



Intel® Rack Scale Design Generic Assets Management Interface

API Specification Version 1.2

July 2016

Revision 004



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

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.

This document contains information on products, services, and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications, and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

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 © 2016 Intel Corporation. All rights reserved.



Contents

1	Introduction	12
1.1	Scope	12
1.2	Intended audience	12
1.3	Terminology	12
1.4	References	13
1.5	Notes and symbol convention	13
2	Generic Asset Management Interface	14
2.1	GAMI API architecture and design principles	14
3	GAMI API Error Handling	16
3.1	API error response	16
3.1.1	Example error JSON object	16
3.2	API error codes	16
3.2.1	General error codes	16
4	GAMI Data Model	17
4.1	Compute System management module	17
4.2	Storage Services management module	17
4.3	Network management model	18
4.4	Chassis management model	18
4.5	Subcomponent collections	18
4.6	Mandatory and optional properties	19
5	GAMI API Definition	20
5.1	GAM module registration	20
5.1.1	Request	21
5.1.2	Response	22
5.2	Heartbeat	22
5.2.1	Request	22
5.2.2	Response	23
5.3	Get manager collection	23
5.3.1	Request	23
5.3.2	Response	24
5.4	Get manager info	24
5.4.1	Request	24
5.4.2	Response	25
5.5	Get collection	29
5.5.1	Request	29
5.5.2	Response	29
5.6	Component notification	30
5.6.1	Notification	30
5.7	Set component attributes	31
5.7.1	Request	31
5.7.2	Response	32
5.8	Get computer system info	32
5.8.1	Request	32
5.8.2	Response	33
5.9	Set computer system attributes	36
5.10	Get processor info	36



5.10.1	Request.....	37
5.10.2	Response	37
5.11	Get memory chunk Info	39
5.11.1	Request.....	40
5.11.2	Response	40
5.12	Get Dimm info.....	41
5.12.1	Request.....	42
5.12.2	Response	42
5.13	Get storage controller info.....	46
5.13.1	Request.....	46
5.13.2	Response	47
5.14	Get drive info.....	48
5.14.1	Request.....	48
5.14.2	Response	49
5.15	Get network interface Info	50
5.15.1	Request.....	50
5.15.2	Response	51
5.16	Set network interface attributes	53
5.17	Get Ethernet switch info	54
5.17.1	Request.....	54
5.17.2	Response	54
5.18	Get Ethernet switch port info	57
5.18.1	Request.....	57
5.18.2	Response	57
5.19	Set Ethernet switch port attributes	61
5.20	Add Ethernet switch port	61
5.20.1	Request.....	61
5.20.2	Response	62
5.21	Delete Ethernet switch port	63
5.21.1	Request.....	63
5.21.2	Response	63
5.22	Add Ethernet switch port members	64
5.22.1	Request.....	64
5.22.2	Response	65
5.23	Delete Ethernet switch port members	65
5.23.1	Request.....	65
5.23.2	Response	66
5.24	Get remote Ethernet switch info	66
5.24.1	Request.....	66
5.24.2	Response	67
5.25	Get VLAN info	68
5.25.1	Request.....	68
5.25.2	Response	69
5.26	Set VLAN attributes.....	70
5.27	Add VLAN	70
5.27.1	Request.....	70
5.27.2	Response	71
5.28	Delete VLAN	71
5.28.1	Request.....	71
5.28.2	Response	72
5.29	Get port VLAN info	72



5.29.1	Request.....	72
5.29.2	Response	73
5.30	Set port VLAN attributes.....	74
5.31	Add port VLAN	74
5.31.1	Request.....	74
5.31.2	Response	75
5.32	Delete port VLAN	75
5.32.1	Request.....	75
5.32.2	Response	76
5.33	Get port static MAC info.....	76
5.33.1	Request.....	77
5.33.2	Response	77
5.34	Set port static MAC attributes	78
5.35	Add port static MAC	78
5.35.1	Request.....	78
5.35.2	Response	79
5.36	Delete port static MAC.....	79
5.36.1	Request.....	79
5.36.2	Response	80
5.37	Get ACL info.....	80
5.37.1	Request.....	81
5.37.2	Response	81
5.38	Add ACL.....	82
5.38.1	Request.....	82
5.38.2	Response	83
5.39	Delete ACL.....	84
5.39.1	Request.....	84
5.39.2	Response	84
5.40	Add ACL port.....	85
5.40.1	Request.....	85
5.40.2	Response	86
5.41	Delete ACL port.....	86
5.41.1	Request.....	86
5.41.2	Response	87
5.42	Get ACL rule info	87
5.42.1	Request.....	87
5.42.2	Response	88
5.43	Add ACL rule	90
5.43.1	Request.....	90
5.43.2	Response	93
5.44	Delete ACL rule	94
5.44.1	Request.....	94
5.44.2	Response	94
5.45	Set ACL rule attributes.....	95
5.46	Get chassis info	96
5.46.1	Request.....	96
5.46.2	Response	96
5.47	Get power zone info.....	98
5.47.1	Request.....	98
5.47.2	Response	99
5.48	Set power zone attributes.....	100



5.49	Get PSU info	100
5.49.1	Request.....	101
5.49.2	Response	101
5.50	Set PSU attributes.....	102
5.51	Get thermal zone info	103
5.51.1	Request.....	103
5.51.2	Response	103
5.52	Get fan info.....	104
5.52.1	Request.....	105
5.52.2	Response	105
5.53	Set fan attributes.....	106
5.54	Get authorization certificate	107
5.54.1	Request.....	107
5.54.2	Response	107
5.55	Get storage services info	108
5.55.1	Request.....	108
5.55.2	Response	109
5.56	Get physical drive info	110
5.56.1	Request.....	110
5.56.2	Response	111
5.57	Get logical drive info.....	112
5.57.1	Request.....	112
5.57.2	Response	113
5.58	Add logical drive.....	115
5.58.1	Request.....	115
5.58.2	Response	116
5.59	Delete logical drive.....	117
5.59.1	Request.....	117
5.59.2	Response	117
5.60	Get iSCSI target Info.....	118
5.60.1	Request.....	118
5.60.2	Response	118
5.61	Set iSCSI target attributes	119
5.62	Add iSCSI target.....	120
5.62.1	Request.....	120
5.62.2	Response	121
5.63	Delete iSCSI target.....	121
5.63.1	Request.....	121
5.63.2	Response	122
6	GAMI Commands Support Requirements.....	123

Figures

Figure 1	Generic Assets Management Interface in the PSME architecture	14
Figure 2	Compute System management data model.....	17
Figure 3	Storage Services management data model.....	17
Figure 4	Network management data model.....	18
Figure 5	Chassis management data model	18
Figure 6	PSME Generic Asset Management Interface API communication overview.....	20



Tables

Table 1	Terminology	12
Table 2	General error codes	16
Table 3	GAMI subcomponent collection definition	18
Table 4	GAM module registration request	21
Table 5	GAM module registration response	22
Table 6	Heartbeat command request	22
Table 7	Heartbeat command response	23
Table 8	Get manager collection response	24
Table 9	Get manager info request	24
Table 10	Get manager info response	25
Table 11	Get collection request	29
Table 12	Get collection response	29
Table 13	Component notification	30
Table 14	Set component attributes request	31
Table 15	Get computer system info request	32
Table 16	Get Computer System Info response	33
Table 17	Configurable computer system attributes	36
Table 18	Get processor info request	37
Table 19	Get processor info response	37
Table 20	Get memory chunk Info request	40
Table 21	Get memory chunk Info response	40
Table 22	Get Dimm info request	42
Table 23	Get Dimm info response	42
Table 24	Get storage controller info request	46
Table 25	Get storage controller info response	47
Table 26	Get drive info request	48
Table 27	Get drive info response	49
Table 28	Get network interface info request	50
Table 29	Get network interface info response	51
Table 30	Configurable network interface attributes	54
Table 31	Get Ethernet switch info request	54
Table 32	Get Ethernet switch info response	54
Table 33	Get Ethernet switch port info request	57
Table 34	Get Ethernet switch port info response	57
Table 35	Configurable Ethernet Switch Port Attributes	61
Table 36	Add Ethernet switch Port request	61
Table 37	Add Ethernet switch port response	62
Table 38	Delete Ethernet switch port request	63
Table 39	Delete Ethernet switch port response	63
Table 40	Add Ethernet switch port members request	64
Table 41	Add Ethernet switch port members response	65
Table 42	Delete Ethernet switch port members request	65
Table 43	Delete Ethernet switch port members response	66
Table 44	Get remote Ethernet switch info request	66
Table 45	Get remote Ethernet switch info response	67
Table 46	Get VLAN info request	68
Table 47	Get VLAN info request	69
Table 48	Configurable VLAN attributes	70



