b23394-12",
                        "manufacturer": "Intel Corporation",
                        "modelNumber": "E323",
                        "partNumber": "29ee2220939"
                },
                "oem": {}
        },
        "id": 345
}
```

5.76 Set PSU Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the PSU attributes listed in the following table.

Table 136 Configurable PSU Attributes

Attribute	Type	Description
powerInput	Number	Present power input
powerOutput	Number	Present power output
oem	Object	OEM specific data

5.77 Get Thermal Zone Info

The AMC retrieves full information about specific thermal zone by sending getThermalZoneInfo command to appropriated GAM Module.

5.77.1 Request

Table 137 Get Thermal Zone Info 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
}
```

5.77.2 Response

GAM Module replies with the following response in successful case:

Table 138 Get Thermal Zone Info 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>
temperature	Number	Yes	Present temperature
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: "Fans"</td></tr><tr><td>slotMask</td><td>String</td><td>No</td><td>Occupied slots mask (see section 4.5 for details)</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"}>
                },
                "temperature": <number>,
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"Fans"}>,
                                "slotMask": <string>
                        }
                ...
                ],
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "temperature": 80,
                "collections": [
                        {
                                "name": "Fans",
                                "type": "Fans",
                                "slotMask": "110110"
                        }
                ],
                "oem": {}
        },
        "id": 345
}
```

5.78 Get Fan Info

The RMM asset manager can query a specific fan information by sending getFanInfo command to the appropriated GAM Module.

5.78.1 Request

Table 139 Get Fan Info 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
}
```

5.78.2 Response

GAM Module replies with the following response in successful case:

Table 140 Get Fan Info response

Parameters	Type	Mandatory	Description
status	Object	Yes	**Attribute** / **Type** / **Mandatory** / **Description** state / String / Yes / Known state of the component: "Enabled", "Disabled", "StandbyOffline", "StandbySpare,","InTest", "Starting", "Absent, "UnavailableOffline" health / String / Yes / Overall health state from the view of this resource: "OK", "Warning", "Critical"
slot	Number	Yes	Zone slot number occupied by a fan
currentSpeed	Number	Yes	Fan current speed
desiredSpeed	Number	Yes	Fan desired speed
fruInfo	Object	No	**Attribute** / **Type** / **Mandatory** / **Description** serialNumber / String / No / Module serial number manufacturer / String / No / Manufacturer name modelNumber / String / No / Model number partNumber / String / No / Part number
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"}>
                },
                "slot": <number>,
                "currentSpeed": <number>,
                "desiredSpeed": <number>
                "fruInfo": {
                       "serialNumber": <string>,
                       "manufacturer": <string>,
                       "modelNumber": <string>,
                       "partNumber": <string>
                },
                "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                       "state": "Enabled",
                       "health": "OK"
                },
                "slot": 1,
                "currentSpeed": 5000,
                "desiredSpeed": 5000
                "fruInfo": {
                       "serialNumber": "123fed3029c-b23394-12",
                       "manufacturer": "Intel Corporation",
                       "modelNumber": "E323",
                       "partNumber": "29ee2220939"
                },
                "oem": {}
        },
        "id": 987
}
```

5.79 Set Fan Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the fan attributes listed in the following table.

Table 141 Configurable Fan Attributes

Attribute	Type	Description
desiredSpeed	Number	Fan desired speed
oem	Object	OEM specific data

5.80 Get Authorization Certificate

The AMC is able to use authorization certificates obtained from underlying GAM Modules.

5.80.1 Request

Table 142 Get Authorization Certificate 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
}
```

5.80.2 Response

GAM Module replies with the following response in successful case:

Table 143 Get Authorization Certificate response

Parameters	Type	Mandatory	Description
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
}
```

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

5.81 Get Storage Services Info

Get Storage Services Info command retrieves information about the storage services.

5.81.1 Request

Table 144 Get Storage Services Info request

Parameters	Type	Mandatory	Description
services	String	Yes	Storage Services instance UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getStorageServicesInfo",
        "params": {
                "services": <string>
        },
        "id": <id>
}
```

Example:

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

5.81.2 Response

Table 145 Get Storage Services Info 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:</td></tr></table>