Table 49	Add VLAN request	70
Table 50	Add VLAN response	71
Table 51	Delete VLAN request	71
Table 52	Delete VLAN response	72
Table 53	Get Port VLAN Info request	72
Table 54	Get port VLAN info response	73
Table 55	Configurable port VLAN attributes	74
Table 56	Add Port VLAN request	74
Table 57	Add port VLAN response	75
Table 58	Delete port VLAN request	75
Table 59	Delete port VLAN response	76
Table 60	Get port static MAC Info request	77
Table 61	Get port static MAC info response	77
Table 62	Configurable port static MAC attributes	78
Table 63	Add port static MAC request	78
Table 64	Add port static MAC response	79
Table 65	Delete Port Static MAC request	79
Table 66	Delete port static MAC response	80
Table 67	Get ACL info request	81
Table 68	Get ACL info response	81
Table 69	Add ACL request	82
Table 70	Add ACL response	83
Table 71	Delete ACL request	84
Table 72	Delete ACL response	84
Table 73	Add ACL port request	85
Table 74	Add ACL port response	86
Table 75	Delete ACL port request	86
Table 76	Delete ACL port response	87
Table 77	Get ACL rule info request	87
Table 78	Get ACL rule info response	88
Table 79	Add ACL rule request	90
Table 80	Add ACL rule response	93
Table 81	Delete ACL rule request	94
Table 82	Delete ACL rule response	94
Table 83	Configurable ACL rule attributes	95
Table 60	Get chassis info request	96
Table 61	Get chassis info response	96
Table 62	Get power zone info request	98
Table 63	Get power zone info response	99
Table 64	Configurable Power Zone Attributes	100
Table 65	Get PSU Info request	101
Table 66	Get PSU info response	101
Table 67	Configurable PSU Attributes	102
Table 68	Get thermal zone info request	103
Table 69	Get Thermal Zone Info response	103
Table 70	Get fan info request	105
Table 71	Get fan info response	105
Table 72	Configurable Fan Attributes	106
Table 73	Get authorization certificate request	107
Table 74	Get authorization certificate response	107



Table 75	Get storage services info request.....	108
Table 76	Get storage services info response.....	109
Table 77	Get physical drive info request.....	110
Table 78	Get drive info response.....	111
Table 79	Get logical drive info request	112
Table 80	Get logical drive info response	113
Table 81	Add Logical Drive request.....	115
Table 82	Add logical drive response	116
Table 83	Delete logical drive request	117
Table 84	Delete logical drive response	117
Table 85	Get iSCSI target info request	118
Table 86	Get iSCSI target info response.....	118
Table 87	Configurable iSCSI Target Attributes.....	120
Table 88	Add iSCSI target request	120
Table 89	Add iSCSI target response.....	121
Table 90	Delete iSCSI target request	121
Table 91	Delete iSCSI target response	122
Table 92	GAMI commands support requirements.....	123



Revision History

Revision	Description	Date
1.07	Merged engineering GAMI revision 6 into new document template.	June 23, 2016
1.06	Changed vlanName field to optional.	June 1, 2016
1.05	Changed Static MAC vlanId field to optional.	May 31, 2016
1.04	Static MAC functionality added	May 5, 2016
1.03	Modified resource definition: ACL Rule	April 19, 2016
1.01	Access Control List functionality added	April 01, 2016
1.0	Final document for 1.2 SW release	April 01, 2016
0.91	GamiID handling changes during module registration	March 31, 2016
0.90	Final document for 1.2 SW release	March 13, 2016
0.83	Proposal updated	March 01, 2016
0.82	Final document for 1.2 SW release - proposal	January 18, 2016
0.81	New memory model added	January 11, 2016
0.80	Initial version aligned with Chinook extensions to Redfish specification	November 30, 2015
0.79	Modified resource definition: processor	November 24, 2015
0.78	Modified resource definition: manager, remote switch, chassis	November 16, 2015
0.77	Modified resource definition: processor, memory, switch port, network interface	October 27, 2015
0.76	GetAuthorizationCertificate command added	October 09, 2015
0.75	Modified resource definition: blade, processor, memory, network interface, storage controller, drive	October 05, 2015
0.74	Switch port commands changed	September 18, 2015
0.73	Switch port default VLAN configuration changed	September 11, 2015
0.72	Component Notification command modified, Set Component Attributes command changed	September 07, 2015
0.71	Link Aggregation support added	September 02, 2015
0.70	Merged with obsoleted version 0.66	August 31, 2015
0.62	GAMI Commands Support Requirements updated	July 28, 2015
0.61	Aligned with IDF release.	July 06, 2015
0.60	Heart Beat command added, Add/Remove/Update Component commands added, Location property added to the module and blade	July 01, 2015
0.59	VlanPort commands changed, boot option added to the blade definition with set command, iSCSI target commands modified.	June 09, 2015
0.58	Initial storage service command set added.	May 20, 2015
0.57	Added: Delete VLAN, Delete VLAN Port, Set Switch Port Attributes.	May 12, 2015
0.55	Unification of RMM commands.	April 30, 2015
0.52	Update RMM index into ID.	April 29, 2015
0.51	Initial merge for PSME and RMM JSON-RPC spec.	April 21, 2015
0.5	Initial revision ready for external publication.	April 17, 2015
0.45	Added Get Chassis Info, and Get Slot Zone Info.	April 16, 2015
0.44	Switch Port data structure modified.	April 15, 2015
0.43	"Get VLAN Info" and "Add VLAN" commands added; "GAMI Commands Support Requirements" section added.	April 14, 2015
0.42	Info about fields mandatory added.	April 13, 2015
0.41	Command parameters description updated.	April 10, 2015
0.4	Preliminary completed revision, ready for review.	April 9, 2015
0.3	Network and storage assets management commands added.	March 31, 2015
0.2	Completed compute assets management command.	March 13, 2015



0.1	First internal draft.	February 20, 2015
-----	-----------------------	-------------------

§



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 1.0 Release for the Bulldog Creek SDV.

1.2 Intended audience

The intended audiences for this document include designers and engineers working with the Software 1.0 Release, porting this software to HW platforms other than the Bulldog Creek SDV.

1.3 Terminology

Table 1 Terminology

Term	Definition
ACL	Access Control List
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
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
332868	Intel® Rack Scale Design GAMI API Specification	http://intel.com/intelRSD
332869	Intel® Rack Scale Design Pod Manager REST API Specification	http://intel.com/intelRSD
332870	Intel® Rack Scale Design Pod Manager Release Notes	http://intel.com/intelRSD
332871	Intel® Rack Scale Design Pod Manager User Guide	http://intel.com/intelRSD
332873	Intel® Rack Scale Design PSME REST API Specification	http://intel.com/intelRSD
332872	Intel® Rack Scale Design PSME Release Notes	http://intel.com/intelRSD
332874	Intel® Rack Scale Design PSME User Guide	http://intel.com/intelRSD
332877	Intel® Rack Scale Design RMM REST API Specification	http://intel.com/intelRSD
332876	Intel® Rack Scale Design RMM Release Notes	http://intel.com/intelRSD
332875	Intel® Rack Scale Design RMM User Guide	http://intel.com/intelRSD
332878	Intel® Rack Scale Design Storage Services API Specification	http://intel.com/intelRSD
332936	Intel® Rack Scale Design BIOS/BMC Tech Guide	http://intel.com/intelRSD
332937	Intel® Rack Scale Design Architectural Requirements Specification	http://intel.com/intelRSD
334611	Intel® Rack Scale Design Getting Started Guide	http://intel.com/intelRSD
n/a	Scalable Platforms Management API	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.

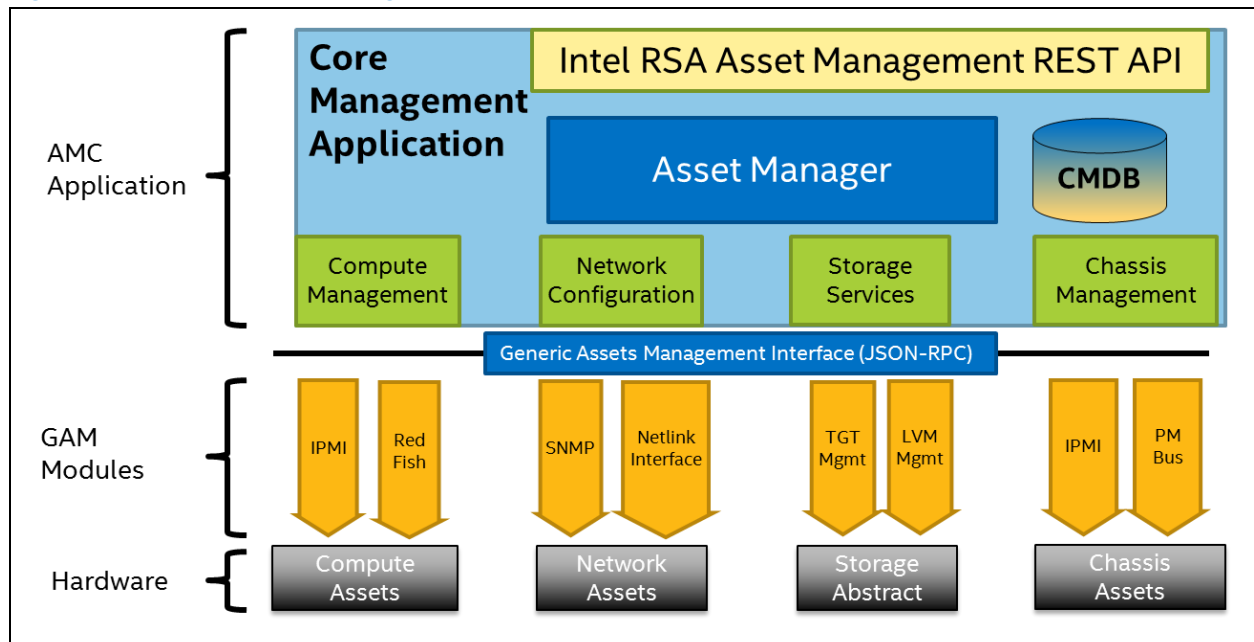


2 Generic Asset Management Interface

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
}
```

§



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 blade"
  },
  "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

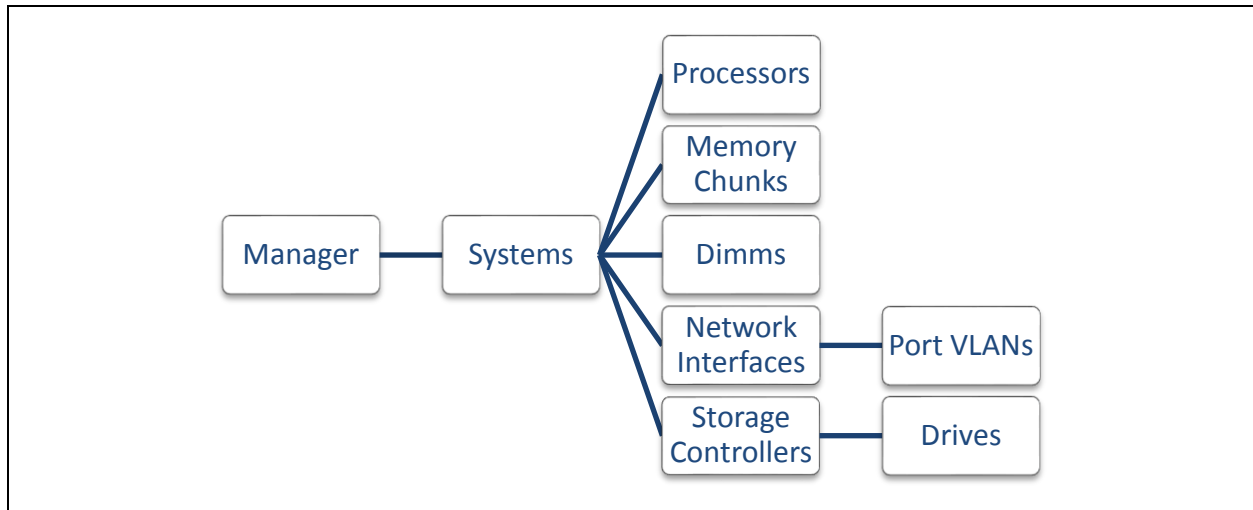
Error Code	Message	Description
-32700	Parse Error	The server received invalid JSON. An error occurred on the server while parsing the JSON text.
-32600	Invalid Request	The JSON sent is not a valid Request object.
-32601	Method not found	The method does not exist / is not available.
-32602	Invalid Parameters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

§

4 GAMI Data Model

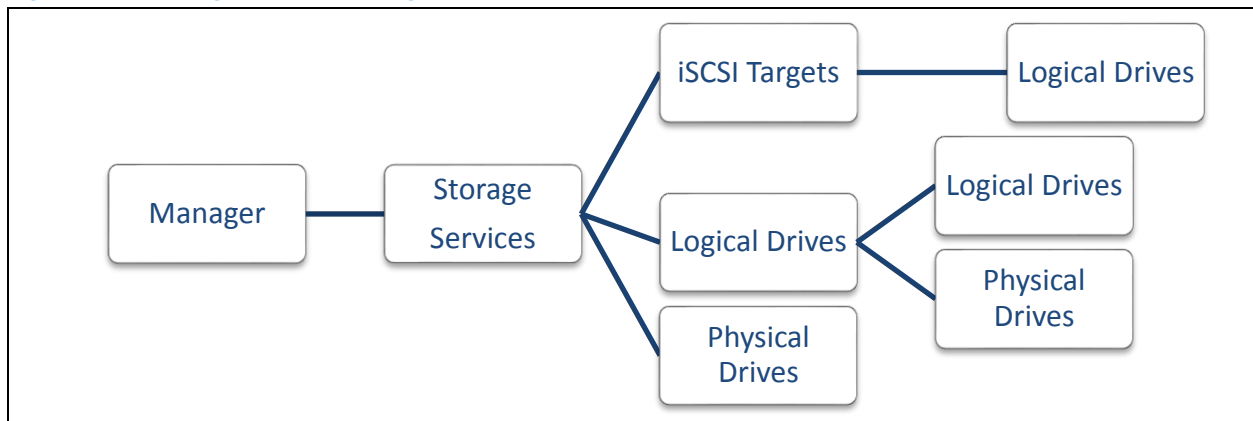
4.1 Compute System management module

Figure 2 Compute System management data model



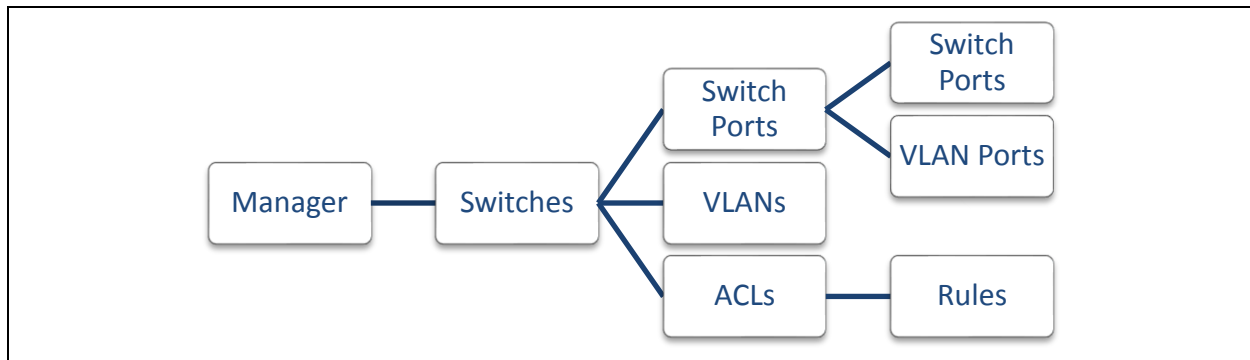
4.2 Storage Services management module

Figure 3 Storage Services management data model



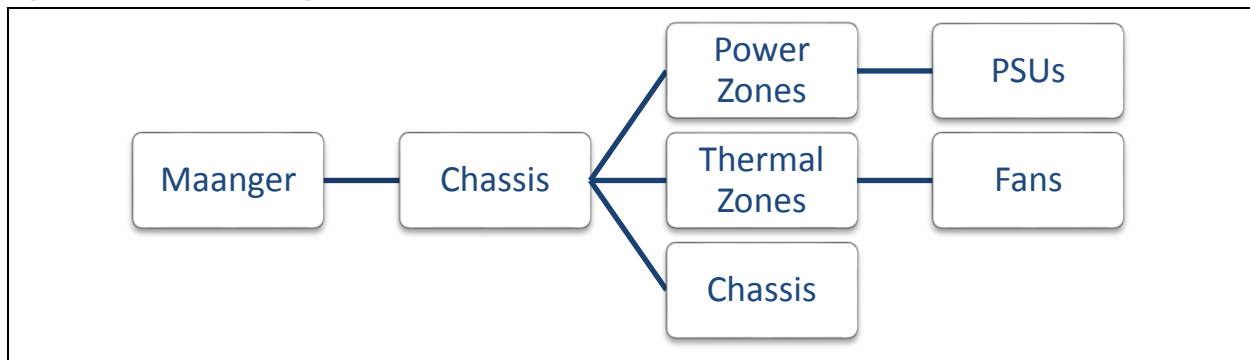
4.3 Network management model

Figure 4 Network management data model



4.4 Chassis management model

Figure 5 Chassis management data model



4.5 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.

4.6 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 structure 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.

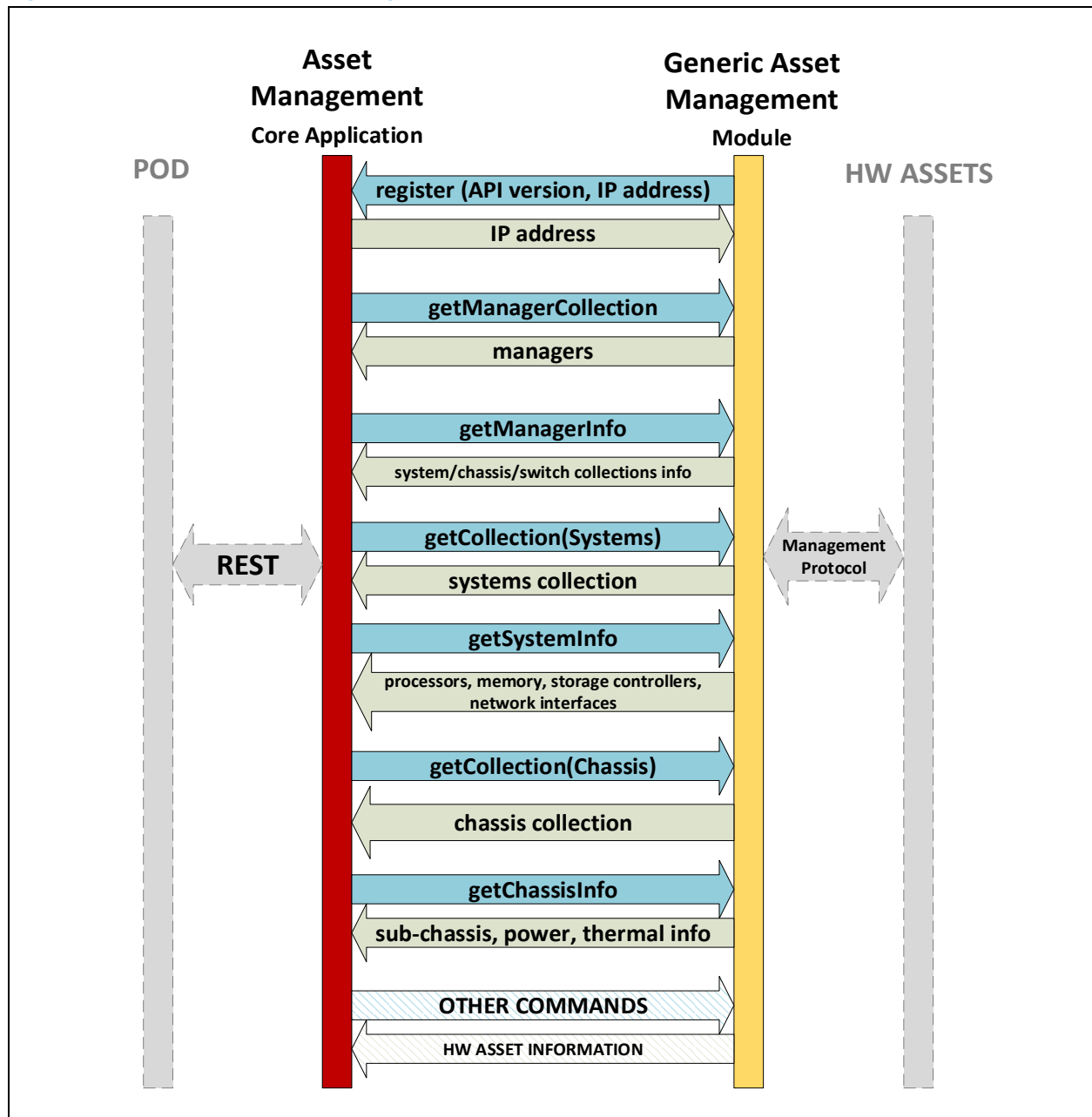
§

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 6 shows an example of the GAM module registration command sequence.

Figure 6 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
gamild	String	Yes	GAM module unique identifier
capabilities	Array: String	Yes	Capability name: "Compute", "Network", "StorageServices", "Chassis"

Serialization:

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

Example:

```
{
  "jsonrpc": "2.0",
  "method": "attach",
  "params": {
    "version": "1.0.0",
    "vendor": "Intel Corporation",
    "ipv4address": "127.0.0.1",
    "port": "8080",
    "gamiId": "40bflaa4-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
gamild	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 manager collection

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

5.3.1 Request

Serialization:

```

{
    "jsonrpc": "2.0",
    "method": "getManagerCollection",
    "id": <id>
}

```



Example

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

5.3.2 Response

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

Table 8 Get manager collection response

Parameters	Type	Mandatory	Description			
managers	Array: Object	Yes	Member	Type	Mandatory	Description
			manager	String	Yes	Manager UUID

Serialization

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

Example:

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

5.4 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 resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			health	String	No	Overall health state from the view of this resource: "OK", "Warning", "Critical"
type	String	Yes	Manager type: "ManagementController", "EnclosureManager", "BMC", "RackManager", "EthernetSwitchManager", "StorageSystemManager", "AuxiliaryController"			
model	String	No	Manager model			
location	String	No	UUID of the chassis being a manager's physical location			
parentId	String	No	Identifier of the parent manager			
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
graphicalConsole	Object	No	enabled	Boolean	Yes	Is service enabled
			Attribute	Type	Mandatory	Description
			enabled	Boolean	Yes	Console availability
			maxSessions	Number	No	Number of session that can be established at the same time



Result	Type	Mandatory	Description			
			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"
			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",
      "health": "OK"
    }
  }
}
```



```
    },
    "type": <{"ManagementController", "EnclosureManager", "BMC",
    "RackManager", "EthernetSwitchManager",
    "StorageSystemManager", "AuxiliaryController"}>,
    "model": <string>,
    "location": <string>,
    "parentId": <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"}>,
            "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",
    "parentId": "85a53567-349b-d293-4ab6-342655440ab3",
    "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": "Systems",
        "type": "Systems"
      },
    ],
  }
}
```



```
{
    {
        "name": "Chassis",
        "type": "Chassis",
    },
    "oem": {}
},
"id": 123
}
```

5.5 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.

5.5.1 Request

Table 11 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": "Blades"
    },
    "id": 987
}
```

5.5.2 Response

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

Table 12 Get collection response

Parameters	Type	Mandatory	Description	
subcomponents	Array: Object	Yes	Name	subcomponents



Parameters	Type	Mandatory	Description			
			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
}
```

5.6 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.

5.6.1 Notification

Table 13 Component notification

Parameters	Type	Mandatory	Description
gamild	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".



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": "40bflaa4-c440-11e5-9202-00a0c923456f",
    "component": "123e4567-e89b-12d3-a456-426655440000",
    "notification": "Remove",
    "parent": "223e4567-e89b-12d3-a456-426655441234"
    "type": "PSU"
    timeStamp: 5443323,
  }
}
```

5.7 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.

5.7.1 Request

Table 14 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
}
```

5.7.2 Response

Serialization:

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

Example:

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

5.8 Get computer system info

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

5.8.1 Request

Table 15 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
}
```

5.8.2 Response

Table 16 Get Computer System Info response

Results	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			health	String	Yes	Overall health state from the view of this resource: "OK", "Warning", "Critical"
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"			
bootOverrideSupported	Array: String	Yes	Boot source override supported targets: "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	Attribute	Type	Mandatory	Description
			vendorId	String	Yes	String with 4 digits hex number
			deviceId	String	Yes	String with 4 digits hex number
usbDevices	Array: Object	No	Attribute	Type	Mandatory	Description
			vendorId	String	Yes	String with 4 digits hex number
			deviceId	String	Yes	String with 4 digits hex number
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



Results	Type	Mandatory	Description																
sku	String	No	System SKU																
assetTag	String	No	Asset tag																
indicatorLED	String	No	Indicator LED info																
collections	Array: Object	Yes	<div>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", "MemoryChunks", "Dimms", "StorageControllers", "NetworkInterfaces"</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></div>	Attribute	Type	Mandatory	Description	name	String	Yes	Collection name	type	String	Yes	Collection type: "Processors", "MemoryChunks", "Dimms", "StorageControllers", "NetworkInterfaces"	slotMask	String	No	Occupied slots mask (see section 4.5 for details)
Attribute	Type	Mandatory	Description																
name	String	Yes	Collection name																
type	String	Yes	Collection type: "Processors", "MemoryChunks", "Dimms", "StorageControllers", "NetworkInterfaces"																
slotMask	String	No	Occupied slots mask (see section 4.5 for details)																
chassis	String	Yes	UUID of the chassis being the physical container for the system																
oem	Object	No	OEM specific data																

Serialization

```
{
  "jsonrpc": "2.0",
  "result": {
    "status": {
      "state": <string>,
      "health": <string>
    },
    "biosVersion": <string>,
    "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>,\n        "manufacturer": <string>,\n        "modelName": <string>,\n        "partNumber": <string>\n    },\n    "sku": <string>,\n    "assetTag": <string>,\n    "indicatorLED": <string>,\n    "collections": [\n        {\n            "name": <string>,\n            "type": <{"Processors", "MemoryChunks", "Dimms",\n                "StorageControllers", "NetworkInterfaces"}>,\n            "slotMask": <string>\n        }\n        ...\n    ],\n    "chassis": <string>,\n    "oem": <object>\n    },\n    "id": <id>\n}\n}
```

Example

```
{\n  "jsonrpc": "2.0",\n  "result": {\n    "status": {\n      "state": "Enabled",\n      "health": "OK"\n    },\n    "biosVersion": "A20F21_A0",\n    "bootOverride": "Continuous",\n    "bootOverrideTarget": "Pxe",\n    "bootOverrideSupported": [\n      "Pxe", "Cd", "Usb", "Hdd", "BiosSetup", "Utilities",\n      "Diags"\n    ],\n    "powerState": "On",\n    "pciDevices": [\n      {\n        "vendorId" : "0x8086",\n        "deviceId" : "0x1234"\n      }\n    ],\n    "usbDevices": [\n      {\n        "vendorId" : "0x8086",\n        "deviceId" : "0x5678"\n      }\n    ],\n    "fruInfo": {\n      "serialNumber": "123fed3029c-b23394-12",\n      "manufacturer": "Intel Corporation",\n      "modelName": "E323",\n      "partNumber": "29ee2220939"\n    }\n  }\n}
```



```
    },
    "sku": "System SKU",
    "assetTag": "User defined asset tag",
    "indicatorLED": "Lit",
    "collections": [
      {
        "name": "Processors",
        "type": "Processors",
        "slotMask": "11"
      },
      {
        "name": "Memory",
        "type": "MemoryChunks"
      },
      {
        "name": "Dimms",
        "type": "Dimms",
        "slotMask": "10101010"
      },
      {
        "name": "Storage",
        "type": "StorageControllers"
      },
      {
        "name": "Network",
        "type": "NetworkInterfaces"
      }
    ],
    "chassis": "123e4567-e89b-12d3-a456-426655440000",
    "oem": {}
  },
  "id": 234
}
```

5.9 Set computer system attributes

The *Set Component Attributes* described in section 5.7 allows configuration of computer system attributes listed in the following table.