						"Enabled", “Disabled”, “StandbyOffline”, “StandbySpare,”,“InTest”, “Starting”, “Absent, “UnavailableOffline”
			health	String	Yes	Overall health state from the view of this resource: "OK", “Warning”, “Critical”
collections	Array: Object	Yes	Subcomponents collections			
			Attribute	**Type**	**Mandatory**	**Description**
			name	String	Yes	Collection name
			type	String	Yes	Collection type: “PhysicalDrives”, “LogicalDrives”, “iSCSITargets”
			slotMask	String	No	Occupied slots mask (see section 4.5 for details)
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”: <{“PhysicalDrives”, “LogicalDrives”,
“iSCSITargets”}>,
                                “slotMask”: <string>
                        }
                ...
                ]
                “oem”: <object>
        },
        “id”: <id>
}
```

Example:

```
{
        “jsonrpc”: “2.0”,
        “result”: {
                “status”: {
                        “state”: “Enabled”,
                        “health”: “OK”
                },
                “collections”: [
                        {
                                “name”: “PhysicalDrives”,
                                “type”: “PhysicalDrives”
```

```
                        },
                        {
                                "name": "LogicalDrives",
                                "type": "LogicalDrives"
                        },
                        {
                                "name": "iSCSITargets",
                                "type": "iSCSITargets"
                        }
                ],
                "oem": {}
        },
        "id": 987
}
```

5.82 Get Physical Drive Info

Get Physical Drive Info command retrieves detailed information about a single drive.

5.82.1 Request

Table 146 Get Physical Drive Info request

Parameters	Type	Mandatory	Description
drive	String	Yes	Managed physical drive UUID

Serialization:

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

Example:

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

5.82.2 Response

Table 147 Get Physical Drive Info response

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

			"Enabled", “Disabled”, “StandbyOffline”, “StandbySpare,”,“InTest”, “Starting”, “Absent, “UnavailableOffline”
			health: String, Yes — Overall health state from the view of this resource: ”OK", “Warning”, “Critical”
interface	String	Yes	Drive interface: "PCIe", “SAS”, “SATA”
type	String	No	Drive type: “HDD”, “SSD”, “NVMe”
capacityGB	Number	Yes	Drive capacity in GB
rpm	Number	No	For traditional drive, rotation per minute
fruInfo	Object	No	
oem	Object	No	OEM specific data

Attribute	Type	Mandatory	Description
serialNumber	String	No	Module serial number
manufacturer	String	No	Manufacturer name
modelNumber	String	No	Model number
partNumber	String	No	Part number

Serialization:

```
{
        “jsonrpc”: “2.0”,
        “result”: {
                “status”: {
                    “state”: <{"Enabled", “Disabled”, “StandbyOffline”,
                    “StandbySpare,”,“InTest”, “Starting”, “Absent,
                    “UnavailableOffline”}>,
                       “health”: <{"OK", “Warning”, “Critical”}>
                },
                “interface”: <string>
                “type”: <string>,
                “capacityGB”: <number>,
                “rpm”: <number>,
                “fruInfo”: {
                       “serialNumber”: <string>,
                       “manufacturer”: <string>,
                       “modelNumber”: <string>,
                       “partNumber”: <string>
                },
        “oem”: <object>
        },
        “id”: <id>
}
```

Example:

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