Table 17 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"
powerState	String	Power state of the system: "On", "ForceOff", "GracefulShutdown", "ForceRestart", "Nmi", "ForceOn", "PushPowerButton"
oem	Object	OEM specific data

5.10 Get processor info

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



5.10.1 Request

Table 18 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
}
```

5.10.2 Response

Table 19 Get processor info response

Result	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			health	String	Yes	Overall health state from the view of this resource: "OK", "Warning", "Critical"
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):			



Result	Type	Mandatory	Description																												
			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>microcodeInfo</td><td>String</td><td>No</td><td>Microcode version</td></tr></table>	Attribute	Type	Mandatory	Description	vendorId	String	No	Numeric vendor id: "GenuineIntel"	numericId	String	No	cpuid.1.eax[31:0]	family	String	No	CPU family	model	String	No	CPU model	step	String	No	CPU stepping	microcodeInfo	String	No	Microcode version
Attribute	Type	Mandatory	Description																												
vendorId	String	No	Numeric vendor id: "GenuineIntel"																												
numericId	String	No	cpuid.1.eax[31:0]																												
family	String	No	CPU family																												
model	String	No	CPU model																												
step	String	No	CPU stepping																												
microcodeInfo	String	No	Microcode version																												
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": <string>,
      "health": <string>
    },
    "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
}

```

5.11 Get memory chunk Info

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



5.11.1 Request

Table 20 Get memory chunk Info request

Parameters	Type	Mandatory	Description
Chunk	String	Yes	Managed memory chunk UUID

Serialization:

```
{
  "jsonrpc": "2.0",
  "method": "getMemoryChunkInfo",
  "params": {
    "chunk": <string>
  },
  "id": <id>
}
```

Example:

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

5.11.2 Response

Table 21 Get memory chunk Info response

Result	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			health	String	Yes	Overall health state from the view of this resource: "OK", "Warning", "Critical"
chunkName	String	No	Name for the memory chunk			
chunkId	Number	No	Memory chunk UID			
chunkSizeMB	Number	Yes	Size of the memory chunk in MB			
type	String	Yes	Memory chunk type: "Volatile", "Persistent", "Block"			
mirrored	Boolean	No	Indicates if memory chunk is mirrored			
spare	Boolean	No	Indicates if memory chunk is a spare chunk			
interleaveSets	Array: Object	No	Attribute	Type	Mandatory	Description
			dimm	String	Yes	DIMM module UUID
			regionId	String	Yes	Region identifier
oem	Object	No	OEM specific data			

Serialization:

```
{
```

```

    "jsonrpc": "2.0",
    "result": {
        "status": {
            "state": <string>,
            "health": <string>
        },
        "chunkName": <string>,
        "chunkId": <number>,
        "chunkSizeMB": <number>,
        "type": <{"Volatile", "Block", "Persistent"}>,
        "mirrored": <boolean>,
        "spare": <boolean>,
        "interleaveSets": [{
            "dimm": <string>,
            "regionId": <string>
        },
        ...
    ],
    "oem": <object>
},
    "id": <id>
}

```

Example:

```

{
    "jsonrpc": "2.0",
    "result": {
        "status": {
            "state": "Enabled",
            "health": "OK"
        },
        "chunkName": "1LM",
        "chunkId": 1,
        "chunkSizeMB": 8192,
        "type": "Volatile",
        "mirrored": false,
        "spare": false,
        "interleaveSets": [
            {
                "dimm": "123e4567-e89b-12d3-a456-426655440000",
                "regionId": "1"
            },
            {
                "dimm": "85a53567-349b-d293-4ab6-342655440ab3",
                "regionId": "1"
            }
        ],
        "oem": {}
    },
    "id": 987
}

```

5.12 Get Dimm info

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



5.12.1 Request

Table 22 Get Dimm info request

Parameters	Type	Mandatory	Description
dimm	String	Yes	Managed DIMM UUID

Serialization:

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

Example:

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

5.12.2 Response

Table 23 Get Dimm info response

Result	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			health	String	Yes	Overall health state from the view of this resource: "OK", "Warning", "Critical"
memoryType	String	No	Memory type: "DRAM", "NVDIMM_N", "NVDIMM_F", "NVDIMM_P"			
dimmType	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"			
dimmMedia	Array: String	No	Memory module media types: "DRAM", "NAND", "Proprietary"			
memoryModes	Array: String	No	Memory modes supported by the memory module: "Volatile", "Persistent", "Block"			



Result	Type	Mandatory	Description			
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	Attribute	Type	Mandatory	Description
			serialNumber	String	No	Serial number
			manufacturer	String	No	Manufacturer name
			modelName	String	No	Model number
			partNumber	String	No	Part number
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			
dimmLocation	Object	No	Property describing DIMM location with respect to processor and memory controller			
			Attribute	Type	Mandatory	Description
			socket	Number	No	Socket number
			controller	Number	No	Memory controller number
			channel	Number	No	Channel number
slot	Number	No	Slot number			
rankCount	Number	No	Number of ranks available in the memory module			
errorCorrection	String	No	Error correction scheme supported for this memory: “NoECC”, “SingleBitECC”, “MultiBitECC”, “AddressParity”			
regions	Array: Object	No	Memory regions configured within the memory module			
			Attribute	Type	Mandatory	Description
			regionId	String	No	Unique region ID representing a specific region within the memory module
			memoryType	String	No	Region memory type: “Volatile”, “Block”, “Persistent”
			offsetMB	Number	No	Offset with in the memory module corresponding to the region begin
sizeMB	Number	No	Region size in MB			
OperatingModes	Array	No				
oem	Object	No	OEM specific data			

Serialization:

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



```
"result": {
  "status": {
    "state": <string>,
    "health": <string>
  },
  "memoryType": <{"DRAM", "NVDIMM_N", "NVDIMM_F", "NVDIMM_P"}>,
  "dimmType": <{"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"}>,
  "dimmMedia": [
    <{"DRAM", "NAND", "Proprietary"}>,
    ...
  ],
  "memoryModes": [
    <{"Volatile", "Persistent", "Block"}>,
    ...
  ],
  "capacityMB": <number>,
  "dataWidthBits": <number>,
  "busWidthBits": <number>,
  "fruInfo": {
    "serialNumber": <string>,
    "manufacturer": <string>,
    "modelName": <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>
  "dimmLocation": {
    "socket": <number>,
    "controller": <number>,
    "channel": <number>,
    "slot": <number>
  },
}
```



```
    "rankCount": <number>,\n    "errorCorrection": <{"NoECC", "SingleBitECC", "MultiBitECC" ,\n        "AddressParity"}>,\n    "regions": [{\n        "regionId": <string>,\n        "memoryType": <{"Volatile", "Block", "Persistent"}>,\n        "offsetMB": <number>,\n        "sizeMB": <number>\n    }]\n    ...]\n    "oem": <object>\n},\n"id": <id>\n}
```

Example:

```
{\n  "jsonrpc": "2.0",\n  "result": {\n    "status": {\n      "state": "Enabled",\n      "health": "OK"\n    },\n    "memoryType": "DRAM",\n    "dimmType": "DDR4",\n    "moduleType": "RDIMM",\n    "dimmMedia": [\n      "DRAM"\n    ],\n    "memoryModes": [\n      "Volatile"\n    ],\n    "capacityMB": 16384,\n    "dataWidthBits": 64,\n    "busWidthBits": 72,\n    "fruInfo": {\n      "serialNumber": "123fed3029c-b23394-12",\n      "manufacturer": "Intel Corporation",\n      "modelName": "E323",\n      "partNumber": "29ee2220939"\n    },\n    "firmwareRevision": "RevAbc",\n    "firmwareApiVersion": "ApiAbc",\n    "functionClasses": [\n      "Volatile"\n    ],\n    "vendorID": "0x8086",\n    "deviceID": "0xAB245f",\n    "operatingSpeedMHz": 2400,\n    "allowedSpeedsMHz": [\n      2133,\n      2400,\n      2667\n    ],\n    "voltageVolt": 1.35,\n  }\n}
```



```
        "minimumVoltageVolt": 1.3,
        "maximumVoltageVolt": 1.4,
        "deviceLocator": "PROC 1 DIMM 1"
        "dimmLocation": {
            "socket": 1,
            "controller": 1,
            "channel": 1,
            "slot": 1
        },
        "rankCount": 1,
        "errorCorrection": "MultiBitECC",
        "regions": [{
            "regionId": "1",
            "memoryType": "Volatile",
            "offsetMB": 0,
            "sizeMB": 16384
        }
    ],
    "oem": {}
},
"id": 987
}
```

5.13 Get storage controller info

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

5.13.1 Request

Table 24 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
}
```



5.13.2 Response

Table 25 Get storage controller info response

Result	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			health	String	Yes	Overall health state from the view of this resource: "OK", "Warning", "Critical"
interface	String	Yes	Storage controller interface: "PCIe", "SAS", "SATA"			
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
collections	Array: Object	Yes	Subcomponents collections			
			Attribute	Type	Mandatory	Description
			name	String	Yes	Collection name
			type	String	Yes	Collection type: "Drives"
			slotMask	String	No	Occupied slots mask (see section 4.5 for details)
physicalId	String	No	Storage controller physical Id (bus type & location)			
oem	Object	No	OEM specific data			

Serialization:

```
{
  "jsonrpc": "2.0",
  "result": {
    "status": {
      "state": <string>,
      "health": <string>
    },
    "interface": <string>,
    "fruInfo": {
      "serialNumber": <string>,
      "manufacturer": <string>,
      "modelNumber": <string>,
      "partNumber": <string>
    },
    "collections": [
      {
        "name": <string>,
        "type": <{"Drives"}>,
        "slotMask": <string>
      }
    ]
  }
}
```



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

Example:

```
{
  "jsonrpc": "2.0",
  "result": {
    "status": {
      "state": "Enabled",
      "health": "OK"
    },
    "interface": "SAS"
    "fruInfo": {
      "serialNumber": "123fed3029c-b23394-12",
      "manufacturer": "Intel Corporation",
      "modelName": "E323",
      "partNumber": "29ee2220939"
    },
    "collections": [
      {
        "name": "Drives",
        "type": "Drives"
      }
    ],
    "physicalId" : "pci@0000:01:00.0",
    "oem": {}
  },
  "id": 987
}
```

5.14 Get drive info

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

5.14.1 Request

Table 26 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
}
```

5.14.2 Response

Table 27 Get drive info response

Result	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			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"			
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)			
oem	Object	No	OEM specific data			

Serialization:

```
{
  "jsonrpc": "2.0",
  "result": {
    "status": {
      "state": <string>,
      "health": <string>
    },
    "interface": <string>
    "type": <string>,
    "rpm": <number>,
    "capacityGB": <number>,
    "fruInfo": {
      "serialNumber": <string>,
      "manufacturer": <string>,
      "modelNumber": <string>,
      "partNumber": <string>
    }
  }
}
```



```
    },
    "firmwareVersion": <string>,
    "physicalId" : <string>,
    "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",
      "modelName": "E323",
      "partNumber": "29ee2220939"
    },
    "firmwareVersion": "0002",
    "physicalId" : "0.1.0",
    "oem": {}
  },
  "id": 987
}
```

5.15 Get network interface Info

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

5.15.1 Request

Table 28 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
}
```

5.15.2 Response

Table 29 Get network interface info response

Result	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			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			



Result	Type	Mandatory	Description			
collections	Array: Object	Yes	Subcomponents collections			
			Attribute	Type	Mandatory	Description
			name	String	Yes	Collection name
			type	String	Yes	Collection type "PortVlans"
			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": <string>,
      "health": <string>
    },
    "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": <{"PortVlans"}>,
      "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": "Vlans",
        "type": "PortVlans"
      }
    ],
    "oem": {}
  },
  "id": 987
}

```

5.16 Set network interface attributes

The *Set Component Attributes* described in the section 5.7 allows configuration of the network interface attributes listed in the following table.

**Table 30 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

5.17 Get Ethernet switch info

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

5.17.1 Request

Table 31 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
}
```

5.17.2 Response

Table 32 Get Ethernet switch info response

Result	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			health	String	Yes	Overall health state from the view of this resource: "OK", "Warning", "Critical"
switchIdentifier	String	Yes	Switch identifier			
technology	String	No	Switch technology: "Ethernet", "PCIe"			



Result	Type	Mandatory	Description			
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	Attribute	Type	Mandatory	Description
			serialNumber	String	No	Module serial number
			manufacturer	String	No	Manufacturer name
			modelName	String	No	Model number
			partNumber	String	No	Part number
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			
			Attribute	Type	Mandatory	Description
			name	String	Yes	Collection name
			type	String	Yes	Collection type: "Ports", "Vlans", "Acls", "NeighborSwitches"
			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": <string>,
      "health": <string>
    },
    "switchIdentifier": <string>
    "technology": <string>,
    "macAddress": <string>,
    "firmwareName": <string>,
    "firmwareVersion": <string>,
    "role": <{"TOR", "EOR", "Drawer", "Unknown"}>,
    "fruInfo": {
      "serialNumber": <string>,
      "manufacturer": <string>,
      "modelName": <string>,
      "partNumber": <string>
    },
    "manufacturingDate": <string>,
    "location": <number>,
    "chassis": <string>,
    "collections": [
      {
        "name": <string>,
```



```
        "type": <{"Ports", "Vlans", "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",  
      "modelName": "E323",  
      "partNumber": "29ee2220939"  
    },  
    "manufacturingDate": "02/21/2015 00:00:00",  
    "location": 1,  
    "chassis": "123e4567-e89b-12d3-a456-426655440000",  
    "maxAclNumber": 4,  
    "collections": [  
      {  
        "name": "Ports",  
        "type": "Ports",  
        "slotMask": "11111100011110001111000111101111"  
      },  
      {  
        "name": "Neighbors",  
        "type": "NeighborSwitches"  
      },  
      {  
        "name": "Acls",  
        "type": "Acls"  
      }  
    ],  
    "oem": {}  
  },  
  "id": 987  
}
```



5.18 Get Ethernet switch port info

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

5.18.1 Request

Table 33 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
}
```

5.18.2 Response

Table 34 Get Ethernet switch port 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 resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"</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>	Attribute	Type	Mandatory	Description	state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"	health	String	Yes	Overall health state from the view of this resource: "OK", "Warning", "Critical"
			Attribute	Type	Mandatory	Description									
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"									
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"												



Result	Type	Mandatory	Description																				
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	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>address</td><td>String</td><td>Yes</td><td>IPv4 address</td></tr><tr><td>subnetMask</td><td>String</td><td>No</td><td>IPv4 subnet mask</td></tr><tr><td>addressOrigin</td><td>String</td><td>No</td><td>IPv4 address origin "DHCP", "Static"</td></tr><tr><td>gateway</td><td>String</td><td>No</td><td>IPv4 gateway for this address</td></tr></table>	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
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	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>address</td><td>String</td><td>Yes</td><td>IPv6 address</td></tr><tr><td>prefixLength</td><td>Number</td><td>No</td><td>IPv6 Address Prefix Length</td></tr><tr><td>addressOrigin</td><td>String</td><td>No</td><td>IPv4 address origin "DHCP", "Static", "SLAAC"</td></tr><tr><td>addressState</td><td>String</td><td>No</td><td>IPv6 address state: "Preferred", "Deprecated", "Tentative", "Failed"</td></tr></table>	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"
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	<table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>switchIdentifier</td><td>String</td><td>No</td><td>Remote switch identifier</td></tr><tr><td>portIdentifier</td><td>String</td><td>No</td><td>Port identifier on the remote switch which is connected to a given local switch port</td></tr><tr><td>cableId</td><td>String</td><td>No</td><td>Cable identifier</td></tr></table>	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				
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 <table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr></table>	Attribute	Type	Mandatory	Description																
Attribute	Type	Mandatory	Description																				



Result	Type	Mandatory	Description			
			name	String	Yes	Collection name
			type	String	Yes	Collection type: "PortVlans", "PortMembers", "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": <string>,
      "health": <string>
    },
    "portIdentifier": <string>,
    "portClass": <{"Physical", "Logical", "Reserved"}>
    "Unknown">,
    "portType": <{"Upstream", "Downstream", "MeshPort",
    "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>,\n        "collections": [\n            {\n                "name": <string>,\n                "type": <{"PortVlans", "PortMembers", "Acls",\n\n"StaticMacs"}>,\n                "slotMask": <string>\n            }\n            ...\n        ],\n        "oem": <object>\n    },\n    "id": <id>\n}
```

Example:

```
{\n    "jsonrpc": "2.0",\n    "result": {\n        "status": {\n            "state": "Enabled",\n            "health": "OK"\n        },\n        "portIdentifier": "1",\n        "portClass": "Physical",\n        "portType": "Downstream",\n        "linkTechnology": "Ethernet",\n        "linkSpeedMbps": 1000,\n        "maxSpeedMbps": 10000,\n        "operationalState": "Up",\n        "administrativeState": "Up",\n        "portWidth": 1,\n        "frameSize": 1520,\n        "autosense": true,\n        "fullDuplex": true,\n        "isManagementPort": false,\n        "lastErrorCode": 0,\n        "errorCleared": false,\n        "lastStateChangeTime": "2015-02-23T14:44:00+00:00",\n        "macAddress": "AA:BB:CC:DD:EE:FF",\n        "ipv4Address": {\n            "address": "10.0.2.10",\n            "subnetMask": "255.255.255.0",\n            "addressOrigin": "DHCP",\n            "gateway": "10.0.2.1"\n        },\n        "ipv6Address": {\n            "address": "fe80::1ec1:deff:fe6f:1c37",\n            "prefixLength": 16,\n            "addressOrigin": "DHCP",\n            "addressState": "Preferred"\n        },\n        "neighborInfo": {\n            "switchIdentifier": "123e4567-e89b-12d3-a456-\n426655440000",
```

```

        "portIdentifier": "19",
        "cableId": "TOR port 19"
    },
    "neighborMac": "12:34:56:78:90:AB",
    "vlanEnable": true,
    "defaultVlan": "cd3e4527-af7b-32d3-a489-987656a3d588",
    "collections": [
        {
            "name": "Vlans",
            "type": "PortVlans"
        },
        {
            "name": "Acls",
            "type": "Acls"
        },
        {
            "name": "StaticMacs",
            "type": "StaticMacs"
        }
    ],
    "oem": {}
},
"id": 987
}

```

5.19 Set Ethernet switch port attributes

The *Set Component Attributes* described in the section 5.7 allows configuration of the switch port attributes listed in the following table.

Table 35 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

5.20 Add Ethernet switch port

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

5.20.1 Request

Table 36 Add Ethernet switch Port request

Parameters	Type	Mandatory	Description
switch	String	Yes	Managed switch UUID
portIdentifier	String	Yes	Port identifier



Parameters	Type	Mandatory	Description
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": "UplinkAggregation1",
    "mode": "LinkAggregationStatic",
    "members": [
      "cd3e4527-af7b-32d3-a489-987656a3d588",
      "863e4567-e87b-64d3-a489-987656540000"
    ],
    "oem": {}
  },
  "id": 987
}
```

5.20.2 Response

Table 37 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
}

```

5.21 Delete Ethernet switch port

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

5.21.1 Request

Table 38 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
}

```

5.21.2 Response

Table 39 Delete Ethernet switch port response

Result	Type	Mandatory	Description
oem	object	No	OEM specific data



Serialization:

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

Example:

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

5.22 Add Ethernet switch port members

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

5.22.1 Request

Table 40 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
}
```

5.22.2 Response

Table 41 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
}
```

5.23 Delete Ethernet switch port members

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

5.23.1 Request

Table 42 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
}
```

5.23.2 Response

Table 43 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
}
```

5.24 Get remote Ethernet switch info

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

5.24.1 Request

Table 44 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
}
```

5.24.2 Response

Table 45 Get remote Ethernet switch info response

Result	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			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": <string>,
    "health": <string>
  },
  "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
}
```

5.25 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).

5.25.1 Request

Table 46 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
}

```

5.25.2 Response

Table 47 Get VLAN info request

Result	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			health	String	Yes	Overall health state from the view of this resource: "OK", "Warning", "Critical"
vlanId	Number	Yes	VLAN identifier			
vlanName	String	No	VLAN name			
vlanEnable	Boolean	Yes	Indicates if VLAN is enabled			
oem	Object	No	OEM specific data			

Serialization:

```

{
    "jsonrpc": "2.0",
    "result": {
        "status": {
            "state": <string>,
            "health": <string>
        },
        "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
}
```

5.26 Set VLAN attributes

The *Set Component Attributes* described in the section 5.7 allows configuration of the VLAN attributes listed in the following table.

Table 48 Configurable VLAN attributes

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

5.27 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).

5.27.1 Request

Table 49 Add VLAN request

Parameters	Type	Mandatory	Description
switch	String	Yes	Managed switch UUID
vlanId	Number	Yes	New VALN Identifier
vlanName	String	No	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
}

```

5.27.2 Response

Table 50 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
}

```

5.28 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).

5.28.1 Request

Table 51 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  
}
```

5.28.2 Response

Table 52 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  
}
```

5.29 Get port VLAN info

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

5.29.1 Request

Table 53 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
}

```

5.29.2 Response

Table 54 Get port VLAN info response

Result	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			health	String	Yes	Overall health state from the view of this resource: "OK", "Warning", "Critical"
vlanId	Number	Yes	VLAN identifier for this port			
vlanName	String	No	VLAN name			
vlanEnable	Boolean	Yes	Indicates if VLAN is enabled			
tagged	Boolean	Yes	Indicates if it is tagged VLAN (if not it is port based VLAN)			
oem	Object	No	OEM specific data			

Serialization:

```

{
    "jsonrpc": "2.0",
    "result": {
        "status": {
            "state": <string>,
            "health": <string>
        },
        "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
  }
}
```

5.30 Set port VLAN attributes

The *Set Component Attributes* described in the section 5.7 allows configuration of the Port VLAN attributes listed in the following table.

Table 55 Configurable port VLAN attributes

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

5.31 Add port VLAN

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

5.31.1 Request

Table 56 Add Port VLAN request

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

Serialization:

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



Example:

```
{
  "jsonrpc": "2.0",
  "method": "addPortVlan",
  "params": {
    "port": "123e4567-e89b-12d3-a456-426655440000",
    "vlanId": 102,
    "vlanName": "Management VLAN",
    "tagged": true,
    "oem": {}
  },
  "id": 987
}
```

5.31.2 Response

Table 57 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
}
```

5.32 Delete port VLAN

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

5.32.1 Request

Table 58 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
}
```

5.32.2 Response

Table 59 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
}
```

5.33 Get port static MAC info

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



5.33.1 Request

Table 60 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
}
```

5.33.2 Response

Table 61 Get port static MAC info response

Result	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			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": <string>,
      "health": <string>
    },
    "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
}
```

5.34 Set port static MAC attributes

The *Set Component Attributes* described in the section 5.7 allows configuration of the Port Static MAC attributes listed in the following table.

Table 62 Configurable port static MAC attributes

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

5.35 Add port static MAC

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

5.35.1 Request

Table 63 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
}

```

5.35.2 Response

Table 64 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
}

```

5.36 Delete port static MAC

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

5.36.1 Request

Table 65 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
}
```

5.36.2 Response

Table 66 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
}
```

5.37 Get ACL info

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



5.37.1 Request

Table 67 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
}
```

5.37.2 Response

Table 68 Get ACL info response

Parameters	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			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: "Rules", "Ports"
oem	Object	No	slotMask	String	No	Occupied slots mask (see section 4.5 for details)
			OEM specific data			

Serialization:

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

```

    "result": {
        "status": {
            "state": <string>,
            "health": <string>
        },
        "collections": [
            {
                "name": <string>,
                "type": <{"Rules", "Ports"}>,
                "slotMask": <string>
            }
            ...
        ],
        "oem": <object>
    },
    "id": <id>
}

```

Example:

```

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

```

5.38 Add ACL

Add ACL command creates new ACL on the switch port.

5.38.1 Request

Table 69 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
}
```

5.38.2 Response

Table 70 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
}
```

5.39 Delete ACL

Delete ACL command removes the existing ACL.

5.39.1 Request

Table 71 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
}
```

5.39.2 Response

Table 72 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
}
```

5.40 Add ACL port

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

5.40.1 Request

Table 73 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
}
```



5.40.2 Response

Table 74 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
}
```

5.41 Delete ACL port

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

5.41.1 Request

Table 75 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
}

```

5.41.2 Response

Table 76 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
}

```

5.42 Get ACL rule info

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

5.42.1 Request

Table 77 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
}
```

5.42.2 Response

Table 78 Get ACL rule info response

Parameters	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			health	String	Yes	Overall health state from the view of this resource: "OK", "Warning", "Critical"
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	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	No	MAC mask



Parameters	Type	Mandatory	Description			
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": <string>,
      "health": <string>
    },
    "ruleId": <number>,
    "action": <{"Permit", "Deny", "Forward", "Mirror"}>,
    "forwardMirrorPort": <string>,
    "mirroredPorts": [
      <string>,
      ...
    ],
    "mirrorType": <{"Egress", "Ingress", "Bidirectional",
"Redirect"}>,
    "vlanId": {
      "id": <number>,
      "mask": <number>
    },
    "sourceIp": {
      "address": <string>,
      "mask": <string>
    },
    "destinationIp": {
      "address": <string>,
      "mask": <string>
    },
    "sourceMac": {
      "address": <string>,
      "mask": <string>
    },
    "destinationMac": {
      "address": <string>,
      "mask": <string>
    },
    "sourceL4Port": {
      "port": <number>,
      "mask": <number>
    },
    "destinationL4Port": {
      "port": <number>
    }
  }
}
```



```
        "mask": <number>
    },
    "protocol": <number>,
    "oem": {}
},
"id": <id>
}
```

Example:

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

5.43 Add ACL rule

Add ACL Rule command creates new rule on the ACL.

5.43.1 Request

Table 79 Add ACL rule request

Parameters	Type	Mandatory	Description
acl	String	Yes	Managed ACL UUID, owner of the rule.



Parameters	Type	Mandatory	Description			
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



Parameters	Type	Mandatory	Description				
			<table><tr><td>mask</td><td>Number</td><td>No</td><td>L4 port mask</td></tr></table>	mask	Number	No	L4 port mask
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": {
      "address": <string>,
      "mask": <string>
    },
    "destinationMac": {
      "address": <string>,
      "mask": <string>
    },
    "sourceL4Port": {
      "port": <number>,
      "mask": <number>
    },
    "destinationL4Port": {
      "port": <number>,
      "mask": <number>
    },
    "protocol": <number>,
    "oem": {}
  },
  "id": <id>
}
```



Example:

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

5.43.2 Response

Table 80 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
}
```

5.44 Delete ACL rule

Delete ACL rule command removes a rule from the ACL.

5.44.1 Request

Table 81 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
}
```

5.44.2 Response

Table 82 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
}
```

5.45 Set ACL rule attributes

The *Set Component Attributes* described in the section 5.7 allows configuration of the ACL rule attributes listed in the following table.

Table 83 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
		id
		Number
		Yes
		VLAN identifier
		mask
		Number
		No
		VLAN mask number
sourceIp	Object	Attribute
		Type
		Mandatory
		address
		String
		Yes
		IPv4 address
		mask
		String
		No
		IP subnet mask
destinationIp	Object	Attribute
		Type
		Mandatory
		address
		String
		Yes
		IPv4 address
		mask
		String
		No
		IP subnet mask
sourceMac	Object	Attribute
		Type
		Mandatory
		address
		String
		Yes
		MAC address
		mask
		String
		No
		MAC mask
destinationMac	Object	Attribute
		Type
		Mandatory
		address
		String
		Yes
		MAC address
		mask
		String
		No
		MAC mask
sourceL4Port	Object	Attribute
		Type
		Mandatory
		port
		Number
		Yes
		L4 port
		mask
		Number
		No
		L4 port mask
destinationL4Port	Object	Attribute
		Type
		Mandatory
		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



5.46 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.

5.46.1 Request

Table 84 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
}
```