```
                "interface": "SATA"
                "type": "HDD",
                "capacityGB": 500,
                "rpm": 7200,
                "fruInfo": {
                        "serialNumber": "123fed3029c-b23394-12",
                        "manufacturer": "Intel Corporation",
                        "modelNumber": "E323",
                        "partNumber": "29ee2220939"
                },
                "oem": {}
        },
        "id": 987
}
```

5.83 Get Logical Drive Info

Get Logical Drive Info command retrieves detailed information about a single logical drive. The logical drive 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 logical drive may be defined on top of physical and logical drives.

5.83.1 Request

Table 148 Get Logical Drive Info request

Parameters	Type	Mandatory	Description
drive	String	Yes	Managed logical drive UUID

Serialization:

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

Example:

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

5.83.2 Response

Table 149 Get Logical Drive Info response

Result	Type	Mandatory	Description			
status	Object	Yes	**Attribute**	**Type**	**Mandatory**	**Description**

			<table><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	Logical drive type: “LVM”, “CEPH”, “RSTe”, “MDRAID”
capacityGB	Number	Yes	Logical drive capacity in GB
mode	String	No	Mode defines how the logical drive is built on top of underlying physical/logical drives. The value shall correspond to the logical drive type. Generic: “JBOD”, “BlockDevice”, “LogicalDrive” Logical Volume Manager: “PV”, “LV”, “LVG” CEPH: “OSD”, “Pool” RSTe, MDRAID: “RAID0Array”, “RAID1Array”, “RAID10Array”, “RAID5Array”, “RAID6Array”, “RAID50Array”, “RAID60Array”
master	String	No	UUID of the master drive used as an image source
snapshot	Boolean	No	Indicates if the logical drive should has been created as a snapshot of the source master drive or a full copy of an image from the source master drive.
image	String	No	Information about an image populated on the logical drive
bootable	Boolean	No	Indicates if the logical drive is populated with a bootable image
protected	Boolean	Yes	Indicated if the logical drive is write/delete protected. Such a drive cannot be neither deleted nor written (e.g. it cannot be exposed using a remote target nor used to build another logical drive). All drives being the master drives (i.e. used to be a source of a drive image for other drive during clone action) shall be protected using this flag.
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: “PhysicalDrives”, “LogicalDrives”</td></tr><tr><td>slotMask</td><td>String</td><td>No</td><td>Occupied slots mask (see section 4.5 for details)</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": <{"LVM", "CEPH", "RSTe", "MDRAID"}>,
                "capacityGB": <number>,
                "mode": <{"JBOD", "BlockDevice", "LogicalDrive", "PV", "LV",
                 "LVG", "OSD", "Pool", "RAID0Array", "RAID1Array",
                 "RAID10Array", "RAID5Array", "RAID6Array", "RAID50Array",
                 "RAID60Array"}>,
                "master": <string>,
                "snapshot": <boolean>,
                "image": <string>,
                "bootable": <boolean>,
                "protected": <boolean>,
                "collections": [
                        {
                                "name": <string>,
                                "type": <{"PhysicalDrives", "LogicalDrives"}>,
                                "slotMask": <string>
                        }
                ...
                ]
        "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "result": {
                "status": {
                        "state": "Enabled",
                        "health": "OK"
                },
                "type": "LVM",
                "capacityGB": 100,
                "mode": "LV",
                "master": "863e4567-e87b-64d3-a489-987656540000",
                "snapshot": false,
                "image": "OpenStack Juno Ubuntu 15.04",
                "bootable": true,
                "protected": false,
                "collections": [
                        {
                                "name": "LogicalDrives",
                                "type": "LogicalDrives"
                        }
                ]
                "oem": {}
        },
        "id": 987
}
```

5.84 Add Logical Drive

Add Logical Drive command creates new logical drive within the storage infrastructure hierarchy. Optionally it may populate newly created logical drive with the image from existing logical drive (e.g. cloning operation).

5.84.1 Request

Table 150 Add Logical Drive request

Result	Type	Mandatory	Description
type	String	Yes	Logical drive type: “LVM”, “CEPH”, “RSTe”, “MDRAID”
capacityGB	Number	Yes	Logical drive capacity in GB
mode	String	Yes	Mode defines how the logical drive is built on top of underlying physical/logical drives. The value shall correspond to the logical drive type. Generic: “JBOD”, “BlockDevice”, “LogicalDrive” Logical Volume Manager: “PV”, “LV”, “LVG” CEPH: “OSD”, “Pool” RSTe, MDRAID: “RAID0Array”, “RAID1Array”, “RAID10Array”, “RAID5Array”, “RAID6Array”, “RAID50Array”, “RAID60Array”
master	String	Yes	UUID of the master drive used as an image source. If this information is not provided then a clean drive will be created
snapshot	Boolean	Yes	Indicates if the logical drive should be created as a snapshot of the source master drive or should be created as a full copy of an image from the source master drive.
image	String	No	Information about an image populated on the logical drive
protected	Boolean	Yes	Indicates if the logical drive should be marked as write/delete protected after creation.
drives	Array: String	Yes	The UUIDs of the drives (physical or logical) that should be used to create the new logical drive on top
oem	Object	No	OEM specific data

Serialization:

```
{
        “jsonrpc”: “2.0”,
        “method”: “addLogicalDrive”,
        “params”: {
                “type”: <{“LVM”, “CEPH”, “RSTe”, “MDRAID”}>,
                “capacityGB”: <number>,
                “mode”: <{“JBOD”, “BlockDevice”, “LogicalDrive”, “PV”,  “LV”,
                 “LVG”, “OSD”, “Pool”, “RAID0Array”, “RAID1Array”,
                 “RAID10Array”, “RAID5Array”, “RAID6Array”, “RAID50Array”,
                 “RAID60Array”}>,
                “master”: <string>,
                “snapshot”: <boolean>,
                “image”: <string>,
                “protected”: <boolean>,
                “drives”: [
                      <string>,
                      ...
                ]
```

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

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addLogicalDrive",
        "params": {
                "type": "LVM",
                "capacityGB": 100,
                "mode": "LV",
                "master": "863e4567-e87b-64d3-a489-987656540000",
                "snapshot": false,
                "image": "OpenStack Juno Ubuntu 15.04",
                "protected": false,
                "drives": [
                        "cd3e4527-af7b-32d3-a489-987656a3d588"
                ],
                "oem": {}
        },
        "id": 987
}
```

5.84.2 Response

Table 151 Add Logical Drive response

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

Serialization:

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

Example:

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

5.85 Delete Logical Drive

Delete Logical Drive command destroys existing logical drive. This command will fail during an attempt at deleting of the protected logical drive.

5.85.1 Request

Table 152 Delete Logical Drive request

Parameters	Type	Mandatory	Description
drive	String	Yes	Managed logical drive UUID

Serialization:

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

Example:

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

5.85.2 Response

Table 153 Delete Logical Drive 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
}
```

5.86 Set Logical Drive Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the Logical Drive attributes listed in the following table.

Table 154 Configurable Logical Drive Attributes

Attribute	Type	Description
bootable	Boolean	True if drive is bootable device. Otherwise false.
oem	Object	OEM specific data

5.87 Get iSCSI Target Info

Get iSCSI Target Info command retrieves detailed information about a single iSCSI remote target configured on top of one or more logical drives.

5.87.1 Request

Table 155 Get iSCSI Target Info request

Parameters	Type	Mandatory	Description
target	String	Yes	Managed iSCSI target UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "getiSCSITargetInfo",
        "params": {
                "target": <string>
        },
        "id": <id>
}
```

Example:

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

5.87.2 Response

Table 156 Get iSCSI Target Info response

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

			health | String | Yes | Overall health state from the view of this resource: "OK", “Warning”, “Critical”
initiatorIQN	String	No	iSCSI initiator IQN. If this attribute has a value it means that it is currently assigned and being used by a composed node
targetAddress	String	Yes	iSCSI target IP address
targetPort	String	Yes	iSCSI target port
targetIQN	String	Yes	iSCSI target IQN
targetLUNs	Array: Object	Yes	Logical drives accessible through iSCSI target **Attribute** | **Type** | **Mandatory** | **Description** LUN | Number | Yes | iSCSI target LUN logicalDrive | String | Yes | Logical drive UUID
authenticationMethod	String	Yes	Supported values: “OneWay”, “Mutual”. This field is also nullable.
chapUsername	String	No	CHAP security username
chapSecret	String	No	CHAP security secret
mutualChapUsername	String	No	Mutual CHAP security username
mutualChapSecret	String	No	Mutual CHAP security secret
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”}>
                },
                “initiatorIQN”: <string>,
                “targetAddress”: <string>,
                “targetPort”: <number>,
                “targetIQN”: <string>
                “targetLUNs”: [{
                       “LUN”: <number>,
                       “logicalDrive”: <string>
                       },
                       ...
                ],
        “oem”: <object>
        },
        “id”: <id>
}
```

Example:

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

```
                },
                "initiatorIQN": "iqn.2015-01.com:pod-1,rack-1,drawer-1,module-
3",
                "targetAddress": "10.102.44.54",
                "targetPort": 3260,
                "targetIQN": "iqn.2015-01.com:ceph-ubuntu14",
                "targetLUNs": [{
                      "LUN": 1,
                      "logicalDrive": "cd3e4527-af7b-32d3-a489-987656a3d588"
                      }
                ],
                "oem": {}
        },
        "id": 987
}
```

5.88 Set iSCSI Target Attributes

The *Set Component Attributes* described in the section 5.11 allows configuration of the iSCSI target attributes listed in the following table.