5.46.2 Response

Table 85 Get chassis info response

Result	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			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"			
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	Attribute	Type	Mandatory	Description
			serialNumber	String	No	Module serial number
			manufacturer	String	No	Manufacturer name
			modelName	String	No	Model number
			partNumber	String	No	Part number
sku	String	No	Switch SKU			
assetTag	String	No	Asset tag			
indicatorLED	String	No	Indicator LED info			
collections	Array: Object	Yes	Subcomponents collections			
			Attribute	Type	Mandatory	Description
			name	String	Yes	Collection name
			type	String	Yes	Collection type: "Chassis", "PowerZones", "ThermalZones"
			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": <string>,
      "health": <string>
    },
    "type": <string>,
    "size": <number>,
    "locationOffset": <number>,
    "parentId": <string>,
    "powerZone": <string>,
    "thermalZone": <string>,
    "fruInfo": {
      "serialNumber": <string>,
      "manufacturer": <string>,
      "modelName": <string>,
      "partNumber": <string>
    },
    "collections": [
      {
        "name": <string>,
        "type": <{"Chassis", "PowerZones",
"ThermalZones"}>,
        "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",
      "modelName": "E323",
      "partNumber": "29ee2220939"
    },
    "collections": [
      {
        "name": "Power",
        "type": "PowerZones"
      },
      {
        "name": "Thermal",
        "type": "ThermalZones"
      },
      {
        "name": "Chassis",
        "type": "Chassis"
      }
    ],
    "oem": {}
  },
  "id": 987
}
```

5.47 Get power zone info

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

5.47.1 Request

Table 86 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
}

```

5.47.2 Response

GAM Module replies with the following response in successful case:

Table 87 Get power zone info response

Parameters	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			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": <string>,
            "health": <string>
        },
        "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
}
```

5.48 Set power zone attributes

The *Set Component Attributes* described in the section 5.7 allows configuration of the power zone attributes listed in the following table.

Table 88 Configurable Power Zone Attributes

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

5.49 Get PSU info

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



5.49.1 Request

Table 89 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
}
```

5.49.2 Response

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

Table 90 Get PSU info response

Parameters	Type	Mandatory	Description			
			Attribute	Type	Mandatory	Description
status	Object	Yes	state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			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": <string>,
      "health": <string>
    },
    "slot": <number>,
    "powerInput": <number>,
    "powerOutput": <number>,
    "powerCapacity": <number>,
    "fruInfo": {
      "serialNumber": <string>,
      "manufacturer": <string>,
      "modelName": <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",
      "modelName": "E323",
      "partNumber": "29ee2220939"
    },
    "oem": {}
  },
  "id": 345
}
```

5.50 Set PSU attributes

The *Set Component Attributes* described in the section 5.7 allows configuration of the PSU attributes listed in the following table.

Table 91 Configurable PSU Attributes

Attribute	Type	Description
powerInput	Number	Present power input
powerOutput	Number	Present power output



oem	Object	OEM specific data
-----	--------	-------------------

5.51 Get thermal zone info

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

5.51.1 Request

Table 92 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
}
```

5.51.2 Response

GAM Module replies with the following response in successful case:

Table 93 Get Thermal Zone Info response

Parameters	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			health	String	Yes	Overall health state from the view of this resource: "OK", "Warning", "Critical"
temperature	Number	Yes	Present temperature			
collections	Array: Object	Yes	Subcomponents collections			
			Attribute	Type	Mandatory	Description
			name	String	Yes	Collection name



			type	String	Yes	Collection type: "Fans"
			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": <string>,
      "health": <string>
    },
    "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
}
```

5.52 Get fan info

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



5.52.1 Request

Table 94 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
}
```

5.52.2 Response

GAM Module replies with the following response in successful case:

Table 95 Get fan info response

Parameters	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			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": <string>,
      "health": <string>
    },
    "slot": <number>,
    "currentSpeed": <number>,
    "desiredSpeed": <number>
    "fruInfo": {
      "serialNumber": <string>,
      "manufacturer": <string>,
      "modelName": <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",
      "modelName": "E323",
      "partNumber": "29ee2220939"
    },
    "oem": {}
  },
  "id": 987
}
```

5.53 Set fan attributes

The *Set Component Attributes* described in the section 5.7 allows configuration of the fan attributes listed in the following table.

Table 96 Configurable Fan Attributes

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

5.54 Get authorization certificate

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

5.54.1 Request

Table 97 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
}
```

5.54.2 Response

GAM Module replies with the following response in successful case:

Table 98 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.

5.55 Get storage services info

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

5.55.1 Request

Table 99 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": {
    "port": "123e4567-e89b-12d3-a456-426655440000"
  },
  "id": 987
}
```



5.55.2 Response

Table 100 Get storage services info response

Result	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			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"
oem	Object	No	slotMask	String	No	Occupied slots mask (see section 4.5 for details)
			OEM specific data			

Serialization:

```
{
  "jsonrpc": "2.0",
  "result": {
    "status": {
      "state": <string>,
      "health": <string>
    },
    "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",
        "slotMask": "00000000"
      }
    ]
    "oem": {}
  },
  "id": 1
}
```



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

5.56 Get physical drive info

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

5.56.1 Request

Table 101 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
}
```



5.56.2 Response

Table 102 Get drive info response

Result	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			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	Attribute	Type	Mandatory	Description
			serialNumber	String	No	Module serial number
			manufacturer	String	No	Manufacturer name
			modelName	String	No	Model number
			partNumber	String	No	Part number
oem	Object	No	OEM specific data			

Serialization:

```
{
  "jsonrpc": "2.0",
  "result": {
    "status": {
      "state": <string>,
      "health": <string>
    },
    "interface": <string>
    "type": <string>,
    "capacityGB": <number>,
    "rpm": <number>,
    "fruInfo": {
      "serialNumber": <string>,
      "manufacturer": <string>,
      "modelName": <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",
        "modelName": "E323",
        "partNumber": "29ee2220939"
    },
    "oem": {}
},
"id": 987
}

```

5.57 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.

5.57.1 Request

Table 103 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
}

```



5.57.2 Response

Table 104 Get logical drive info response

Result	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			health	String	Yes	Overall health state from the view of this resource: "OK", "Warning", "Critical"
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			
			Attribute	Type	Mandatory	Description
			name	String	Yes	Collection name
			type	String	Yes	Collection type: "PhysicalDrives", "LogicalDrives"
			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": <string>,
```



```
        "health": <string>
    },
    "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
}
```

5.58 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).

5.58.1 Request

Table 105 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
}
```

5.58.2 Response

Table 106 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
}
```



5.59 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.

5.59.1 Request

Table 107 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
}
```

5.59.2 Response

Table 108 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
}
```



5.60 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.

5.60.1 Request

Table 109 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": {
    "drive": "123e4567-e89b-12d3-a456-426655440000"
  },
  "id": 987
}
```

5.60.2 Response

Table 110 Get iSCSI target info response

Result	Type	Mandatory	Description			
status	Object	Yes	Attribute	Type	Mandatory	Description
			state	String	Yes	Known state of the resource: "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent"
			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



Result	Type	Mandatory	Description
oem	Object	No	OEM specific data

Serialization:

```
{
  "jsonrpc": "2.0",
  "result": {
    "status": {
      "state": <string>,
      "health": <string>
    },
    "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
}
```

5.61 Set iSCSI target attributes

The *Set Component Attributes* described in the section 5.7 allows configuration of the iSCSI target attributes listed in the following table.

**Table 111 Configurable iSCSI Target Attributes**

Attribute	Type	Description
targetIQN	String	iSCSI target IQN
initiatorIQN	String	iSCSI initiator IQN
oem	Object	OEM specific data

5.62 Add iSCSI target

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

5.62.1 Request

Table 112 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 <table><tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr><tr><td>LUN</td><td>Number</td><td>Yes</td><td>iSCSI target LUN</td></tr><tr><td>logicalDrive</td><td>String</td><td>Yes</td><td>Logical drive UUID</td></tr></table>	Attribute	Type	Mandatory	Description	LUN	Number	Yes	iSCSI target LUN	logicalDrive	String	Yes	Logical drive UUID
Attribute	Type	Mandatory	Description												
LUN	Number	Yes	iSCSI target LUN												
logicalDrive	String	Yes	Logical drive UUID												
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
}
```

5.62.2 Response

Table 113 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
}
```

5.63 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.

5.63.1 Request

Table 114 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
}
```

5.63.2 Response

Table 115 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

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

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

Table 116 GAMI commands support requirements

Command	Compute	Storage Services	Network	Chassis
GAM Module Registration	Mandatory	Mandatory	Mandatory	Mandatory
Get Managers Collection	Mandatory	Mandatory	Mandatory	Mandatory
Get Manager Info	Mandatory	Mandatory	Mandatory	Mandatory
Get Collection	Mandatory	Mandatory	Mandatory	Mandatory
Component Notification	Mandatory	Mandatory	Mandatory	Mandatory
Get Computer System Info	Mandatory			
Set Computer System Attributes	Mandatory			
Get Processor Info	Mandatory			
Get Memory Chunk Info	Mandatory			
Get DIMM Info	Mandatory			
Get Storage Controller Info	Mandatory			
Get Drive Info	Mandatory			
Get Network Interface Info	Mandatory			
Set Network Interface Attributes	Optional			
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	

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



Command	Compute	Storage Services	Network	Chassis
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	
Get Rule Info			Optional	
Add Rule			Optional	
Set Rule Attributes			Optional	
Delete Rule			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
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		

§

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)