Table 157 Configurable iSCSI Target Attributes

Attribute	Type	Description
initiatorIQN	String	iSCSI initiator IQN
authenticationMethod	String	Supported values: "OneWay", "Mutual". This field is also nullable.
chapUsername	String	CHAP security username
chapSecret	String	CHAP security secret
mutualChapUsername	String	Mutual CHAP security username
mutualChapSecret	String	Mutual CHAP security secret
oem	Object	OEM specific data

5.89 Add iSCSI Target

Add iSCSI Target Info command creates new iSCSI remote target on top of one or more logical drives.

5.89.1 Request

Table 158 Add iSCSI Target request

Result	Type	Mandatory	Description
initiatorIQN	String	No	iSCSI initiator IQN. If this attribute has a value it means that it is currently allocated.
targetIQN	String	Yes	iSCSI target IQN
targetLUNs	Array: Object	Yes	Logical drives accessible through iSCSI target **Attribute** / **Type** / **Mandatory** / **Description** LUN / Number / Yes / iSCSI target LUN logicalDrive / String / Yes / Logical drive UUID
authenticationMethod	String	Yes	Supported values: "OneWay", "Mutual". This field is also nullable.
chapUsername	String	No	CHAP security username
chapSecret	String	No	CHAP security secret
mutualChapUsername	String	No	Mutual CHAP security username

mutualChapSecret	String	No	Mutual CHAP security secret
oem	Object	No	OEM specific data

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "addiSCSITarget",
        "params": {
                "initiatorIQN": <string>,
                "targetIQN": <string>
                "targetLUNs": [{
                        "LUN": <number>,
                        "logicalDrive": <string>
                        },
                        ...
                ],
        "oem": <object>
        },
        "id": <id>
}
```

Example:

```
{
        "jsonrpc": "2.0",
        "method": "addiSCSITarget",
        "params": {
                "initiatorIQN": "iqn.2015-01.com:pod-1,rack-1,drawer-1,module-
3",
                "targetIQN": "iqn.2015-01.com:ceph-ubuntu14",
                "targetLUNs": [{
                        "LUN": 1,
                        "logicalDrive": "cd3e4527-af7b-32d3-a489-987656a3d588"
                        }
                ],
                "oem": {}
        },
        "id": 987
}
```

5.89.2 Response

Table 159 Add iSCSI Target response

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

Serialization:

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

```
}
```

Example:

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

5.90 Delete iSCSI Target

Delete iSCSI Target command destroys existing iSCSI target. This command removes the iSCSI target only, the corresponding logical drive(s) will not be modified.

5.90.1 Request

Table 160 Delete iSCSI Target request

Parameters	Type	Mandatory	Description
target	String	Yes	Managed iSCSI target UUID

Serialization:

```
{
        "jsonrpc": "2.0",
        "method": "deleteiSCSITarget",
        "params": {
                "target": <string>
        },
        "id": <id>
}
```

Example:

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

5.90.2 Response

Table 161 Delete iSCSI Target 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
}
```

§

6 GAMI Commands Support Requirements

This specification defines for types of GAM modules:

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

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

The following table provide information about GAMI commands support requirements for different types of GAM modules.

Table 162 GAMI Commands Support Requirements

Command	Compute	Storage Services	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				
Get Memory Info	Mandatory				
Get Storage Controller Info	Mandatory				
Get Storage Subsystem Info	Mandatory				Optional
Set Storage Subsystem Attributes	Optional				Optional
Get Drive Info				Mandatory	Optional
Set Drive Attributes				Optional	
Erase Drive Securely				Optional	
Get Network Interface Info	Mandatory				
Set Network Interface Attributes	Optional				
Get Switch Info					Mandatory
Get Port Info					Mandatory
Set Port Attributes					Mandatory
Get PCIe Device Info					Mandatory
Get PCIe Function Info					Mandatory
Get Zone Info					Mandatory
Add Zone					Mandatory
Delete Zone					Mandatory
Add Zone Endpoints					Mandatory

Command	Compute	Storage Services	Network	Chassis	PCIe Fabric
Delete Zone Endpoints					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	
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	

[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).
[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)

Command	Compute	Storage Services	Network	Chassis	PCIe Fabric
Get Storage Services Info		Mandatory			
Get Physical Drive Info		Mandatory			
Get Logical Drive Info		Mandatory			
Add Logical Drive		Mandatory			
Delete Logical Drive		Mandatory			
Get iSCSI Target Info		Mandatory			
Set iSCSI Target Attributes		Mandatory			
Add iSCSI Target		Mandatory			
Delete iSCSI Target request		Mandatory			

§